Package ‘lmomco’

September 24, 2025

Type Package

Title L-Moments, Censored L-Moments, Trimmed L-Moments, L-Comoments, and Many Distributions

Version 2.5.3

Depends R (>= 3.5.0), utils

Imports goftest, Lmoments, MASS

Suggests copBasic

Date 2025-09-21

Description Extensive functions for Lmoments (LMs) and probability-weighted moments (PWMs), distribution parameter estimation, LMs for distributions, LM ratio diagrams, multivariate Lcomoments, and asymmetric (asy) trimmed LMs (TLMs). Maximum likelihood and maximum product spacings estimation are available. Right-tail and left-tail LM censoring by threshold or indicator variable are available. LMs of residual (resid) and reversed (rev) residual life are implemented along with 13 quantile operators for reliability analyses. Exact analytical bootstrap estimates of order statistics, LMs, and LM var-covars are available. Harri-Coble Tau34-squared Normality Test is available. Distributions with L, TL, and added (+) support for right-tail censoring (RC) encompass: Asy Exponential (Exp) Power [L], Asy Triangular [L], Cauchy [TL], Eta-Mu [L], Exp. [L], Gamma [L], Generalized (Gen) Exp Poisson [L], Gen Extreme Value [L], Gen Lambda [L, TL], Gen Logistic [L], Gen Normal [L], Gen Pareto [L+RC, TL], Govindarajulu [L], Gumbel [L], Kappa [L], Kappa-Mu [L], Kumaraswamy [L], Laplace [L], Linear Mean Residual Quantile Function [L], Normal [L], 3p log-Normal [L], Pearson Type III [L], Polynomial Density-Quantile 3 and 4 [L], Rayleigh [L], Rev-Gumbel [L+RC], Rice [L], Singh Maddala [L], Slash [TL], 3p Student t [L], Truncated Exponential [L], Wakeby [L], and Weibull [L].

Maintainer William Asquith <william.asquith@ttu.edu>

Repository CRAN

URL https://www.amazon.com/dp/1463508417/

NeedsCompilation no

LazyData true

Author William Asquith [aut, cre] (ORCID: <https://orcid.org/0000-0002-7400-1861>)

Date/Publication 2025-09-24 05:10:39 UTC

Contents

`lmomco-package` *L-moments, Censored L-moments, Trimmed L-moments, L-comoments, and Many Distributions*

Description

The **lmomco** package is a comparatively comprehensive implementation of L-moments in addition to probability-weighted moments, and parameter estimation for numerous familiar and not-so-familiar distributions. L-moments and their cousins are based on certain linear combinations of order statistic expectations. Being based on linear mathematics and thus especially robust compared to conventional moments, they are particular suitable for analysis of rare events of non-Normal data. L-moments are consistent and often have smaller sampling variances than maximum likelihood in small to moderate sample sizes. L-moments are especially useful in the context of quantile functions. The method of L-moments (lmr2par) is augmented here with access to the methods of maximum likelihood (mle2par) and maximum product of spacings (mps2par) as alternatives for parameter estimation bound into the distributions of the **lmomco** package.

About 370 user-level functions are implemented in **lmomco** that range from low-level utilities forming an application programming interface (API) to high-level sophisticated data analysis and visualization operators. The "See Also" section lists recommended function entry points for new users. The nomenclature (d, p, r, q)-lmomco is directly analogous to that for distributions built-in to R. To conclude, the R packages **lmom** (Hosking), **lmomRFA** (Hosking), **Lmoments** (Karvanen) might also be of great interest.

How does **lmomco** basically work? The design of **lmomco** is to fit distributions to the L-moments of sample data. Distributions are specified by a `type` argument for very many functions. The package stores both L-moments (see vec2lmom) and parameters (see vec2par) in simple R `list` structures—very elementary. The following code shows a comparison of parameter estimation for a random sample (rlmomco) of a GEV distribution using L-moments (lmoms coupled with lmom2par or simply lmr2par), maximum likelihood (MLE, mle2par), and maximum product of spacings (MPS, mps2par). (A note of warning, the MLE and MPS algorithms might not converge with the initial parameters—for purposes of "learning" about this package just rerun the code below again for another random sample.)

```
parent.lmoments <- vec2lmom(c(3.08, 0.568, -0.163)); ty <- "gev"
Q <- rlmomco(63, lmom2par(parent.lmoments, type=ty)) # random sample
init <- lmoms(Q); init$ratios[3] <- 0 # failure rates for mps and mle are
# substantially lowered if starting from the middle of the distribution's
# shape to form the initial parameters for init.para
lmr  <- lmr2par(Q, type=ty)                # method of L-moments
mle  <- mle2par(Q, type=ty, init.lmr=init) # method of MLE
mps  <- mps2par(Q, type=ty, init.lmr=init) # method of MPS
lmr1 <- lmr$para; mle1 <- mle$para; mps1 <- mps$para
```

The `lmr1`, `mle1`, and `mps1` variables each contain distribution parameter estimates, but before they are inspected, how about quick comparison to another R package (**eva**)?

```
lmr2 <- eva::gevrFit(Q, method="pwm")$par.ests # PWMs == L-moments
mle2 <- eva::gevrFit(Q, method="mle")$par.ests # method of MLE
mps2 <- eva::gevrFit(Q, method="mps")$par.ests # method of MPS
# Package eva uses a different sign convention on the GEV shape parameter
mle2[3] <- -mle2[3]; mps2[3] <- -mps2[3]; lmr2[3] <- -lmr2[3];
```

Now let us inspect the contents of the six estimates of the three GEV parameters by three different methods:

```
message("LMR(lmomco): ", paste(round(lmr1, digits=5), collapse="  "))
message("LMR(   eva): ", paste(round(lmr2, digits=5), collapse="  "))
message("MLE(lmomco): ", paste(round(mle1, digits=5), collapse="  "))
message("MLE(   eva): ", paste(round(mle2, digits=5), collapse="  "))
message("MPS(lmomco): ", paste(round(mps1, digits=5), collapse="  "))
message("MPS(   eva): ", paste(round(mps2, digits=5), collapse="  "))
```

The results show compatible estimates between the two packages. Lastly, let us plot what these distributions look like using the **lmomco** functions: add.lmomco.axis, nonexceeds, pp, and qlmomco.

```
par(las=2, mgp=c(3, 0.5, 0)); FF <- nonexceeds(); qFF <- qnorm(FF)
PP <- pp(Q); qPP <- qnorm(PP); Q <- sort(Q)
plot(  qFF, qlmomco(FF, lmr), xaxt="n", xlab="", tcl=0.5,
                              ylab="QUANTILE", type="l")
lines( qFF, qlmomco(FF, mle), col="blue")
lines( qFF, qlmomco(FF, mps), col="red" )
points(qPP, Q, lwd=0.6, cex=0.8, col=grey(0.3)); par(las=1)
add.lmomco.axis(las=2, tcl=0.5, side.type="NPP")
```

Author(s)

William Asquith <william.asquith@ttu.edu>

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496, doi:10.1016/j.csda.2006.07.016.

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8, https://www.amazon.com/dp/1463508417/.

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970, doi:10.1016/j.csda.2012.12.013.

Dey, D.K., Roy, Dooti, Yan, Jun, 2016, Univariate extreme value analysis, chapter 1, *in* Dey, D.K., and Yan, Jun, eds., Extreme value modeling and risk analysis—Methods and applications: Boca Raton, FL, CRC Press, pp. 1–22.

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational statistics and data analysis, vol. 43, pp. 299-314, doi:10.1016/S01679473(02)002505.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press, https://www.amazon.com/dp/0521019400/.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York, https://www.amazon.com/dp/0817683607/.

Serfling, R., and Xiao, P., 2007, A contribution to multivariate L-moments—L-comoment matrices: Journal of Multivariate Analysis, v. 98, pp. 1765–1781, doi:10.1016/j.jmva.2007.01.008.

See Also

lmoms, dlmomco, plmomco, rlmomco, qlmomco, lmom2par, plotlmrdia, lcomoms2

.lmomcohash *Storage of Lookup Tables for the lmomco Package*

Description

This is a hidden data object contained in the R/sysdata.rda file of the **lmomco** package. The system files inst/doc/SysDataBuilder01.R and SysDataBuilder02.R of the package are responsible for the construction of these data with the exception of the Eta-Mu and Kappa-Mu distribution content.

Format

An R environment with entries:

AEPkh2lmrTable A data.frame of asymmetric exponential power (4-parameter) relations between its two shape parameters, numerical, and theoretical L-skew and L-kurtosis. The table stems from inst/doc/SysDataBuilder01.R. (See also paraep4)

EMU_lmompara_byeta A data.frame of pre-computed table of relations between the parameters and L-moments of the Eta-Mu distribution. (See also lmomemu, paremu)

KMU_lmompara_bykappa A data.frame of pre-computed table of relations between the parameters and L-moments of the Kappa-Mu distribution. (See also lmomkmu, parkmu)

RiceTable A data.frame with coefficient of L-variation, signal to noise ratio, a parameter G, and L-skew and L-kurtosis of the Rice distribution. This is useful for quick parameter estimation. The table stems from inst/doc/SysDataBuilder01.R. (See also lmomrice, parrice)

RiceTable.maxLCV Maximum coefficient of L-variation representable (or apparently so) within R. The value stems from inst/doc/SysDataBuilder01.R.

RiceTable.minLCV Minimum coefficient of L-variation representable (or apparently so) within R. The value stems from inst/doc/SysDataBuilder01.R.

tau46list Various relations of Tau4-Tau6 for symmetrical distributions and used to support the access layer provided by lmrdia46 for Tau4-Tau6 L-moment ratio diagrams. The tables in the list stem from inst/doc/SysDataBuilder02.R, which is designed to be run after the SysDataBuilder01.R.

add.lmomco.axis *Add an lmomco Axis to a Plot*

Description

This function provides special support for adding probability-like axes to an existing plot. The function supports a recurrence interval (RI) axis, normal probability axis (NPP), and standard normal variate (SNV) axis. The function is built around the interface model that standard normal transformation of the values for the respective axis controlled by this function are being plotted; this means that qnorm() should be wrapped on the values of nonexceedance probability. This is an ease oversight to make (see Examples section below and note use of qnorm(pp)).

The function provides a convenient interface for labeling and titling two axes, so adjustments to default margins might be desired. The pertinent control is achieved using the `par()` function, which might be of the form `par(mgp=c(3,0.5,0), mar=c(5,4,4,3))` say for plotting the **lmomco** axis both on the left and right (see `z.par2cdf` for an example).

Usage

```
add.lmomco.axis(side=1, twoside=FALSE, twoside.suppress.labels=FALSE,
                side.type=c("NPP", "RI", "SNV"),
                otherside.type=c("NA", "RI", "SNV", "NPP"),
                alt.lab=NA, alt.other.lab=NA, npp.as.aep=FALSE,
                case=c("upper", "lower"),
                NPP.control=NULL, RI.control=NULL, SNV.control=NULL, ...)
```

Arguments

`side`	The side of the plot (1=bottom, 2=left, 3=top, 4=right).
`twoside`	A logical triggering whether the tick marks are echoed on the opposite side. This value is forced to `FALSE` if `otherside.type` is not `"NA"`.
`twoside.suppress.labels`	A logical to turn off labeling on the opposite side. This is useful if only the ticks (major and minor) are desired.
`side.type`	The axis type for the primary `side`.
`otherside.type`	The optional axis type for the opposite side. The default is a literal not applicable.
`alt.lab`	A short-cut to change the axis label without having to specify a `*.control` argument and its `label` attribute. The label attribute of `alt.lab` is not `NA` is used instead of the defaults. This argument overrides behavior of the `otherside.type` labeling so use of `alt.lab` only makes sense if `otherside.type` is left as `NA`.
`alt.other.lab`	Similar to `alt.lab` but can house an alternative label (see **Examples**.
`npp.as.aep`	Convert nonexceedance probability to exceedance probability, which is a que for `alt.other.lab` and nonexceedance probabilities are changed by $1 - F$, but the real coordinates for plotting remain in the nonexceedance probability context.
`case`	The will switch between all upper case or mixed case for the default labels.
`NPP.control`	An optional R `list` used to influence the NPP axis.
`RI.control`	An optional R `list` used to influence the RI axis.
`SNV.control`	An optional R `list` used to influence the SNV axis.
`...`	Additional arguments that are passed to the R function `Axis`.

Value

No value is returned. This function is used for its side effects.

Note

The `NPP.control`, `RI.control`, and `SNV.control` are R `list` structures that can be populated (and perhaps someday extended) to feed various settings into the respective axis types. In brief:

The `NPP.control` provides

`label`	The title for the NPP axis—be careful with value of `as.exceed`.
`probs`	A vector of nonexceedance probabilities F.
`probs.lab`	A vector of nonexceedance probabilities F to label.
`digits`	The digits for the R function `format` to enhance appearance.
`line`	The line for the R function `mtext` to place `label`.
`as.exceed`	A logical triggering $S = 1 - F$.

The `RI.control` provides

`label`	The title for the RI axis.
`Tyear`	A vector of T-year recurrence intervals.
`line`	The line for the R function `mtext` to place `label`.

The `SNV.control` provides

`label`	The title for the SNV axis.
`begin`	The beginning "number of standard deviations".
`end`	The ending "number of standard deviations".
`by`	The step between `begin` and `end`.
`line`	The line for the R function `mtext` to place `label`.

The user is responsible for appropriate construction of the `control` lists. Very little error trapping is made to keep the code base tight. The defaults when the function definition are likely good for many types of applications. Lastly, the manipulation of the `mgp` parameter in the example is to show how to handle the offset between the numbers and the ticks when the ticks are moved to pointing inward, which is opposite of the default in R.

Author(s)

W.H. Asquith

See Also

`prob2T`, `T2prob`, `add.log.axis`

Examples

```
par(mgp=c(3,0.5,0)) # going to tick to the inside, change some parameters
X <- sort(rnorm(65)); pp <- pp(X) # generate synthetic data
plot(qnorm(pp), X, xaxt="n", xlab="", ylab="QUANTILE", xlim=c(-2,3))
add.lmomco.axis(las=2, tcl=0.5, side.type="RI", otherside.type="NPP")
par(mgp=c(3,1,0)) # restore defaults
```

```
## Not run:
opts <- options(scipen=6); par(mgp=c(3,0.5,0))
X <- sort(rexp(65, rate=.0001))*100; pp <- pp(X) # generate synthetic data
plot(qnorm(pp), X, yaxt="n", xaxt="n", xlab="", ylab="", log="y")
add.log.axis(side=2,    tcl=+0.8*abs(par()$tcl),         two.sided=TRUE)
add.log.axis(logs=c(1), tcl=-0.5*abs(par()$tcl), side=2, two.sided=TRUE)
add.log.axis(logs=c(1), tcl=+1.3*abs(par()$tcl), side=2, two.sided=TRUE)
add.log.axis(logs=1:8, side=2, make.labs=TRUE, las=1, label="QUANTILE")
add.lmomco.axis(las=2, tcl=0.5, side.type="NPP", npp.as.aep=TRUE, case="lower")
options(opts)
par(mgp=c(3,1,0)) # restore defaults
## End(Not run)
```

`add.log.axis` *Add a Polished Logarthimic Axis to a Plot*

Description

This function provides special support for adding superior looking base-10 logarithmic axes relative to **R**'s defaults, which are an embarassment. The **Examples** section shows an overly elaborate version made by repeated calls to this function with a drawback that each call redraws the line of the axis so deletion in editing software might be required. This function is indexed under the "lmomco functions" because of its relation to `add.lmomco.axis` and is not named `add.lmomcolog.axis` because such a name is too cumbersome.

Usage

```
add.log.axis(make.labs=FALSE, logs=c(2, 3, 4, 5, 6, 7, 8, 9), side=1,
             two.sided=FALSE, label=NULL, x=NULL, col.ticks=1, ...)
```

Arguments

`make.labs`	A logical controlling whether the axis is labled according to the values in `logs`.
`logs`	A numeric vector of log-cycles for which ticking and (or) labeling is made. These are normalized to the first log-cycle, so a value of 3 would spawn values such as $\cdots, 0.03, 0.3, 3, 30, \cdots$ through a range exceeding the axis limits. The default anticipates that a second call to the function will be used to make longer ticks at the even log-cycles; hence, the value 1 is not in the default vector. The **Examples** section provides a thorough demonstration.
`side`	An integer specifying which side of the plot the axis is to be drawn on, and argument corresponds the axis side argument of the `axis` function. The axis is placed as follows: 1=below, 2=left, 3=above, and 4=right.
`two.sided`	A logical controlling whether the side oppose of `side` also is to be drawn.
`label`	The label (title) of the axis, which is placed by a call to function `mtext`, and thus either the `xlab` or `ylab` arguments for `plot` should be set to the empty string `""`.

x	This is an optional data vector (untransformed!), which will compute nice axis limits and return them. These limits will align with (snap to) the integers within a log10-cycle.
col.ticks	This is the same argument as the axis function.
...	Additional arguments to pass to axis.

Value

No value is returned, except if argument x is given, for which nice axis limits are returned. By overall design, this function is used for its side effects.

Author(s)

W.H. Asquith

See Also

add.lmomco.axis

Examples

```
## Not run:
par(mgp=c(3,0.5,0)) # going to tick to the inside, change some parameters
X <- 10^sort(rnorm(65)); pp <- pp(X) # generate synthetic data
ylim <- add.log.axis(x=X) # snap to some nice integers within the cycle
plot(qnorm(pp), X, xaxt="n", yaxt="n", xlab="", ylab="", log="y",
     xlim=c(-2,3), ylim=ylim, pch=6, yaxs="i", col=4)
add.lmomco.axis(las=2, tcl=0.5, side.type="RI", otherside.type="NPP")
# Logarithmic axis: the base ticks to show logarithms
add.log.axis(side=2,      tcl=0.8*abs(par()$tcl), two.sided=TRUE)
#                    the long even-cycle tick, set to inside and outside
add.log.axis(logs=c(1),   tcl=-0.5*abs(par()$tcl), side=2, two.sided=TRUE)
add.log.axis(logs=c(1),   tcl=+1.3*abs(par()$tcl), side=2, two.sided=TRUE)
#                    now a micro tick at the 1.5 logs but only on the right
add.log.axis(logs=c(1.5), tcl=+0.5*abs(par()$tcl), side=4)
#                    and only label the micro tick at 1.5 on the right
add.log.axis(logs=c(1.5), side=4, make.labs=TRUE, las=3) # but weird rotate
#                    add the bulk tick labeling and axis label.
add.log.axis(logs=c(1, 2, 3, 4, 6), side=2, make.labs=TRUE, las=1, label="QUANTILE")
par(mgp=c(3,1,0)) # restore defaults
## End(Not run)
```

---

amarilloprecip *Annual Maximum Precipitation Data for Amarillo, Texas*

Description

Annual maximum precipitation data for Amarillo, Texas

Usage

```
data(amarilloprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(amarilloprecip)
summary(amarilloprecip)
```

`Apwm2BpwmRC` *Conversion between A- and B-Type Probability-Weighted Moments for Right-Tail Censoring of an Appropriate Distribution*

Description

This function converts "A"-type probability-weighted moments (PWMs, β_r^A) to the "B"-type β_r^B. The β_r^A are the ordinary PWMs for the m left noncensored or observed values. The β_r^B are more complex and use the m observed values and the $m - n$ right-tailed censored values for which the censoring threshold is known. The "A"- and "B"-type PWMs are described in the documentation for `pwmRC`.

This function uses the defined relation between to two PWM types when the β_r^A are known along with the parameters (`para`) of a right-tail censored distribution inclusive of the censoring fraction $\zeta = m/n$. The value ζ is the right-tail censor fraction or the probability $\mathrm{Pr}\{\}$ that x is less than the quantile at ζ nonexceedance probability ($\mathrm{Pr}\{x < X(\zeta)\}$). The relation is

$$\beta_{r-1}^B = r^{-1}\{\zeta^r r\beta_{r-1}^A + (1-\zeta^r)X(\zeta)\},$$

where $1 \leq r \leq n$ and n is the number of moments, and $X(\zeta)$ is the value of the quantile function at nonexceedance probability ζ. Finally, the `RC` in the function name is to denote `Right`-tail `Censoring`.

Usage

```
Apwm2BpwmRC(Apwm,para)
```

Arguments

Apwm
: A vector of A-type PWMs: β_r^A.

para
: The parameters of the distribution from a function such as pargpaRC in which the β_r^A are contained in a list element titled betas and the right-tail censoring fraction ζ is contained in an element titled zeta.

Value

An R list is returned.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

Bpwm2ApwmRC, pwmRC

Examples

```
# Data listed in Hosking (1995, table 29.2, p. 551)
H <- c(3,4,5,6,6,7,8,8,9,9,9,10,10,11,11,11,13,13,13,13,13,
             17,19,19,25,29,33,42,42,51.9999,52,52,52)
      # 51.9999 was really 52, a real (noncensored) data point.
z <-  pwmRC(H,52)
# The B-type PMWs are used for the parameter estimation of the
# Reverse Gumbel distribution. The parameter estimator requires
# conversion of the PWMs to L-moments by pwm2lmom().
para <- parrevgum(pwm2lmom(z$Bbetas),z$zeta) # parameter object
Bbetas <- Apwm2BpwmRC(z$Abetas,para)
Abetas <- Bpwm2ApwmRC(Bbetas$betas,para)
# Assertion that both of the vectors of B-type PWMs should be the same.
str(Abetas)   # A-type PWMs of the distribution
str(z$Abetas) # A-type PWMs of the original data
```

`are.lmom.valid` *Are the L-moments valid*

Description

The L-moments have particular constraints on magnitudes and relation to each other. This function evaluates and L-moment object whether the bounds for $\lambda_2 > 0$ (L-scale), $|\tau_3| < 1$ (L-skew), $\tau_4 < 1$ (L-kurtosis), and $|\tau_5| < 1$ are satisfied. An optional check on $\tau_4 \geq (5\tau_3^2 - 1)/4$ is made. Also for further protection, the finitenesses of the mean (λ_1) and λ_2 are also checked. These checks provide protection against say L-moments being computed on the logarithms of some data but the data themselves have values less than or equal to zero.

The TL-moments as implemented by the `TL` functions (`TLmoms`) are not applicable to the boundaries (well finiteness of course). The `are.lmom.valid` function should not be consulted on the TL-moments.

Usage

```
are.lmom.valid(lmom, checkt3t4=TRUE)
```

Arguments

`lmom`	An L-moment object created by `lmoms`, `lmom.ub`, `pwm2lmom`; and
`checkt3t4`	A logical triggering the above test on L-skew to L-kurtosis. This bounds in very small samples can be violated—usually the user will want this set and until (first release in 2017, v2.2.6) this bounds check was standard in **lmomco** for over a decade.

Value

`TRUE`	L-moments are valid.
`FALSE`	L-moments are not valid.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmom.ub`, `lmoms`, `pwm2lmom`

Examples

```
lmr <- lmoms(rnorm(20))
if(are.lmom.valid(lmr)) print("They are.")
## Not run:
X <- c(1.7106278,  1.7598761,  1.2111335,  0.3447490,  1.8312889,
       1.3938445, -0.5376054, -0.2341009, -0.4333601, -0.2545229)
are.lmom.valid(lmoms(X))
are.lmom.valid(pwm2lmom(pwm.pp(X, a=0.5)))

# Prior to version 2.2.6, the next line could leak through as TRUE. This was a problem.
# Nonfiniteness of the mean or L-scale should have been checked; they are for v2.2.6+
are.lmom.valid(lmoms(log10(c(1,23,235,652,0)), nmom=1)) # of other nmom

## End(Not run)
```

---

are.par.valid *Are the Distribution Parameters Consistent with the Distribution*

Description

This function is a dispatcher on top of the `are.parCCC.valid` functions, where `CCC` represents the distribution type: `aep4`, `cau`, `emu`, `exp`, `gam`, `gep`, `gev`, `glo`, `gno`, `gov`, `gpa`, `gum`, `kap`, `kmu`, `kur`, `lap`, `ln3`, `nor`, `pe3`, `ray`, `revgum`, `rice`, `sla`, `smd`, `st3`, `texp`, `tri`, `wak`, or `wei`. For **lmomco** functionality, `are.par.valid` is called only by `vec2par` in the process of converting a vector into a proper distribution parameter object.

Usage

```
are.par.valid(para, paracheck=TRUE, ...)
```

Arguments

`para`	A distribution parameter object having at least attributes `type` and `para`.
`paracheck`	A logical controlling whether the parameters are checked for validity and if `paracheck=TRUE` then effectively this whole function becomes turned off.
`...`	Additional arguments for the `are.parCCC.valid` call that is made internally.

Value

`TRUE`	If the parameters are consistent with the distribution specified by the `type` attribute of the parameter object.
`FALSE`	If the parameters are not consistent with the distribution specified by the `type` attribute of the parameter object.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

vec2par, dist.list

Examples

```
vec  <- c(12, 120)           # parameters of exponential distribution
para <- vec2par(vec, "exp")  # build exponential distribution parameter
                             # object
# The following two conditionals are equivalent as are.parexp.valid()
# is called within are.par.valid().
if(   are.par.valid(para)) Q <- quaexp(0.5, para)
if(are.parexp.valid(para)) Q <- quaexp(0.5, para)
```

---

are.paraep4.valid *Are the Distribution Parameters Consistent with the 4-Parameter Asymmetric Exponential Power Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfaep4, pdfaep4, quaaep4, and lmomaep4) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.paraep4.valid function.

Usage

```
are.paraep4.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by paraep4 or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are aep4 consistent.
FALSE	If the parameters are not aep4 consistent.

Note

This function calls `is.aep4` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Delicado, P., and Goria, M.N., 2008, A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution: Computational Statistics and Data Analysis, v. 52, no. 3, pp. 1661–1673.

See Also

`is.aep4`, `paraep4`

Examples

```
para <- vec2par(c(0,1, 0.5, 4), type="aep4")
if(are.paraep4.valid(para)) Q <- quaaep4(0.5,para)
```

`are.parcau.valid` *Are the Distribution Parameters Consistent with the Cauchy Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfcau`, `pdfcau`, `quacau`, and `lmomcau`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parcau.valid` function.

Usage

```
are.parcau.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `parcau` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are cau consistent.
FALSE	If the parameters are not cau consistent.

Note

This function calls is.cau to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

Gilchrist, W.G., 2000, Statistical modeling with quantile functions: Chapman and Hall/CRC, Boca Raton, FL.

See Also

is.cau, parcau

Examples

```
para <- vec2par(c(12,12),type='cau')
if(are.parcau.valid(para)) Q <- quacau(0.5,para)
```

---

are.paremu.valid — *Are the Distribution Parameters Consistent with the Eta-Mu Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfemu, pdfemu, quaemu, lmomemu), and lmomemu require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.paremu.valid function. The documentation for pdfemu provides the conditions for valid parameters.

Usage

```
are.paremu.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by paremu or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are emu consistent.
FALSE	If the parameters are not emu consistent.

Note

This function calls is.emu to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

See Also

is.emu, paremu

Examples

```
## Not run:
para <- vec2par(c(0.4, .04), type="emu")
if(are.paremu.valid(para)) Q <- quaemu(0.5,para) #
## End(Not run)
```

---

are.parexp.valid — *Are the Distribution Parameters Consistent with the Exponential Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfexp, pdfexp, quaexp, and lmomexp) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parexp.valid function.

Usage

```
are.parexp.valid(para, nowarn=FALSE)
```

Arguments

para
: A distribution parameter list returned by parexp.

nowarn
: A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE
: If the parameters are exp consistent.

FALSE
: If the parameters are not exp consistent.

Note

This function calls is.exp to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.exp, parexp

Examples

```
para <- parexp(lmoms(c(123,34,4,654,37,78)))
if(are.parexp.valid(para)) Q <- quaexp(0.5,para)
```

---

are.pargam.valid *Are the Distribution Parameters Consistent with the Gamma Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgam, pdfgam, quagam, and lmomgam) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargam.valid function. The parameters are restricted to the following conditions.

$$\alpha > 0 \text{ and } \beta > 0.$$

Alternatively, a three-parameter version is available following the parameterization of the Generalized Gamma distribution used in the **gamlss.dist** package and and for **lmomco** is documented under pdfgam. The parameters for this version are

$$\mu > 0;\quad \sigma > 0;\quad -\infty < \nu < \infty$$

for parameters number 1, 2, 3, respectively.

Usage

```
are.pargam.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by pargam or vec2par.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `gam` consistent.
`FALSE`	If the parameters are not `gam` consistent.

Note

This function calls is.gam to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.gam, pargam

Examples

```
para <- pargam(lmoms(c(123,34,4,654,37,78)))
if(are.pargam.valid(para)) Q <- quagam(0.5,para)
```

---

are.pargdd.valid — *Are the Distribution Parameters Consistent with the Gamma Difference Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgdd, pdfgdd, quagdd, and lmomgdd) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargdd.valid function.

Usage

```
are.pargdd.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargdd or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are gdd consistent.
FALSE	If the parameters are not gdd consistent.

Note

This function calls is.gdd to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Klar, B., 2015, A note on gamma difference distributions: Journal of Statistical Computation and Simulation v. 85, no. 18, pp. 1–8, doi:10.1080/00949655.2014.996566.

See Also

`is.gdd`, `pargdd`

Examples

```
#
```

---

are.pargep.valid — *Are the Distribution Parameters Consistent with the Generalized Exponential Poisson Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfgep`, `pdfgep`, `quagep`, and `lmomgep`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.pargep.valid` function. The parameters must be $\beta > 0$, $\kappa > 0$, and $h > 0$.

Usage

```
are.pargep.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `pargep` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are gep consistent.
`FALSE`	If the parameters are not gep consistent.

Note

This function calls `is.gep` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Barreto-Souza, W., and Cribari-Neto, F., 2009, A generalization of the exponential-Poisson distribution: Statistics and Probability, 79, pp. 2493–2500.

See Also

is.gep, pargep

Examples

```
#para <- pargep(lmoms(c(123,34,4,654,37,78)))
#if(are.pargep.valid(para)) Q <- quagep(0.5,para)
```

---

are.pargev.valid *Are the Distribution Parameters Consistent with the Generalized Extreme Value Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgev, pdfgev, quagev, and lmomgev) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargev.valid function.

Usage

```
are.pargev.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargev or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are gev consistent.
FALSE	If the parameters are not gev consistent.

Note

This function calls is.gev to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.gev, pargev

Examples

```
para <- pargev(lmoms(c(123, 34, 4, 654, 37, 78)))
if(are.pargev.valid(para)) Q <- quagev(0.5, para)
```

---

are.pargld.valid — *Are the Distribution Parameters Consistent with the Generalized Lambda Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgld, pdfgld, quagld, and lmomgld) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargld.valid function.

Usage

```
are.pargld.valid(para, verbose=FALSE, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargld or vec2par.
verbose	A logical switch on additional output to the user—default is FALSE.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Details

Karian and Dudewicz (2000) outline valid parameter space of the Generalized Lambda distribution. First, according to Theorem 1.3.3 the distribution is valid if and only if

$$\alpha(\kappa F^{\kappa-1} + h(1-F)^{h-1}) \geq 0.$$

for all $F \in [0, 1]$. The `are.pargld.valid` function tests against this condition by incrementing through $[0, 1]$ by $dF = 0.0001$. This is a brute force method of course. Further, Karian and Dudewicz (2002) provide a diagrammatic representation of regions in κ and h space for suitable α in which the distribution is valid. The `are.pargld.valid` function subsequently checks against the 6 valid regions as a secondary check on Theorem 1.3.3. The regions of the distribution are defined for suitably choosen α by

Region 1: $\kappa \leq -1$ and $h \geq 1$,

Region 2: $\kappa \geq 1$ and $h \leq -1$,

Region 3: $\kappa \geq 0$ and $h \geq 0$,

Region 4: $\kappa \leq 0$ and $h \leq 0$,

Region 5: $h \geq (-1/\kappa)$ and $-1 \geq \kappa \leq 0$, and

Region 6: $h \leq (-1/\kappa)$ and $h \geq -1$ and $\kappa \geq 1$.

Value

`TRUE`	If the parameters are `gld` consistent.
`FALSE`	If the parameters are not `gld` consistent.

Note

This function calls `is.gld` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

`is.gld`, `pargld`

Examples

```
## Not run:
para <- vec2par(c(123,34,4,3),type='gld')
if(are.pargld.valid(para)) Q <- quagld(0.5,para)

# The following is an example of inconsistent L-moments for fitting but
# prior to lmomco version 2.1.2 and untrapped error was occurring.
lmr <- lmoms(c(33, 37, 41, 54, 78, 91, 100, 120, 124))
para <- pargld(lmr); are.pargld.valid(para)
## End(Not run)
```

---

are.parglo.valid — *Are the Distribution Parameters Consistent with the Generalized Logistic Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfglo`, `pdfglo`, `quaglo`, and `lmomglo`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parglo.valid` function.

Usage

```
are.parglo.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `parglo` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `glo` consistent.
`FALSE`	If the parameters are not `glo` consistent.

Note

This function calls `is.glo` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.glo, parglo

Examples

```
para <- parglo(lmoms(c(123,34,4,654,37,78)))
if(are.parglo.valid(para)) Q <- quaglo(0.5,para)
```

---

are.pargno.valid — *Are the Distribution Parameters Consistent with the Generalized Normal Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgno, pdfgno, quagno, and lmomgno) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargno.valid function.

Usage

```
are.pargno.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargno or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are gno consistent.
FALSE	If the parameters are not gno consistent.

Note

This function calls is.gno to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.gno, pargno

Examples

```
para <- pargno(lmoms(c(123,34,4,654,37,78)))
if(are.pargno.valid(para)) Q <- quagno(0.5,para)
```

---

are.pargov.valid — *Are the Distribution Parameters Consistent with the Govindarajulu Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgov, pdfgov, quagov, and lmomgov) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargov.valid function.

Usage

```
are.pargov.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargov or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are gov consistent.
FALSE	If the parameters are not gov consistent.

Note

This function calls is.gov to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

is.gov, pargov

Examples

```
para <- pargov(lmoms(c(123,34,4,654,37,78)))
if(are.pargov.valid(para)) Q <- quagov(0.5,para)
```

---

are.pargpa.valid *Are the Distribution Parameters Consistent with the Generalized Pareto Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgpa, pdfgpa, quagpa, and lmomgpa) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargpa.valid function.

Usage

```
are.pargpa.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargpa or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are gpa consistent.
FALSE	If the parameters are not gpa consistent.

Note

This function calls is.gpa to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.gpa, pargpa

Examples

```
para <- pargpa(lmoms(c(123,34,4,654,37,78)))
if(are.pargpa.valid(para)) Q <- quagpa(0.5,para)
```

---

are.pargum.valid — *Are the Distribution Parameters Consistent with the Gumbel Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfgum, pdfgum, quagum, and lmomgum) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.pargum.valid function.

Usage

```
are.pargum.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by pargum or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are gum consistent.
FALSE	If the parameters are not gum consistent.

Note

This function calls is.gum to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.gum, pargum

Examples

```
para <- pargum(lmoms(c(123,34,4,654,37,78)))
if(are.pargum.valid(para)) Q <- quagum(0.5,para)
```

---

are.parkap.valid — *Are the Distribution Parameters Consistent with the Kappa Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfkap, pdfkap, quakap, and lmomkap) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parkap.valid function.

Usage

```
are.parkap.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parkap or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are kap consistent.
FALSE	If the parameters are not kap consistent.

Note

This function calls is.kap to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1994, The four-parameter kappa distribution: IBM Journal of Reserach and Development, v. 38, no. 3, pp. 251–258.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.kap, parkap

Examples

```
para <- parkap(lmoms(c(123,34,4,654,37,78)))
if(are.parkap.valid(para)) Q <- quakap(0.5,para)
```

`are.parkmu.valid` *Are the Distribution Parameters Consistent with the Kappa-Mu Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`pdfkmu`, `cdfkmu`, `quakmu`, and `lmomkmu`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parkmu.valid` function. The documentation for `pdfkmu` provides the conditions for valid parameters.

Usage

```
are.parkmu.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `parkmu` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `kmu` consistent.
`FALSE`	If the parameters are not `kmu` consistent.

Note

This function calls `is.kmu` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

See Also

`is.kmu`, `parkmu`

Examples

```
para <- vec2par(c(0.5, 1.5), type="kmu")
if(are.parkmu.valid(para)) Q <- quakmu(0.5,para)
```

are.parkur.valid *Are the Distribution Parameters Consistent with the Kumaraswamy Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfkur, pdfkur, quakur, and lmomkur) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parkur.valid function.

Usage

```
are.parkur.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parkur or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are kur consistent.
FALSE	If the parameters are not kur consistent.

Note

This function calls is.kur to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Jones, M.C., 2009, Kumaraswamy's distribution—A beta-type distribution with some tractability advantages: Statistical Methodology, v. 6, pp. 70–81.

See Also

is.kur, parkur

Examples

```
para <- parkur(lmoms(c(0.25, 0.4, 0.6, 0.65, 0.67, 0.9)))
if(are.parkur.valid(para)) Q <- quakur(0.5,para)
```

`are.parlap.valid` *Are the Distribution Parameters Consistent with the Laplace Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdflap`, `pdflap`, `qualap`, and `lmomlap`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parlap.valid` function.

Usage

```
are.parlap.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `parlap` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `lap` consistent.
`FALSE`	If the parameters are not `lap` consistent.

Note

This function calls `is.lap` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: IBM Research Report RC12210, T.J. Watson Research Center, Yorktown Heights, New York.

See Also

is.lap, parlap

Examples

```
para <- parlap(lmoms(c(123,34,4,654,37,78)))
if(are.parlap.valid(para)) Q <- qualap(0.5,para)
```

---

are.parlmrq.valid   *Are the Distribution Parameters Consistent with the Linear Mean Residual Quantile Function Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdflmrq, pdflmrq, qualmrq, and lmomlmrq) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parlmrq.valid function. The constraints on the parameters are listed under qualmrq. The documentation for qualmrq provides the conditions for valid parameters.

Usage

```
are.parlmrq.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parlmrq or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are lmrq consistent.
FALSE	If the parameters are not lmrq consistent.

Note

This function calls is.lmrq to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

See Also

is.lmrq, parlmrq

Examples

```
para <- parlmrq(lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2)))
if(are.parlmrq.valid(para)) Q <- qualmrq(0.5,para)
```

---

are.parln3.valid — *Are the Distribution Parameters Consistent with the 3-Parameter Log-Normal Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfln3, pdfln3, qualn3, and lmomln3) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parln3.valid function.

Usage

```
are.parln3.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parln3 or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are ln3 consistent.
FALSE	If the parameters are not ln3 consistent.

Note

This function calls is.ln3 to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

is.ln3, parln3

Examples

```
para <- parln3(lmoms(c(123,34,4,654,37,78)))
if(are.parln3.valid(para)) Q <- qualn3(0.5,para)
```

---

are.parnor.valid — *Are the Distribution Parameters Consistent with the Normal Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfnor, pdfnor, quanor, and lmomnor) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parnor.valid function.

Usage

```
are.parnor.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parnor or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are nor consistent.
FALSE	If the parameters are not nor consistent.

Note

This function calls is.nor to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.nor, parnor

Examples

```
para <- parnor(lmoms(c(123,34,4,654,37,78)))
if(are.parnor.valid(para)) Q <- quanor(0.5,para)
```

---

are.parpdq3.valid *Are the Distribution Parameters Consistent with the Polynomial Density-Quantile#*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfpdq3, pdfpdq3, quapdq3, and lmompdq3) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parpdq3.valid function.

Usage

```
are.parpdq3.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parpdq3 or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are pdq3 consistent.
FALSE	If the parameters are not pdq3 consistent.

Note

This function calls `is.pdq3` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

See Also

`is.pdq3`, `parpdq3`

Examples

```
para <- parpdq3(lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52)))
if(are.parpdq3.valid(para)) Q <- quapdq3(0.5, para)
```

`are.parpdq4.valid` *Are the Distribution Parameters Consistent with the Polynomial Density-Quantile4*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfpdq4`, `pdfpdq4`, `quapdq4`, and `lmompdq4`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parpdq4.valid` function.

Usage

```
are.parpdq4.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by `parpdq4` or `vec2par`.
nowarn	A logical switch on warning suppression. If TRUE then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are pdq4 consistent.
FALSE	If the parameters are not pdq4 consistent.

Note

This function calls `is.pdq4` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

See Also

`is.pdq4`, `parpdq4`

Examples

```
para <- parpdq4(lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52)))
if(are.parpdq4.valid(para)) Q <- quapdq4(0.5, para)
```

`are.parpe3.valid` *Are the Distribution Parameters Consistent with the Pearson Type III Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfpe3`, `pdfpe3`, `quape3`, and `lmompe3`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parpe3.valid` function.

Usage

```
are.parpe3.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by `parpe3` or `vec2par`.
nowarn	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are pe3 consistent.
FALSE	If the parameters are not pe3 consistent.

Note

This function calls is.pe3 to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.pe3, parpe3

Examples

```
para <- parpe3(lmoms(c(123,34,4,654,37,78)))
if(are.parpe3.valid(para)) Q <- quape3(0.5,para)
```

---

are.parray.valid — *Are the Distribution Parameters Consistent with the Rayleigh Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfray, pdfray, quaray, and lmomray) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parray.valid function.

Usage

```
are.parray.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by `parray` or `vec2par`.
nowarn	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are `ray` consistent.
FALSE	If the parameters are not `ray` consistent.

Note

This function calls `is.ray` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

See Also

`is.ray`, `parray`

Examples

```
para <- parray(lmoms(c(123,34,4,654,37,78)))
if(are.parray.valid(para)) Q <- quaray(0.5,para)
```

`are.parrevgum.valid` *Are the Distribution Parameters Consistent with the Reverse Gumbel Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfrevgum`, `pdfrevgum`, `quarevgum`, and `lmomrevgum`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parrevgum.valid` function.

Usage

```
are.parrevgum.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by `parrevgum` or `vec2par`.
nowarn	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are `revgum` consistent.
FALSE	If the parameters are not `revgum` consistent.

Note

This function calls `is.revgum` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

`is.revgum`, `parrevgum`

Examples

```
para <- vec2par(c(.9252, .1636, .7),type='revgum')
if(are.parrevgum.valid(para)) Q <- quarevgum(0.5,para)
```

are.parrice.valid *Are the Distribution Parameters Consistent with the Rice Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfrice, pdfrice, quarice, and lmomrice) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parrice.valid function.

Usage

```
are.parrice.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parrice or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are rice consistent.
FALSE	If the parameters are not rice consistent.

Note

This function calls is.rice to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

is.rice, parrice

Examples

```
#para <- parrice(lmoms(c(123,34,4,654,37,78)))
#if(are.parrice.valid(para)) Q <- quarice(0.5,para)
```

are.parsla.valid — *Are the Distribution Parameters Consistent with the Slash Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfsla`, `pdfsla`, `quasla`, and `lmomsla`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parsla.valid` function.

Usage

```
are.parsla.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `parsla` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `sla` consistent.
`FALSE`	If the parameters are not `sla` consistent.

Note

This function calls `is.sla` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Rogers, W.H., and Tukey, J.W., 1972, Understanding some long-tailed symmetrical distributions: Statistica Neerlandica, v. 26, no. 3, pp. 211–226.

See Also

`is.sla`, `parsla`

Examples

```
para <- vec2par(c(12,1.2),type='sla')
if(are.parsla.valid(para)) Q <- quasla(0.5,para)
```

are.parsmd.valid *Are the Distribution Parameters Consistent with the Singh–Maddala Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfsmd, pdfsmd, quasmd, and lmomsmd) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parsmd.valid` function. The parameter constraints are simple $a > 0$ (scale), $b > 0$ (shape), and $q > 0$ (shape).

Usage

```
are.parsmd.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by parsmd or vec2par.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `smd` consistent.
`FALSE`	If the parameters are not `smd` consistent.

Note

This function calls is.smd to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Shahzad, M.N., and Zahid, A., 2013, Parameter estimation of Singh Maddala distribution by moments: International Journal of Advanced Statistics and Probability, v. 1, no. 3, pp. 121–131, doi:10.14419/ijasp.v1i3.1206.

See Also

is.smd, parsmd

Examples

```
#para <- parsmd(lmoms(c(123, 34, 4, 654, 37, 78)))
#if(are.parsmd.valid(para)) Q <- quasmd(0.5, para)
```

`are.parst3.valid` *Are the Distribution Parameters Consistent with the 3-Parameter Student t Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdfst3`, `pdfst3`, `quast3`, and `lmomst3`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.parst3.valid` function.

Usage

```
are.parst3.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `parst3` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `st3` consistent.
`FALSE`	If the parameters are not `st3` consistent.

Note

This function calls `is.st3` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

is.st3, parst3

Examples

```
para <- parst3(lmoms(c(90,134,100,114,177,378)))
if(are.parst3.valid(para)) Q <- quast3(0.5,para)
```

---

are.partexp.valid *Are the Distribution Parameters Consistent with the Truncated Exponential Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdftexp, pdftexp, quatexp, and lmomtexp) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.partexp.valid function.

Usage

```
are.partexp.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parexp or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are texp consistent.
FALSE	If the parameters are not texp consistent.

Note

This function calls is.texp to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Vogel, R.M., Hosking, J.R.M., Elphick, C.S., Roberts, D.L., and Reed, J.M., 2008, Goodness of fit of probability distributions for sightings as species approach extinction: Bulletin of Mathematical Biology, DOI 10.1007/s11538-008-9377-3, 19 p.

See Also

`is.texp`, `partexp`

Examples

```
para <- partexp(lmoms(c(90,134,100,114,177,378)))
if(are.partexp.valid(para)) Q <- quatexp(0.5,para)
```

---

are.partri.valid *Are the Distribution Parameters Consistent with the Asymmetric Triangular Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (`cdftri`, `pdftri`, `quatri`, and `lmomtri`) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the `are.partri.valid` function.

Usage

```
are.partri.valid(para, nowarn=FALSE)
```

Arguments

`para`	A distribution parameter list returned by `partri` or `vec2par`.
`nowarn`	A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

`TRUE`	If the parameters are `tri` consistent.
`FALSE`	If the parameters are not `tri` consistent.

Note

This function calls `is.tri` to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

See Also

is.tri, partri

Examples

```
para <- partri(lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52)))
if(are.partri.valid(para)) Q <- quatri(0.5,para)
```

---

are.parwak.valid — *Are the Distribution Parameters Consistent with the Wakeby Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfwak, pdfwak, quawak, and lmomwak) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parwak.valid function.

Usage

```
are.parwak.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parwak or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are wak consistent.
FALSE	If the parameters are not wak consistent.

Note

This function calls is.wak to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.wak, parwak

Examples

```
para <- parwak(lmoms(c(123,34,4,654,37,78)))
if(are.parwak.valid(para)) Q <- quawak(0.5,para)
```

`are.parwei.valid` *Are the Distribution Parameters Consistent with the Weibull Distribution*

Description

Is the distribution parameter object consistent with the corresponding distribution? The distribution functions (cdfwei, pdfwei, quawei, and lmomwei) require consistent parameters to return the cumulative probability (nonexceedance), density, quantile, and L-moments of the distribution, respectively. These functions internally use the are.parwei.valid function.

Usage

```
are.parwei.valid(para, nowarn=FALSE)
```

Arguments

para	A distribution parameter list returned by parwei or vec2par.
nowarn	A logical switch on warning suppression. If TRUE then options(warn=-1) is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.

Value

TRUE	If the parameters are wei consistent.
FALSE	If the parameters are not wei consistent.

Note

This function calls is.wei to verify consistency between the distribution parameter object and the intent of the user.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

is.wei, parwei

Examples

```
para <- parwei(lmoms(c(123,34,4,654,37,78)))
if(are.parwei.valid(para)) Q <- quawei(0.5,para)
```

BEhypergeo *Barnes Extended Hypergeometric Function*

Description

This function computes the Barnes Extended Hypergeometric function, which in **lmomco** is useful in applications involving expectations of order statistics for the Generalized Exponential Poisson (GEP) distribution (see lmomgep). The function is

$$F_{p,q}(\mathbf{n}, \mathbf{d}; \lambda) = \sum_{k=0}^{\infty} \frac{\lambda^k}{\Gamma(k+1)} \frac{\Pi_{i=1}^{p}\Gamma(n_i+k)\Gamma^{-1}(n_i)}{\Pi_{i=1}^{q}\Gamma(d_i+k)\Gamma^{-1}(d_i)},$$

where $\mathbf{n} = [n_1, n_2, \ldots, n_p]$ for p operands and $\mathbf{d} = [d_1, d_2, \ldots, d_q]$ for q operands, and $\lambda > 0$ is a parameter.

Usage

```
BEhypergeo(p,q, N,D, lambda, eps=1E-12, maxit=500)
```

Arguments

p	An integer value.
q	An integer value.
N	A scalar or vector associated with the p summation (see Details).
D	A scalar or vector associated with the q summation (see Details).
lambda	A real value $\lambda > 0$.
eps	The relative convergence error on which to break an infinite loop.
maxit	The maximum number of interations before a mandatory break on the loop, and a warning will be issued.

Details

For the GEP both $\mathbf{n}$ and $\mathbf{d}$ are vectors of the same value, such as $\mathbf{n} = [1, \ldots, 1]$ and $\mathbf{d} = [2, \ldots, 2]$. This implementation is built around this need by the GEP and if the length of either vector is not equal to the operand then the first value of the vector is repeated the operand times. For example for $\mathbf{n}$, if `n = 1`, then `n = rep(n[1], length(p))` and so on for $\mathbf{d}$. Given that `n` and `d` are vectorized for the GEP, then a shorthand is used for the GEP mathematics shown herein:

$$F_{22}^{12}(h(j+1)) \equiv F_{2,2}([1, \ldots, 1], [2, \ldots, 2]; h(j+1)),$$

for the h parameter of the distribution.

Lastly, for **lmomco** and the GEP the arguments only involve $p = q = 2$ and $N = 1$, $D = 2$, so the function is uniquely a function of the h parameter of the distribution:

```
H <- 10^seq(-10,10, by=0.01)
F22 <- sapply(1:length(H), function(i) BEhypergeo(2,2,1,1, H[i])$value
plot(log10(H), log10(F22), type="l")
```

For this example, the solution increasingly wobbles towards large h, which is further explored by

```
plot(log10(H[1:(length(H)-1)]), diff(log10(F22)), type="l", xlim=c(0,7))
plot(log10(H[H > 75 & H < 140]), c(NA,diff(log10(F22[H > 75 & H < 140]))),
     type="b"); lines(c(2.11,2.11), c(0,10))
```

It can be provisionally concluded that the solution to $F_{22}^{12}(\cdot)$ begins to be suddenly questionable because of numerical difficulties beyond $\log(h) = 2.11$. Therefore, it is given that $h < 128$ might be an operational numerical upper limit.

Value

An R `list` is returned.

`value`	The value for the function.
`its`	The number of iterations j.
`error`	The error of convergence.

Author(s)

W.H. Asquith

References

Kus, C., 2007, A new lifetime distribution: Computational Statistics and Data Analysis, v. 51, pp. 4497–4509.

See Also

`lmomgep`

Examples

```
BEhypergeo(2,2,1,2,1.5)
```

`bfrlmomco` *Bonferroni Curve of the Distributions*

Description

This function computes the Bonferroni Curve for quantile function $x(F)$ (`par2qua`, `qlmomco`). The function is defined by Nair et al. (2013, p. 179) as

$$B(u) = \frac{1}{\mu u} \int_0^u x(p)\; \mathrm{d}p,$$

where $B(u)$ is Bonferroni curve for quantile function $x(F)$ and μ is the conditional mean for quantile $u = 0$ (`cmlmomco`). The Bonferroni curve is related to the Lorenz curve ($L(u)$, `lrzlmomco`) by

$$B(u) = \frac{L(u)}{u}.$$

Usage

```
bfrlmomco(f, para)
```

Arguments

`f`	Nonexceedance probability ($0 \le F \le 1$).
`para`	The parameters from `lmom2par` or `vec2par`.

Value

Bonferroni curve value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

`qlmomco`, `lrzlmomco`

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0

"afunc" <- function(u) { return(par2qua(u,A,paracheck=FALSE)) }
f <- 0.65 # Both computations report: 0.5517342
Bu1 <- 1/(cmlmomco(f=0,A)*f) * integrate(afunc, 0, f)$value
Bu2 <- bfrlmomco(f, A)
```

Bpwm2ApwmRC — *Conversion between B- and A-Type Probability-Weighted Moments for Right-Tail Censoring of an Appropriate Distribution*

Description

This function converts "B"-type probability-weighted moments (PWMs, β_r^B) to the "A"-type β_r^A. The β_r^A are the ordinary PWMs for the m left noncensored or observed values. The β_r^B are more complex and use the m observed values and the $m - n$ right-tailed censored values for which the censoring threshold is known. The "A"- and "B"-type PWMs are described in the documentation for pwmRC.

This function uses the defined relation between to two PWM types when the β_r^B are known along with the parameters (para) of a right-tail censored distribution inclusive of the censoring fraction $\zeta = m/n$. The value ζ is the right-tail censor fraction or the probability $\mathrm{Pr}\{\}$ that x is less than the quantile at ζ nonexceedance probability ($\mathrm{Pr}\{x < X(\zeta)\}$). The relation is

$$\beta_{r-1}^A = \frac{r\beta_{r-1}^B - (1-\zeta^r)X(\zeta)}{r\zeta^r},$$

where $1 \leq r \leq n$ and n is the number of moments, and $X(\zeta)$ is the value of the quantile function at nonexceedance probability ζ. Finally, the RC in the function name is to denote Right-tail Censoring.

Usage

```
Bpwm2ApwmRC(Bpwm,para)
```

Arguments

Bpwm	A vector of B-type PWMs: β_r^B.
para	The parameters of the distribution from a function such as pargpaRC in which the β_r^B are contained in a list element titled betas and the right-tail censoring fraction ζ is contained in an element titled zeta.

Value

An R list is returned.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

Apwm2BpwmRC and pwmRC

Examples

```
# Data listed in Hosking (1995, table 29.2, p. 551)
H <- c(3,4,5,6,6,7,8,8,9,9,9,10,10,11,11,11,13,13,13,13,13,
             17,19,19,25,29,33,42,42,51.9999,52,52,52)
      # 51.9999 was really 52, a real (noncensored) data point.
z <-  pwmRC(H,52)
# The B-type PMWs are used for the parameter estimation of the
# Reverse Gumbel distribution. The parameter estimator requires
# conversion of the PWMs to L-moments by pwm2lmom().
para <- parrevgum(pwm2lmom(z$Bbetas),z$zeta) # parameter object
Abetas <- Bpwm2ApwmRC(z$Bbetas,para)
Bbetas <- Apwm2BpwmRC(Abetas$betas,para)
# Assertion that both of the vectors of B-type PWMs should be the same.
str(Bbetas)   # B-type PWMs of the distribution
str(z$Bbetas) # B-type PWMs of the original data
```

---

canyonprecip *Annual Maximum Precipitation Data for Canyon, Texas*

Description

Annual maximum precipitation data for Canyon, Texas

Usage

```
data(canyonprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(canyonprecip)
summary(canyonprecip)
```

---

cdf2lmom *Compute an L-moment from Cumulative Distribution Function*

Description

Compute a single L-moment from a cumulative distribution function. This function is sequentially called by cdf2lmoms to mimic the output structure for multiple L-moments seen by other L-moment computation functions in **lmomco**.

For $r = 1$, the quantile function is actually used for numerical integration to compute the mean. The expression for the mean is

$$\lambda_1 = \int_0^1 x(F)\; \mathrm{d}F,$$

for quantile function $x(F)$ and nonexceedance probability F. For $r \geq 2$, the L-moments can be computed from the cumulative distribution function $F(x)$ by

$$\lambda_r = \frac{1}{r}\sum_{j=0}^{r-2}(-1)^j \binom{r-2}{j}\binom{r}{j+1}\int_{-\infty}^{\infty}[F(x)]^{r-j-1} \times [1-F(x)]^{j+1}\; \mathrm{d}x.$$

This equation is described by Asquith (2011, eq. 6.8), Hosking (1996), and Jones (2004).

Usage

```
cdf2lmom(r, para, fdepth=0, silent=TRUE, ...)
```

Arguments

`r`	The order of the L-moment.
`para`	The parameters from `lmom2par` or similar.
`fdepth`	The depth of the nonexceedance/exceedance probabilities to determine the lower and upper integration limits for the integration involving $F(x)$ through a call to the `par2qua` function. The default of 0 implies the quantile for $F = 0$ and quantile for $F = 1$ as the respective lower and upper limits.
`silent`	A logical to be passed into `cdf2lmom` and then onto the `try` functions encompassing the `integrate` function calls.
`...`	Additional arguments to pass to `par2qua` and `par2cdf`.

Value

The value for the requested L-moment is returned (λ_r).

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Hosking, J.R.M., 1996, Some theoretical results concerning L-moments: Research Report RC14492, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York.

Jones, M.C., 2004, On some expressions for variance, covariance, skewness and L-moments: Journal of Statistical Planning and Inference, v. 126, pp. 97–106.

See Also

cdf2lmoms

Examples

```
para <- vec2par(c(.9,.4), type="nor")
cdf2lmom(4, para) # summarize the value
```

---

cdf2lmoms *Compute L-moments from Cumulative Distribution Function*

Description

Compute the L-moments from a cumulative distribution function. For $r \geq 1$, the L-moments can be computed by sequential calling of cdf2lmom. Consult the documentation of that function for mathematical definitions.

Usage

```
cdf2lmoms(para, nmom=6, fdepth=0, silent=TRUE, lambegr=1, ...)
```

Arguments

para	The parameters from lmom2par or similar.
nmom	The number of moments to compute. Default is 6.
fdepth	The depth of the nonexceedance/exceedance probabilities to determine the lower and upper integration limits through a call to the par2qua function. The default of 0 implies the quantile for $F = 0$ and quantile for $F = 1$ as the respective lower and upper limits.

silent	A logical to be passed into cdf2lmom and then onto the try functions encompassing the integrate function calls.
lambegr	The rth order to begin the sequence for L-moment computation. Can be used as a means to bypass a mean computation if the user has an alternative method for the mean or other central tendency characterization in which case lambegr = 2.
...	Additional arguments to pass to cdf2lmom.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is $\hat{\lambda}_1^{(0,0)}$, second element is $\hat{\lambda}_2^{(0,0)}$, and so on.
ratios	Vector of the L-moment ratios. Second element is $\hat{\tau}^{(0,0)}$, third element is $\hat{\tau}_3^{(0,0)}$ and so on.
trim	Level of symmetrical trimming used in the computation, which will equal NULL is not support for trimming is provided by this function.
leftrim	Level of left-tail trimming used in the computation, which will equal NULL is not support for trimming is provided by this function.
rightrim	Level of right-tail trimming used in the computation, which will equal NULL is not support for trimming is provided by this function.
source	An attribute identifying the computational source of the L-moments: “cdf2lmoms”.

Author(s)

W.H. Asquith

See Also

cdf2lmom, lmoms

Examples

```
cdf2lmoms(vec2par(c(10,40), type="ray"))
## Not run:
# relatively slow computation
vec2par(c(.9,.4), type="emu"); cdf2lmoms(para, nmom=4)
vec2par(c(.9,.4), type="emu"); cdf2lmoms(para, nmom=4, fdepth=0)
## End(Not run)
```

`cdfaep4` *Cumulative Distribution Function of the 4-Parameter Asymmetric Exponential Power Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the 4-parameter Asymmetric Exponential Power distribution given parameters (ξ, α, κ, and h) computed by `paraep4`. The cumulative distribution function is

$$F(x) = \frac{\kappa^2}{(1+\kappa^2)}\,\gamma([(\xi - x)/(\alpha\kappa)]^h,\ 1/h),$$

for $x < \xi$ and

$$F(x) = 1 - \frac{1}{(1+\kappa^2)}\,\gamma([\kappa(x - \xi)/\alpha]^h,\ 1/h),$$

for $x \ge \xi$, where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, α is a scale parameter, κ is a shape parameter, h is another shape parameter, and $\gamma(Z, s)$ is the upper tail of the incomplete gamma function for the two arguments. The upper tail of the incomplete gamma function is `pgamma(Z, shape, lower.tail=FALSE)` in R and mathematically is

$$\gamma(Z, a) = \int_Z^\infty y^{a-1} \exp(-y)\,\mathrm{d}y\,/\,\Gamma(a).$$

If the τ_3 of the distribution is zero (symmetrical), then the distribution is known as the Exponential Power.

Usage

```
cdfaep4(x, para, paracheck=TRUE)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `paraep4` or `vec2par`.
`paracheck`	A logical controlling whether the parameters and checked for validity.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Delicado, P., and Goria, M.N., 2008, A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution: Computational Statistics and Data Analysis, v. 52, no. 3, pp. 1661–1673.

See Also

pdfaep4, quaaep4, lmomaep4, paraep4

Examples

```
x <- -0.1
para <- vec2par(c(0, 100, 0.5, 4), type="aep4")
FF <- cdfaep4(-.1,para)
cat(c("F=",FF,"  and estx=",quaaep4(FF, para),"\n"))
## Not run:
delx <- .1
x <- seq(-20,20, by=delx);
K <- 1;
PAR <- list(para=c(0,1, K, 0.5), type="aep4");
plot(x,cdfaep4(x, PAR), type="n",ylim=c(0,1), xlim=range(x),
     ylab="NONEXCEEDANCE PROBABILITY");
lines(x,cdfaep4(x,PAR), lwd=4);
lines(quaaep4(cdfaep4(x,PAR),PAR), cdfaep4(x,PAR), col=2)
PAR <- list(para=c(0,1, K, 1), type="aep4");
lines(x,cdfaep4(x, PAR), lty=2, lwd=4);
lines(quaaep4(cdfaep4(x,PAR),PAR), cdfaep4(x,PAR), col=2)
PAR <- list(para=c(0,1, K, 2), type="aep4");
lines(x,cdfaep4(x, PAR), lty=3, lwd=4);
lines(quaaep4(cdfaep4(x,PAR),PAR), cdfaep4(x,PAR), col=2)
PAR <- list(para=c(0,1, K, 4), type="aep4");
lines(x,cdfaep4(x, PAR), lty=4, lwd=4);
lines(quaaep4(cdfaep4(x,PAR),PAR), cdfaep4(x,PAR), col=2)
## End(Not run)
```

---

cdfben *Cumulative Distribution Function of the Benford Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Benford distribution (Benford's Law) given parameters defining the number of first M-significant digits and the numeric base. The cumulative distribution function has a somewhat simple analytical form by direct summation of the probability mass function (pmfben).

Usage

```
cdfben(d, para=list(para=c(1, 10)), ...)
```

Arguments

`d`	A integer value vector of M-significant digits.
`para`	The number of the first M-significant digits followed by the numerical base (only base10 supported) and the list structure mimics similar uses of the **lmomco** list structure. Default are the first significant digits and hence the digits 1 through 9.
`...`	Additional arguments to pass (not likely to be needed but changes in base handling might need this).

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Benford, F., 1938, The law of anomalous numbers: Proceedings of the American Philosophical Society, v. 78, no. 4, pp. 551–572, https://www.jstor.org/stable/984802.

Goodman, W., 2016, The promises and pitfalls of Benford's law: Significance (Magazine), June 2015, pp. 38–41, doi:10.1111/j.17409713.2016.00919.x.

See Also

pmfben, quaben

Examples

```
para <- list(para=c(2, 10))
cdfben(c(15, 25), para=para) # 0.2041200 0.4149733

sum(diff(cdfben(seq(10,99,0.1), para=para))) + cdfben(10, para=para) # 1
```

cdfcau *Cumulative Distribution Function of the Cauchy Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Cauchy distribution given parameters (ξ and α) computed by parcau. The cumulative distribution function is

$$F(x) = \frac{\arctan(Y)}{\pi} + 0.5,$$

where Y is

$$Y = \frac{x - \xi}{\alpha}\text{, and}$$

where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
cdfcau(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parcau or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

Gilchrist, W.G., 2000, Statistical modeling with quantile functions: Chapman and Hall/CRC, Boca Raton, FL.

See Also

pdfcau, quacau, lmomcau, parcau

Examples

```
para <- c(12,12)
cdfcau(50,vec2par(para,type='cau'))
```

cdfemu *Cumulative Distribution Function of the Eta-Mu Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Eta-Mu ($\eta : \mu$) distribution given parameters (η and μ) computed by parkmu. The cumulative distribution function is complex and numerical integration of the probability density function pdfemu is used or the Yacoub (2007) $Y_\nu(a, b)$ integral. The cumulative distribution function in terms of this integral is

$$F(x) = 1 - Y_\nu\left(\frac{H}{h}, x\sqrt{2h\mu}\right),$$

where

$$Y_\nu(a, b) = \frac{2^{3/2-\nu}\sqrt{\pi}(1-a^2)^\nu}{a^{\nu-1/2}\Gamma(\nu)} \int_b^\infty x^{2\nu} \exp(-x^2)\, I_{\nu-1/2}(ax^2)\, \mathrm{d}x,$$

where $I_\nu(a)$ is the "νth-order modified Bessel function of the first kind."

Usage

```
cdfemu(x, para, paracheck=TRUE, yacoubsintegral=TRUE)
```

Arguments

`x`	A real value vector.
`para`	The parameters from paremu or vec2par.
`paracheck`	A logical controlling whether the parameters and checked for validity.
`yacoubsintegral`	A logical controlling whether the integral by Yacoub (2007) is used instead of numerical integration of pdfemu.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

pdfemu, quaemu, lmomemu, paremu

Examples

```
para <- vec2par(c(0.5, 1.4), type="emu")
cdfemu(1.2, para, yacoubsintegral=TRUE)
cdfemu(1.2, para, yacoubsintegral=FALSE)
## Not run:
delx <- 0.01; x <- seq(0,3, by=delx)
nx <- 20*log10(x)
plot(c(-30,10), 10^c(-3,0), log="y", xaxs="i", yaxs="i",
     xlab="RHO", ylab="cdfemu(RHO)", type="n")
m <- 0.75
mus <- c(0.7425, 0.7125, 0.675, 0.6, 0.5, 0.45)
for(mu in mus) {
   eta <- sqrt((m / (2*mu))^-1 - 1)
   lines(nx, cdfemu(x, vec2par(c(eta, mu), type="emu")))
}
mtext("Yacoub (2007, figure 8)")

# Now add some last boundary lines
mu <- m; eta <- sqrt((m / (2*mu))^-1 - 1)
lines(nx, cdfemu(x, vec2par(c(eta, mu), type="emu")),  col=8, lwd=4)
mu <- m/2; eta <- sqrt((m / (2*mu))^-1 - 1)
lines(nx, cdfemu(x, vec2par(c(eta, mu), type="emu")), col=4, lwd=2, lty=2)


delx <- 0.01; x <- seq(0,3, by=delx)
nx <- 20*log10(x)
m <- 0.75; col <- 4; lty <- 2
plot(c(-30,10), 10^c(-3,0), log="y", xaxs="i", yaxs="i",
     xlab="RHO", ylab="cdfemu(RHO)", type="n")
for(mu in c(m/2,seq(m/2+0.01,m,by=0.01), m-0.001, m)) {
   if(mu > 0.67) { col <- 2; lty <- 1 }
   eta <- sqrt((m / (2*mu))^-1 - 1)
   lines(nx, cdfemu(x, vec2par(c(eta, mu), type="emu")),
         col=col, lwd=.75, lty=lty)
}
## End(Not run)
```

---

cdfexp *Cumulative Distribution Function of the Exponential Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Exponential distribution given parameters (ξ and α computed by parexp. The cumulative distribution function is

$$F(x) = 1 - \exp(Y),$$

where Y is

$$\frac{-(x - \xi)}{\alpha},$$

where $F(x)$ is the nonexceedance probability for the quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
cdfexp(x, para)
```

Arguments

x	A real value vector.
para	The parameters from `parexp` or `vec2par`.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, p. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`pdfexp`, `quaexp`, `lmomexp`, `parexp`

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfexp(50,parexp(lmr))
```

cdfgam *Cumulative Distribution Function of the Gamma Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Gamma distribution given parameters (α and β) computed by pargam. The cumulative distribution function has no explicit form but is expressed as an integral:

$$F(x) = \frac{\beta^{-\alpha}}{\Gamma(\alpha)} \int_0^x t^{\alpha-1} \exp(-t/\beta)\; \mathrm{d}t,$$

where $F(x)$ is the nonexceedance probability for the quantile x, α is a shape parameter, and β is a scale parameter.

Alternatively, a three-parameter version is available following the parameterization of the Generalized Gamma distribution used in the **gamlss.dist** package and is

$$F(x) = \frac{\theta^\theta\,|\nu|}{\Gamma(\theta)} \int_0^x \frac{z^\theta}{x} \exp(-z\theta)\; \mathrm{d}x,$$

where $z = (x/\mu)^\nu$, $\theta = 1/(\sigma^2\,|\nu|^2)$ for $x > 0$, location parameter $\mu > 0$, scale parameter $\sigma > 0$, and shape parameter $-\infty < \nu < \infty$. The three parameter version is automatically triggered if the length of the `para` element is three and not two.

Usage

```
cdfgam(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from pargam or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfgam, quagam, lmomgam, pargam

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfgam(50,pargam(lmr))

# A manual demonstration of a gamma parent
G  <- vec2par(c(0.6333,1.579),type='gam') # the parent
F1 <- 0.25         # nonexceedance probability
x  <- quagam(F1,G) # the lower quartile (F=0.25)
a  <- 0.6333       # gamma parameter
b  <- 1.579        # gamma parameter
# compute the integral
xf <- function(t,A,B) { t^(A-1)*exp(-t/B) }
Q  <- integrate(xf,0,x,A=a,B=b)
# finish the math
F2 <- Q$val*b^(-a)/gamma(a)
# check the result
if(abs(F1-F2) < 1e-8) print("yes")

## Not run:
# 3-p Generalized Gamma Distribution and gamlss.dist package parameterization
gg <- vec2par(c(7.4, 0.2, 14), type="gam"); X <- seq(0.04,9, by=.01)
GGa <- gamlss.dist::pGG(X, mu=7.4, sigma=0.2, nu=14)
GGb <- cdfgam(X, gg) # lets compare the two cumulative probabilities
plot( X, GGa, type="l", xlab="X", ylab="PROBABILITY", col=3, lwd=6)
lines(X, GGb, col=2, lwd=2) #
## End(Not run)

## Not run:
# 3-p Generalized Gamma Distribution and gamlss.dist package parameterization
gg <- vec2par(c(4, 1.5, -.6), type="gam"); X <- seq(0,1000, by=1)
GGa <- 1-gamlss.dist::pGG(X, mu=4, sigma=1.5, nu=-.6) # Note 1-... (pGG bug?)
GGb <- cdfgam(X, gg) # lets compare the two cumulative probabilities
plot( X, GGa, type="l", xlab="X", ylab="PROBABILITY", col=3, lwd=6)
lines(X, GGb, col=2, lwd=2) #
## End(Not run)
```

---

cdfgdd *Cumulative Distribution Function of the Gamma Difference Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Gamma Difference distribution (Klar, 2015) given parameters ($\alpha_1 > 0$, $\beta_1 > 0$, $\alpha_2 > 0$, $\beta_2 > 0$) computed by pargdd. The cumulative distribution function is complex and numerical integration is used.

$$F(x) = \frac{\beta_2^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_{\max\{0,-t\}}^{\infty} x^{\alpha_2-1} e^{-\beta_2 x} \gamma\bigl(\alpha_1, \beta_1(x+t)\bigr)\, \mathrm{d}x,$$

where $F(x)$ is the nonexceedance probability for quantile $x \in (-\infty, \infty)$, $\Gamma(y)$ is the complete gamma function, and $\gamma(a, y)$ is the lower incomplete gamma function

$$\gamma(a, y) = \int_0^y t^{a-1} e^{-t}\, \mathrm{d}t.$$

The so-called Gamma Difference distribution is the distribution for the difference of two Gamma random variables $X_1 \sim \Gamma(\alpha_1, \beta_1)$ and $X_1 \sim \Gamma(\alpha_2, \beta_2)$; $X = X_1 - X_2$ is a Gamma Difference random variable. The distribution has other names in the literature.

Usage

```
cdfgdd(x, para, paracheck=TRUE, silent=TRUE, ...)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `pargdd` or `vec2par`.
`paracheck`	A logical controlling whether the parameters are checked for validity.
`silent`	The argument of `silent` for the `try()` operation wrapped on `integrate()`.
`...`	Additional argument to pass.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Klar, B., 2015, A note on gamma difference distributions: Journal of Statistical Computation and Simulation v. 85, no. 18, pp. 1–8, doi:10.1080/00949655.2014.996566.

See Also

`pdfgdd`, `quagdd`, `lmomgdd`, `pargdd`

Examples

```
## Not run:
x <- seq(-5, 7, by=0.01)
para <- list(para=c(3,   1, 1, 1), type="gdd")
plot(x, cdfgdd(x, para), type="l", xlim=c(-5,7), ylim=c(0, 1),
     xlab="x", ylab="distribution function of gamma difference distribution")
para <- list(para=c(2,   1, 1, 1), type="gdd")
lines(x, cdfgdd(x, para), lty=2)
para <- list(para=c(1,   1, 1, 1), type="gdd")
lines(x, cdfgdd(x, para), lty=3)
para <- list(para=c(0.5, 1, 1, 1), type="gdd")
lines(x, cdfgdd(x, para), lty=4) #
## End(Not run)
```

`cdfgep` *Cumulative Distribution Function of the Generalized Exponential Poisson Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Generalized Exponential Poisson distribution given parameters (β, κ, and h) computed by pargep. The cumulative distribution function is

$$F(x) = \left(\frac{1 - \exp[-h + h\exp(-\eta x)]}{1 - \exp(-h)}\right)^{\kappa},$$

where $F(x)$ is the nonexceedance probability for quantile $x > 0$, $\eta = 1/\beta$, $\beta > 0$ is a scale parameter, $\kappa > 0$ is a shape parameter, and $h > 0$ is another shape parameter.

Usage

```
cdfgep(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from pargep or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Barreto-Souza, W., and Cribari-Neto, F., 2009, A generalization of the exponential-Poisson distribution: Statistics and Probability, 79, pp. 2493–2500.

See Also

pdfgep, quagep, lmomgep, pargep

Examples

```
gep <- list(para=c(2, 1.5, 3), type="gep")
cdfgep(0.48,gep)
```

---

cdfgev *Cumulative Distribution Function of the Generalized Extreme Value Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Generalized Extreme Value distribution given parameters (ξ, α, and κ) computed by pargev. The cumulative distribution function is

$$F(x) = \exp(-\exp(-Y)),$$

where Y is

$$Y = -\kappa^{-1} \log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$ and

$$Y = (x-\xi)/\alpha,$$

for $\kappa = 0$, where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of x is $-\infty < x \leq \xi + \alpha/\kappa$ if $k > 0$; $\xi + \alpha/\kappa \leq x < \infty$ if $\kappa \leq 0$. Note that the shape parameter κ parameterization of the distribution herein follows that in tradition by the greater L-moment community and others use a sign reversal on κ. (The **evd** package is one example.)

Usage

```
cdfgev(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from pargev or vec2par.
paracheck	A logical switch as to whether the validity of the parameters should be checked.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfgev, quagev, lmomgev, pargev

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
cdfgev(50, pargev(lmr))
```

`cdfgld` *Cumulative Distribution Function of the Generalized Lambda Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Generalized Lambda distribution given parameters (ξ, α, κ, and h) computed by `pargld`. The cumulative distribution function has no explicit form and requires numerical methods. The R function `uniroot` is used to root the quantile function `quagld` to compute the nonexceedance probability. The function returns 0 or 1 if the `x` argument is at or beyond the limits of the distribution as specified by the parameters.

Usage

```
cdfgld(x, para, paracheck)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `pargld` or `vec2par`.
`paracheck`	A logical switch as to whether the validity of the parameters should be checked. Default is `paracheck=TRUE`. This switch is made so that the root solution needed for `cdfgld` exhibits an extreme speed increase because of the repeated calls to `quagld`.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

`pdfgld`, `quagld`, `lmomgld`, `pargld`

Examples

```
## Not run:
  P <- vec2par(c(123,340,0.4,0.654),type='gld')
  cdfgld(300,P, paracheck=FALSE)

  par <- vec2par(c(0,-7.901925e+05, 6.871662e+01, -3.749302e-01), type="gld")
  supdist(par)

## End(Not run)
```

`cdfglo` *Cumulative Distribution Function of the Generalized Logistic Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Generalized Logistic distribution given parameters (ξ, α, and κ) computed by parglo. The cumulative distribution function is

$$F(x) = 1/(1 + \exp(-Y)),$$

where Y is

$$Y = -\kappa^{-1} \log\left(1 - \frac{\kappa(x - \xi)}{\alpha}\right),$$

for $\kappa \neq 0$ and

$$Y = (x - \xi)/\alpha,$$

for $\kappa = 0$, where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
cdfglo(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parglo or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfglo, quaglo, lmomglo, parglo

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfglo(50,parglo(lmr))
```

cdfgno *Cumulative Distribution Function of the Generalized Normal Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Generalized Normal distribution given parameters (ξ, α, and κ) computed by pargno. The cumulative distribution function is

$$F(x) = \Phi(Y),$$

where Φ is the cumulative distribution function of the Standard Normal distribution and Y is

$$Y = -\kappa^{-1} \log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$ and

$$Y = (x-\xi)/\alpha,$$

for $\kappa = 0$, where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
cdfgno(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from pargno or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfgno, quagno, lmomgno, pargno, cdfln3

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfgno(50,pargno(lmr))
```

---

cdfgov *Cumulative Distribution Function of the Govindarajulu Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Govindarajulu distribution given parameters (ξ, α, and β) computed by pargov. The cumulative distribution function has no explicit form and requires numerical methods. The R function `uniroot` is used to root the quantile function quagov to compute the nonexceedance probability. The function returns 0 or 1 if the `x` argument is at or beyond the limits of the distribution as specified by the parameters.

Usage

```
cdfgov(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from pargov or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

Nair, N.U., Sankaran, P.G., and Vineshkumar, B., 2012, The Govindarajulu distribution—Some Properties and applications: Communications in Statistics, Theory and Methods, v. 41, no. 24, pp. 4391–4406.

See Also

pdfgov, quagov, lmomgov, pargov

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfgov(50,pargov(lmr))
```

---

cdfgpa *Cumulative Distribution Function of the Generalized Pareto Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Generalized Pareto distribution given parameters (ξ, α, and κ) computed by pargpa. The cumulative distribution function is

$$F(x) = 1 - \exp(-Y),$$

where Y is

$$Y = -\kappa^{-1} \log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$ and

$$Y = (x - \xi)/\alpha,$$

for $\kappa = 0$, where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of x is $\xi \leq x \leq \xi + \alpha/\kappa$ if $k > 0$; $\xi \leq x < \infty$ if $\kappa \leq 0$. Note that the shape parameter κ parameterization of the distribution herein follows that in tradition by the greater L-moment community and others use a sign reversal on κ. (The **evd** package is one example.)

Usage

```
cdfgpa(x, para)
```

Arguments

x	A real value vector.
para	The parameters from pargpa or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfgpa, quagpa, lmomgpa, pargpa

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
cdfgpa(50, pargpa(lmr))
```

---

cdfgum *Cumulative Distribution Function of the Gumbel Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Gumbel distribution given parameters (ξ and α) computed by pargum. The cumulative distribution function is

$$F(x) = \exp(-\exp(Y)),$$

where

$$Y = -\frac{x - \xi}{\alpha},$$

where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
cdfgum(x, para)
```

Arguments

x	A real value vector.
para	The parameters from pargum or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfgum, quagum, lmomgum, pargum

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfgum(50,pargum(lmr))
```

---

cdfkap — *Cumulative Distribution Function of the Kappa Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Kappa of the distribution computed by parkap. The cumulative distribution function is

$$F(x) = \left(1 - h\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right)^{1/\kappa}\right)^{1/h},$$

where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, α is a scale parameter, κ is a shape parameter, and h is another shape parameter.

Usage

```
cdfkap(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parkap or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1994, The four-parameter kappa distribution: IBM Journal of Reserach and Development, v. 38, no. 3, pp. 251–258.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfkap, quakap, lmomkap, parkap

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78,21,32,231,23))
cdfkap(50,parkap(lmr))
```

---

cdfkmu — *Cumulative Distribution Function of the Kappa-Mu Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Kappa-Mu ($\kappa : \mu$) distribution given parameters (κ and μ) computed by parkmu. The cumulative distribution function is complex and numerical integration of the probability density function pdfkmu is used. Alternatively, the cumulative distribution function may be defined in terms of the Marcum Q function

$$F(x) = 1 - Q_\nu\left(\sqrt{2\kappa\mu},\, x\sqrt{2(1+\kappa)\mu}\right),$$

where $F(x)$ is the nonexceedance probability for quantile x and $Q_v(a,b)$ is the Marcum Q function defined by

$$Q_\nu(a,b) = \frac{1}{\alpha^{\nu-1}}\int_b^\infty t^\nu\, \exp(-(t^2+a^2)/2)\, I_{\nu-1}(at)\, \mathrm{d}t,$$

which can be numerically difficult to work with and particularly so with real number values for ν. $I_\nu(a)$ is the "νth-order modified Bessel function of the first kind."

Following an apparent breakthrough(?) by Shi (2012), ν can be written as $\nu = n + \Delta$ where n is an integer and $0 < \Delta \leq 1$. The author of **lmomco** refers to this alternative formulation as the "delta nu method". The Marcum Q function for $\nu > 0$ ($n = 1, 2, 3, \cdots$) is

$$Q_\nu(a,b) = Q_\Delta(a,b) + \exp(-(a^2+b^2)/2)\sum_{i=0}^{n-1}\left(\frac{b}{a}\right)^{i+\Delta} I_{i+\Delta}(ab),$$

and the function for $\nu \leq 0$ ($n = -1, -2, -3, \cdots$) is

$$Q_\nu(a,b) = Q_\Delta(a,b) - \exp(-(a^2+b^2)/2) \times \sum_{i=n}^{-1} \left(\frac{b}{a}\right)^{i+\Delta} \mathrm{I}_{i+\Delta}(ab),$$

and the function for $\nu = 0$ is

$$Q_\nu(a,b) = Q_\Delta(a,b) + \exp(-(a^2+b^2)/2).$$

Shi (2012) concludes that the "merit" of these two expressions is that the evaulation of the Marcum Q function is reduced to the numerical evaluation of $Q_\Delta(a,b)$. This difference can result in measurably faster computation times (confirmed by limited analysis by the author of **lmomco**) and possibly better numerical performance.

Shi (2012) uses notation and text that implies evaluation of the far-right additive term (the summation) for $n = 0$ as part of the condition $\nu > 0$. To clarify, Shi (2012) implies for $\nu > 0; n = 0$ (but $n = 0$ occurs also for $-1 < \nu <= 0$) the following computation

$$Q_\nu(a,b) = Q_\Delta(a,b) + \exp(-(a^2+b^2)/2) \times \left[\left(\frac{b}{a}\right)^{\Delta} \mathrm{I}_\Delta(ab) + \left(\frac{b}{a}\right)^{\Delta-1} \mathrm{I}_{\Delta-1}(ab)\right]$$

This result produces incompatible cumulative distribution functions of the distribution using $Q_\nu(a,b)$ for $-1 < \nu < 1$. Therefore, the author of **lmomco** concludes that Shi (2012) is in error (or your author misinterprets the summation notation) and that the specific condition for $\nu = 0$ shown above and lacking $\sum$ is correct; there are three individual and separate conditions to support the Marcum Q function using the "delta nu method": $\nu \leq -1$, $-1 < \nu < 1$, and $\nu \geq -1$.

Usage

```
cdfkmu(x, para, paracheck=TRUE, getmed=TRUE, qualo=NA, quahi=NA,
                marcumQ=TRUE, marcumQmethod=c("chisq", "delta", "integral"))
```

Arguments

`x` A real value vector.

`para` The parameters from `parkmu` or `vec2par`.

`paracheck` A logical controlling whether the parameters and checked for validity.

`getmed` Numerical problems rolling onto the distribution from the right can result in erroneous F being integrated of `pdfkmu`. This option is used to interrupt recurrsion, but if `TRUE`, then the median will be computed and for those x values less than the median and F initially computing as greater than 50 percent, are reset to 0. Users are unlikely to need this option changed. But the hack can be turned off by setting `getmed=FALSE` as the user level.

`qualo` A lower limit of the range of x to look for a `uniroot` of $F(x) = 0.5$ to estimate the median quantile that is used to mitigate for erroneous numerical results. This argument is passed along to `quakmu` but also used as a truncation point for which $F = 1$ is returned if $x <$ `qualo`. Lastly, see the last example below.

quahi — An upper limit of the range of x to look for a uniroot of $F(x) = 0.5$ to estimate the median quantile that is used to mitigate for erroneous numerical results. This argument is passed along to quakmu but also used as a truncation point for which $F = 1$ is returned if $x >$ quahi. Lastly, see the last example below.

marcumQ — A logical controlling whether the Marcum Q function is used instead of numerical integration of pdfkmu.

marcumQmethod — Which method for Marcum Q computation is to be used (see source code).

Value

Nonexceedance probability (F) for x.

Note

Code developed from Weinberg (2006). The biascor feature is of my own devise and this Poisson method does not seem to accommodate nu < 1 although Chornoboy claims valid for non-negative integer. The example implementation here will continue to use real values of nu.

```
See NEWS file and entries for version 2.0.1 for this "R Marcum"
"marcumq" <- function(a, b, nu=1) {
      pchisq(b^2, df=2*nu, ncp=a^2, lower.tail=FALSE) }

"marcumq.poissons" <-
   function(a,b, nu=NULL, nsim=10000, biascor=0.5) {
   asint <- as.logical(nu
   biascor <- ifelse(! asint, 0, biascor)
   marcumQint <- marcumq(a, b, nu=nu)
   B <- rpois(nsim, b^2/2)
   A <- nu - 1 + biascor + rpois(nsim, a^2/2)
   L <- B <= A
   marcumQppois <- length(L[L == TRUE])/nsim
   z <- list(MarcumQ.by.usingR = marcumQint,
             MarcumQ.by.poisson = marcumQppois)
   return(z)
}
x <- y <- vector()
for(i in 1:10000) {
   nu <- i/100
   z <- marcumq.poissons(12.4, 12.5, nu=nu)
   x[i] <- z$MarcumQ.by.usingR
   y[i] <- z$MarcumQ.by.poisson
}
plot(x,y, pch=16, col=rgb(x,0,0,.2),
     xlab="Marcum Q-function using R (ChiSq distribution)",
     ylab="Marcum Q-function by two Poisson random variables")
abline(0,1, lty=2)
```

Author(s)

W.H. Asquith

References

Shi, Q., 2012, Semi-infinite Gauss-Hermite quadrature based approximations to the generalized Marcum and Nuttall Q-functions and further applications: First IEEE International Conference on Communications in China—Communications Theory and Security (CTS), pp. 268–273, ISBN 978–1–4673–2815–9,12.

Weinberg, G.V., 2006, Poisson representation and Monte Carlo estimation of generalized Marcum Q-function: IEEE Transactions on Aerospace and Electronic Systems, v. 42, no. 4, pp. 1520–1531.

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81.

See Also

pdfkmu, quakmu, lmomkmu, parkmu

Examples

```
## Not run:
x <- seq(0,3, by=0.5)
para <- vec2par(c(0.69, 0.625), type="kmu")
cdfkmu(x, para, marcumQ=TRUE, marcumQmethod="chisq")
cdfkmu(x, para, marcumQ=TRUE, marcumQmethod="delta")
cdfkmu(x, para, marcumQ=FALSE) # about 3 times slower
## End(Not run)
## Not run:
para <- vec2par(c(0.69, 0.625), type="kmu")
quahi <- supdist(para, delexp=.1)$support[2]
cdfkmu(quahi, para, quahi=quahi)

## End(Not run)
## Not run:
delx <- 0.01
x <- seq(0,3, by=delx)

plot(c(0,3), c(0,1), xlab="RHO", ylab="cdfkmu(RHO)", type="n")
para <- list(para=c(0, 0.75), type="kmu")
cdf <- cdfkmu(x, para)
lines(x, cdf, col=2, lwd=4)
para <- list(para=c(1, 0.5625), type="kmu")
cdf <- cdfkmu(x, para)
lines(x, cdf, col=3, lwd=4)

kappas <- c(0.00000001, 0.69, 1.37,  2.41, 4.45, 10.48, 28.49)
mus    <- c(0.75, 0.625,  0.5,  0.375, 0.25,  0.125, 0.05)
for(i in 1:length(kappas)) {
   kappa <- kappas[i]
   mu    <- mus[i]
```

```
   para <- list(para=c(kappa, mu), type="kmu")
   cdf <- cdfkmu(x, para)
   lines(x, cdf, col=i)
}

## End(Not run)
## Not run:
delx <- 0.005
x <- seq(0,3, by=delx)
nx <- 20*log10(x)
plot(c(-30,10), 10^c(-4,0), log="y", xaxs="i", yaxs="i",
     xlab="RHO", ylab="cdfkmu(RHO)", type="n")
m <- 1.25
mus <- c(0.25, 0.50, 0.75, 1, 1.25, 0)
for(mu in mus) {
   col <- 1
   kappa <- m/mu - 1 + sqrt((m/mu)*((m/mu)-1))
   para <- vec2par(c(kappa, mu), type="kmu")
   if(! is.finite(kappa)) {
      para <- vec2par(c(Inf,m), type="kmu")
      col <- 2
   }
   lines(nx, cdfkmu(x, para), col=col)
}
mtext("Yacoub (2007, figure 4)")

## End(Not run)
## Not run:
# The Marcum Q use for the CDF avoid numerical integration of pdfkmu(), but
# below is an example for which there is some failure that remains to be found.
para <- vec2par(c(10, 23), type="kmu")
# The following are reliable but slower as they avoid the Marcum Q function
# and use traditional numerical integration of the PDF function.
A <- cdfkmu(c(0.10, 0.35, 0.9, 1, 1.16), para, marcumQ=FALSE)
# Continuing, the first value in c() has an erroneous value for the next call.
B <- cdfkmu(c(0.10, 0.35, 0.9, 1, 1.16), para, marcumQ=TRUE)
# But this distribution is tightly peaks and well away from the origin, so in
# order to snap the erroneous value to zero, we need a successful median
# computation.  We can try again using the qualo argument to pass through to
# quakmu() like the following:
C <- cdfkmu(c(0.10, 0.35, 0.9, 1, 1.16), para, marcumQ=TRUE, qualo=0.4)
# The existance of the median for the last one also triggers a truncation of
# the CDF to 0 when negative solution results for the 0.35, although the
# negative is about -1E-14.

## End(Not run)
## Not run:
# Does the discipline of the signal litature just "know" about the apparent
# upper support of the Kappa-Mu being quite near or even at pi?
"simKMU" <- function() {
   km <- 10^runif(2, min=-3, max=3)
   f <- cdfkmu(pi, vec2par(km, type="kmu"))
   return(c(km, f))
```

```
}
EndStudy <- sapply(1:1000, function(i) { simKMU() } )
boxplot(EndStudy[3,])

## End(Not run)
```

---

cdfkur *Cumulative Distribution Function of the Kumaraswamy Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Kumaraswamy distribution given parameters (α and β) computed by parkur. The cumulative distribution function is

$$F(x) = 1 - (1 - x^{\alpha})^{\beta},$$

where $F(x)$ is the nonexceedance probability for quantile x, α is a shape parameter, and β is a shape parameter.

Usage

```
cdfkur(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parkur or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Jones, M.C., 2009, Kumaraswamy's distribution—A beta-type distribution with some tractability advantages: Statistical Methodology, v. 6, pp. 70–81.

See Also

pdfkur, quakur, lmomkur, parkur

Examples

```
lmr <- lmoms(c(0.25, 0.4, 0.6, 0.65, 0.67, 0.9))
cdfkur(0.5,parkur(lmr))
```

cdflap *Cumulative Distribution Function of the Laplace Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Laplace distribution given parameters (ξ and α) computed by parlap. The cumulative distribution function is

$$F(x) = \frac{1}{2}\mathrm{exp}((x - \xi)/\alpha) \text{ for } x \leq \xi,$$

and

$$F(x) = 1 - \frac{1}{2}\mathrm{exp}(-(x - \xi)/\alpha) \text{ for } x > \xi,$$

where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
cdflap(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parlap or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: IBM Research Report RC12210, T.J. Watson Research Center, Yorktown Heights, New York.

See Also

pdflap, qualap, lmomlap, parlap

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdflap(50,parlap(lmr))
```

`cdflmrq` *Cumulative Distribution Function of the Linear Mean Residual Quantile Function Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the "Linear Mean Residual Quantile Function" distribution given parameters computed by `parlmrq`. The cumulative distribution function has no explicit form and requires numerical methods. The R function `uniroot` is used to root the quantile function `qualmrq` to compute the nonexceedance probability. The function returns 0 or 1 if the `x` argument is at or beyond the limits of the distribution as specified by the parameters. The `cdflmrq` function is also used with numerical methods to solve the `pdflmrq`.

Usage

```
cdflmrq(x, para, paracheck=FALSE)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `parlmrq` or `vec2par`.
`paracheck`	A logical switch as to whether the validity of the parameters should be checked. Default is `paracheck=TRUE`. This switch is made so that the root solution needed for `cdflmrq` exhibits an extreme speed increase because of the repeated calls to `qualmrq`.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

See Also

`pdflmrq`, `qualmrq`, `lmomlmrq`, `parlmrq`

Examples

```
lmr <- lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2))
cdflmrq(2,parlmrq(lmr))
```

cdfln3 *Cumulative Distribution Function of the 3-Parameter Log-Normal Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Log-Normal3 distribution given parameters (ζ, lower bounds; $\mu_{\log}$, location; and $\sigma_{\log}$, scale) computed by parln3. The cumulative distribution function (same as Generalized Normal distribution, cdfgno) is

$$F(x) = \Phi(Y),$$

where Φ is the cumulative ditribution function of the Standard Normal distribution and Y is

$$Y = \frac{\log(x-\zeta) - \mu_{\log}}{\sigma_{\log}},$$

where ζ is the lower bounds (real space) for which $\zeta < \lambda_1 - \lambda_2$ (checked in are.parln3.valid), $\mu_{\log}$ be the mean in natural logarithmic space, and $\sigma_{\log}$ be the standard deviation in natural logarithm space for which $\sigma_{\log} > 0$ (checked in are.parln3.valid) is obvious because this parameter has an analogy to the second product moment. Letting $\eta = \exp(\mu_{\log})$, the parameters of the Generalized Normal are $\zeta + \eta$, $\alpha = \eta\sigma_{\log}$, and $\kappa = -\sigma_{\log}$. At this point, the algorithms (cdfgno) for the Generalized Normal provide the functional core.

Usage

```
cdfln3(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from parln3 or vec2par.

Value

Nonexceedance probability (F) for x.

Note

The parameterization of the Log-Normal3 results in ready support for either a known or unknown lower bounds. Details regarding the parameter fitting and control of the ζ parameter can be seen under the Details section in parln3.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

pdfln3, qualn3, lmomln3, parln3, cdfgno

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfln3(50,parln3(lmr))
```

---

cdfnor *Cumulative Distribution Function of the Normal Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Normal distribution given parameters of the distribution computed by parnor. The cumulative distribution function is

$$F(x) = \Phi((x - \mu)/\sigma),$$

where $F(x)$ is the nonexceedance probability for quantile x, μ is the arithmetic mean, and σ is the standard deviation, and Φ is the cumulative distribution function of the Standard Normal distribution, and thus the R function pnorm is used.

Usage

```
cdfnor(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parnor or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfnor, quanor, lmomnor, parnor

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfnor(50,parnor(lmr))
```

---

cdfpdq3 *Cumulative Distribution Function of the Polynomial Density-Quantile3 Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Polynomial Density-Quantile3 (PDQ3) distribution given parameters (ξ, α, κ) computed by parpdq4. The cumulative distribution function has no explicit form and requires numerical methods. The R function `uniroot()` is used to root the quantile function quapdq3 to compute the nonexceedance probability. The distribution's canonical definition is in terms of the quantile function (quapdq3).

Usage

```
cdfpdq3(x, para, paracheck=TRUE)
```

Arguments

`x`	A real value vector.
`para`	The parameters from parpdq3 or vec2par.
`paracheck`	A logical switch as to whether the validity of the parameters should be checked. Default is `paracheck=TRUE`. This switch is made so that the root solution needed for cdfpdq3 shows an extreme speed increase because of the repeated calls to `quapdq3`.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

pdfpdq3, quapdq3, lmompdq3, parpdq3, cdfpdq4

Examples

```
## Not run:
  FF <- seq(0.001, 0.999, by=0.001)
  para  <- list(para=c(0.6933, 1.5495, 0.5488), type="pdq3")
  Fpdq3 <- cdfpdq3(quapdq3(FF, para), para)
  plot(FF, Fpdq3, type="l", col=grey(0.8), lwd=4)
  # should be a 1:1 line, it is
## End(Not run)

## Not run:
  para <- list(para=c(0.6933, 1.5495, 0.5488), type="pdq3")
  X <- seq(-5, +12, by=(12 - -5) / 500)
  plot( X, cdfpdq3(X, para), type="l", col=grey(0.8), lwd=4, ylim=c(0, 1))
  lines(X, pf( exp(X), df1=7, df2=1), lty=2)
  lines(X, c(NA, diff( cdfpdq3(X, para))          / ((12 - -5) / 500)))
  lines(X, c(NA, diff(  pf(exp(X), df1=7, df2=1)) / ((12 - -5) / 500)), lty=2) #
## End(Not run)
```

---

cdfpdq4 *Cumulative Distribution Function of the Polynomial Density-Quantile4 Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Polynomial Density-Quantile4 (PDQ4) distribution given parameters (ξ, α, κ) computed by parpdq4. The cumulative distribution function has no explicit form and requires numerical methods. The R function `uniroot()` is used to root the quantile function quapdq4 to compute the nonexceedance probability. The distribution's canonical definition is in terms of the quantile function (quapdq4).

Usage

```
cdfpdq4(x, para, paracheck=TRUE)
```

Arguments

- `x` — A real value vector.
- `para` — The parameters from `parpdq4` or `vec2par`.
- `paracheck` — A logical switch as to whether the validity of the parameters should be checked. Default is `paracheck=TRUE`. This switch is made so that the root solution needed for `cdfpdq4` shows an extreme speed increase because of the repeated calls to `quapdq4`.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

`pdfpdq4`, `quapdq4`, `lmompdq4`, `parpdq4`, `cdfpdq3`

Examples

```
## Not run:
  FF <- seq(0.001, 0.999, by=0.001)
  para  <- list(para=c(0, 0.4332, -0.7029), type="pdq4")
  Fpdq4 <- cdfpdq4(quapdq4(FF, para), para)
  plot(FF, Fpdq4, type="l", col=grey(0.8), lwd=4)
  # should be a 1:1 line, it is
## End(Not run)

## Not run:
  para <- list(para=c(0, 0.4332, -0.7029), type="pdq4")
  X <- seq(-5, +12, by=(12 - -5) / 500)
  plot( X, cdfpdq4(X, para), type="l", col=grey(0.8), lwd=4, ylim=c(0, 1))
  lines(X, pf( exp(X), df1=5, df2=4), lty=2)
  lines(X, c(NA, diff( cdfpdq4(X, para))          / ((12 - -5) / 500)))
  lines(X, c(NA, diff( pf( exp(X), df1=5, df2=4)) / ((12 - -5) / 500)), lty=2) #
## End(Not run)
```

cdfpe3 *Cumulative Distribution Function of the Pearson Type III Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Pearson Type III distribution given parameters (μ, σ, and γ) computed by parpe3. These parameters are equal to the product moments: mean, standard deviation, and skew (see pmoms). The cumulative distribution function is

$$F(x) = \frac{G\left(\alpha, \frac{Y}{\beta}\right)}{\Gamma(\alpha)},$$

for $\gamma \neq 0$ and where $F(x)$ is the nonexceedance probability for quantile x, G is defined below and is related to the incomplete gamma function of R (`pgamma()`), Γ is the complete gamma function, ξ is a location parameter, β is a scale parameter, α is a shape parameter, and $Y = x - \xi$ if $\gamma > 0$ and $Y = \xi - x$ if $\gamma < 0$. These three "new" parameters are related to the product moments by

$$\alpha = 4/\gamma^2,$$

$$\beta = \frac{1}{2}\sigma|\gamma|,$$

$$\xi = \mu - 2\sigma/\gamma.$$

Lastly, the function $G(\alpha, x)$ is

$$G(\alpha, x) = \int_0^x t^{(a-1)} \exp(-t)\,\mathrm{d}t.$$

If $\gamma = 0$, the distribution is symmetrical and simply is the normal distribution with mean and standard deviation of μ and σ, respectively. Internally, the $\gamma = 0$ condition is implemented by `pnorm()`. If $\gamma > 0$, the distribution is right-tail heavy, and $F(x)$ is the returned nonexceedance probability. On the other hand if $\gamma < 0$, the distribution is left-tail heavy and $1 - F(x)$ is the actual nonexceedance probability that is returned.

Usage

```
cdfpe3(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parpe3 or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfpe3, quape3, lmompe3, parpe3

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfpe3(50,parpe3(lmr))
```

---

cdfray *Cumulative Distribution Function of the Rayleigh Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Rayleigh distribution given parameters (ξ and α) computed by parray. The cumulative distribution function is

$$F(x) = 1 - \exp[-(x-\xi)^2/(2\alpha^2)],$$

where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
cdfray(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parray or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

See Also

pdfray, quaray, lmomray, parray

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfray(50,parray(lmr))
```

---

cdfrevgum *Cumulative Distribution Function of the Reverse Gumbel Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Reverse Gumbel distribution given parameters (ξ and α) computed by parrevgum. The cumulative distribution function is

$$F(x) = 1 - \exp(-\exp(Y)),$$

where

$$Y = -\frac{x - \xi}{\alpha},$$

where $F(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
cdfrevgum(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parrevgum or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

pdfrevgum, quarevgum, lmomrevgum, parrevgum

Examples

```
# See p. 553 of Hosking (1995)
# Data listed in Hosking (1995, table 29.3, p. 553)
D <- c(-2.982, -2.849, -2.546, -2.350, -1.983, -1.492, -1.443,
       -1.394, -1.386, -1.269, -1.195, -1.174, -0.854, -0.620,
       -0.576, -0.548, -0.247, -0.195, -0.056, -0.013,  0.006,
        0.033,  0.037,  0.046,  0.084,  0.221,  0.245,  0.296)
D <- c(D,rep(.2960001,40-28)) # 28 values, but Hosking mentions
                              # 40 values in total
z <-  pwmRC(D,threshold=.2960001)
str(z)
# Hosking reports B-type L-moments for this sample are
# lamB1 = -0.516 and lamB2 = 0.523
btypelmoms <- pwm2lmom(z$Bbetas)
# My version of R reports lamB1 = -0.5162 and lamB2 = 0.5218
str(btypelmoms)
rg.pars <- parrevgum(btypelmoms,z$zeta)
str(rg.pars)
# Hosking reports xi=0.1636 and alpha=0.9252 for the sample
# My version of R reports xi = 0.1635 and alpha = 0.9254
F  <- nonexceeds()
PP <- pp(D) # plotting positions of the data
D  <- sort(D)
plot(D,PP)
lines(D,cdfrevgum(D,rg.pars))
```

---

cdfrice *Cumulative Distribution Function of the Rice Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Rice distribution given parameters (ν and SNR) computed by parrice. The cumulative distribution function is complex and numerical integration of the probability density function pdfrice is used.

$$F(x) = 1 - Q\left(\frac{\nu}{\alpha}, \frac{x}{\alpha}\right),$$

where $F(x)$ is the nonexceedance probability for quantile x, $Q(a,b)$ is the Marcum Q-function, and ν/α is a form of signal-to-noise ratio SNR. If $\nu = 0$, then the Rayleigh distribution results and `pdfray` is used. The Marcum Q-function is difficult to work with and the **lmomco** uses the `integrate` function on `pdfrice` (however, see the Note).

Usage

```
cdfrice(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `parrice` or `vec2par`.

Value

Nonexceedance probability (F) for x.

Note

A user of **lmomco** reported that the Marcum Q function can be computed using R functions. An implementation is shown in this note.

```
See NEWS file and entries for version 2.0.1 for this "R Marcum"
"marcumq" <- function(a, b, nu=1) {
      pchisq(b^2, df=2*nu, ncp=a^2, lower.tail=FALSE) }
```

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`pdfrice`, `quarice`, `lmomrice`, `parrice`

Examples

```
lmr <- vec2lmom(c(45,0.27), lscale=FALSE)
cdfrice(35,parrice(lmr))
```

cdfsla *Cumulative Distribution Function of the Slash Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Slash distribution given parameters (ξ and α) of the distribution provided by parsla or vec2par. The cumulative distribution function is

$$F(x) = \Phi(Y) - [\phi(0) - \phi(Y)]/Y,$$

for $Y \neq 0$ and

$$F(x) = 1/2,$$

for $Y = 0$, where $F(x)$ is the nonexceedance probability for quantile x, $Y = (x - \xi)/\alpha$, ξ is a location parameter, and α is a scale parameter. The function $\Phi(Y)$ is the cumulative distribution function of the Standard Normal distribution, and $\phi(Y)$ is the probability density function of the Standard Normal distribution.

Usage

```
cdfsla(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parsla or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Rogers, W.H., and Tukey, J.W., 1972, Understanding some long-tailed symmetrical distributions: Statistica Neerlandica, v. 26, no. 3, pp. 211–226.

See Also

pdfsla, quasla, lmomsla, parsla

Examples

```
para <- c(12, 1.2)
cdfsla(50, vec2par(para, type="sla"))
```

cdfsmd *Cumulative Distribution Function of the Singh–Maddala Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Singh–Maddala (Burr Type XII) distribution given parameters (a, b, and q) of the distribution computed by parsmd. The cumulative distribution function is

$$F(x) = 1 - \left(1 + \left[(x-\xi)/a\right]^b\right)^{-q},$$

where $F(x)$ is the nonexceedance probability for quantile x with $0 \le x \le \infty$, ξ is a location parameter, a is a scale parameter ($a > 0$), b is a shape parameter ($b > 0$), and q is another shape parameter ($q > 0$).

Usage

```
cdfsmd(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parsmd or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Kumar, D., 2017, The Singh–Maddala distribution—Properties and estimation: International Journal of System Assurance Engineering and Management, v. 8, no. S2, 15 p., doi:10.1007/s13198-01706001.

Shahzad, M.N., and Zahid, A., 2013, Parameter estimation of Singh Maddala distribution by moments: International Journal of Advanced Statistics and Probability, v. 1, no. 3, pp. 121–131, doi:10.14419/ijasp.v1i3.1206.

See Also

pdfsmd, quasmd, lmomsmd, parsmd

Examples

```
# The SMD approximating the normal and use x=0
tau4_of_normal <- 30 * pi^-1 * atan(sqrt(2)) - 9 # from theory
cdfsmd(0, parsmd( vec2lmom( c( -pi, pi, 0, tau4_of_normal ) ) ) ) # 0.7138779
pnorm( 0, mean=-pi, sd=pi*sqrt(pi))                                # 0.7136874

## Not run:
t3 <- 0.6
t4 <- (t3 * (1 + 5 * t3))/(5 + t3) # L-kurtosis of GPA from lmrdia()
paraA <- parsmd( vec2lmom( c( -1000, 200, t3, t4-0.02 ) ) )
paraB <- parsmd( vec2lmom( c( -1000, 200, t3, t4      ) ) )
paraC <- parsmd( vec2lmom( c( -1000, 200, t3, t4+0.02 ) ) )
FF <- nonexceeds(); x <- quasmd(FF, paraA)
plot( x, prob2grv(cdfsmd(x, paraA)), col="red", type="l",
      xlab="Quantile", ylab="Gumbel Reduced Variate, prob2grv()")
lines(x, prob2grv(cdfsmd(x, paraB)), col="green")
lines(x, prob2grv(cdfsmd(x, paraC)), col="blue" ) #
## End(Not run)
```

---

cdfst3 — *Cumulative Distribution Function of the 3-Parameter Student t Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the 3-parameter Student t distribution given parameters (ξ, α, ν) computed by parst3. There is no explicit solution for the cumulative distribution function for value X but built-in R functions can be used. For U = ξ and A = α and for $1.001 \leq \nu \leq 10^5.5$, one can use `pt((X-U)/A, N)` for N = ν. The R function `pt` is used for the 1-parameter Student t cumulative distribution function. The limits for ν stem from study of ability for theoretical integration of the quantile function to produce viable τ_4 and τ_6 (see `inst/doc/t4t6/studyST3.R`).

Usage

```
cdfst3(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from parst3 or vec2par.
paracheck	A logical on whether the parameter should be check for validity.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

pdfst3, quast3, lmomst3, parst3

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
cdfst3(191.5143, parst3(lmr)) # 75th percentile
```

---

cdftexp — *Cumulative Distribution Function of the Truncated Exponential Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Truncated Exponential distribution given parameters (ψ and α) computed by partexp. The parameter ψ is the right truncation of the distribution and α is a scale parameter. The cumulative distribution function, letting $\beta = 1/\alpha$ to match nomenclature of Vogel and others (2008), is

$$F(x) = \frac{1 - \exp(-\beta t)}{1 - \exp(-\beta \psi)},$$

where $F(x)$ is the nonexceedance probability for the quantile $0 \le x \le \psi$ and $\psi > 0$ and $\alpha > 0$. This distribution represents a nonstationary Poisson process.

The distribution is restricted to a narrow range of L-CV ($\tau_2 = \lambda_2/\lambda_1$). If $\tau_2 = 1/3$, the process represented is a stationary Poisson for which the cumulative distribution function is simply the uniform distribution and $F(x) = x/\psi$. If $\tau_2 = 1/2$, then the distribution is represented as the usual exponential distribution with a location parameter of zero and a rate parameter β (scale parameter $\alpha = 1/\beta$). These two limiting conditions are supported.

Usage

```
cdftexp(x, para)
```

Arguments

x	A real value vector.
para	The parameters from partexp or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Vogel, R.M., Hosking, J.R.M., Elphick, C.S., Roberts, D.L., and Reed, J.M., 2008, Goodness of fit of probability distributions for sightings as species approach extinction: Bulletin of Mathematical Biology, DOI 10.1007/s11538-008-9377-3, 19 p.

See Also

pdftexp, quatexp, lmomtexp, partexp

Examples

```
cdftexp(50,partexp(vec2lmom(c(40,0.38), lscale=FALSE)))
## Not run:
F <- seq(0,1,by=0.001)
A <- partexp(vec2lmom(c(100, 1/2), lscale=FALSE))
x <- quatexp(F, A)
plot(x, cdftexp(x, A), pch=16, type='l')
by <- 0.01; lcvs <- c(1/3, seq(1/3+by, 1/2-by, by=by), 1/2)
reds <- (lcvs - 1/3)/max(lcvs - 1/3)
for(lcv in lcvs) {
    A <- partexp(vec2lmom(c(100, lcv), lscale=FALSE))
    x <- quatexp(F, A)
    lines(x, cdftexp(x, A), pch=16, col=rgb(reds[lcvs == lcv],0,0))
}

  # Vogel and others (2008) example sighting times for the bird
  # Eskimo Curlew, inspection shows that these are fairly uniform.
  # There is a sighting about every year to two.
  T <- c(1946, 1947, 1948, 1950, 1955, 1956, 1959, 1960, 1961,
         1962, 1963, 1964, 1968, 1970, 1972, 1973, 1974, 1976,
         1977, 1980, 1981, 1982, 1982, 1983, 1985)
  R <- 1945 # beginning of record
  S <- T - R
  lmr <- lmoms(S)
  PARcurlew <- partexp(lmr)
  # read the warning message and then force the texp to the
  # stationary process model (min(tau_2) = 1/3).
  lmr$ratios[2] <- 1/3
  lmr$lambdas[2] <- lmr$lambdas[1]*lmr$ratios[2]
  PARcurlew <- partexp(lmr)
  Xmax <- quatexp(1, PARcurlew)
  X <- seq(0,Xmax, by=.1)
  plot(X, cdftexp(X,PARcurlew), type="l")
  # or use the MVUE estimator
```

```
  TE <- max(S)*((length(S)+1)/length(S)) # Time of Extinction
  lines(X, punif(X, min=0, max=TE), col=2)
## End(Not run)
```

`cdftri` *Cumulative Distribution Function of the Asymmetric Triangular Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Asymmetric Triangular distribution given parameters (ν, ω, and ψ) computed by `partri`. The cumulative distribution function is

$$F(x) = \frac{(x-\nu)^2}{(\omega-\nu)(\psi-\nu)},$$

for $x < \omega$,

$$F(x) = 1 - \frac{(\psi-x)^2}{(\psi-\omega)(\psi-\nu)},$$

for $x > \omega$, and

$$F(x) = \frac{(\omega-\nu)}{(\psi-\nu)},$$

for $x = \omega$ where $x(F)$ is the quantile for nonexceedance probability F, ν is the minimum, ψ is the maximum, and ω is the mode of the distribution.

Usage

```
cdftri(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `partri` or `vec2par`.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

See Also

`pdftri`, `quatri`, `lmomtri`, `partri`

Examples

```
lmr <- lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52))
cdftri(50,partri(lmr))
```

cdfwak *Cumulative Distribution Function of the Wakeby Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Wakeby distribution given parameters (ξ, α, β, γ, and δ) computed by parwak. The cumulative distribution function has no explicit form, but the pdfwak (density) and quawak (quantiles) do.

Usage

```
cdfwak(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parwak or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfwak, quawak, lmomwak, parwak

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
cdfwak(50,parwak(lmr))
```

cdfwei *Cumulative Distribution Function of the Weibull Distribution*

Description

This function computes the cumulative probability or nonexceedance probability of the Weibull distribution given parameters (ζ, β, and δ) of the distribution computed by parwei. The cumulative distribution function is

$$F(x) = 1 - \exp(Y^{\delta}),$$

where Y is

$$Y = -\frac{x + \zeta}{\beta},$$

where $F(x)$ is the nonexceedance probability for quantile x, ζ is a location parameter, β is a scale parameter, and δ is a shape parameter.

The Weibull distribution is a reverse Generalized Extreme Value distribution. As result, the Generalized Extreme Value algorithms are used for implementation of the Weibull in this package. The relations between the Generalized Extreme Value parameters (ξ, α, and κ) are

$$\kappa = 1/\delta,$$

$$\alpha = \beta/\delta, \text{ and}$$

$$\xi = \zeta - \beta,$$

which are taken from Hosking and Wallis (1997).

In R, the cumulative distribution function of the Weibull distribution is `pweibull`. Given a Weibull parameter object `para`, the R syntax is `pweibull(x+para$para[1], para$para[3], scale=para$para[2])`. For the current implementation for this package, the reversed Generalized Extreme Value distribution is used `1-cdfgev(-x,para)`.

Usage

```
cdfwei(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parwei or vec2par.

Value

Nonexceedance probability (F) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pdfwei, quawei, lmomwei, parwei

Examples

```
# Evaluate Weibull deployed here and within R (pweibull)
lmr <- lmoms(c(123,34,4,654,37,78))
WEI <- parwei(lmr)
F1  <- cdfwei(50,WEI)
F2  <- pweibull(50+WEI$para[1],shape=WEI$para[3],scale=WEI$para[2])
if(F1 == F2) EQUAL <- TRUE

# The Weibull is a reversed generalized extreme value
Q <- sort(rlmomco(34,WEI)) # generate Weibull sample
lm1 <- lmoms(Q)    # regular L-moments
lm2 <- lmoms(-Q)   # L-moment of negated (reversed) data
WEI <- parwei(lm1) # parameters of Weibull
GEV <- pargev(lm2) # parameters of GEV
F <- nonexceeds()  # Get a vector of nonexceedance probs
plot(pp(Q),Q)
lines(cdfwei(Q,WEI),Q,lwd=5,col=8)
lines(1-cdfgev(-Q,GEV),Q,col=2) # line overlaps previous
```

---

check.fs *Check Vector of Nonexceedance Probabilities*

Description

This function checks that a nonexceedance probability (F) is in the $0 \leq F \leq 1$ range. It does not check that the distribution specified by parameters for $F = 0$ or $F = 1$ is valid. End point checking is left to additional internal checks within the functions implementing the distribution. The function is intended for internal use to build a flow of logic throughout the distribution functions. Users are not anticipated to need this function themselves. The check.fs function is separate because of the heavy use of the logic across a myriad of functions in **lmomco**.

Usage

```
check.fs(fs)
```

Arguments

fs	A vector of nonexceedance probablity values.

Value

TRUE	The nonexceedance probabilities are valid.
FALSE	The nonexceedance probabilities are invalid.

Author(s)

W.H. Asquith

See Also

quaaep4, quaaep4kapmix, quacau, quaemu, quaexp, quagam, quagep, quagev, quagld, quaglo, quagno, quagov, quagpa, quagum, quakap, quakmu, quakur, qualap, qualmrq, qualn3, quanor, quape3, quaray, quarevgum, quarice, quasla, quast3, quatexp, quawak, quawei

Examples

```
F <- c(0.5,0.7,0.9,1.1)
if(check.fs(F) == FALSE) cat("Bad nonexceedances 0<F<1\n")
```

---

check.pdf *Check and Potentially Graph Probability Density Functions*

Description

This convenience function checks that a given probability density function (pdf) from **lmomco** appears to numerically be valid. By definition a pdf function must integrate to unity. This function permits some flexibility in the limits of integration and provides a high-level interface from graphical display of the pdf.

Usage

```
check.pdf(pdf, para, lowerF=0.001, upperF=0.999,
                 eps=0.02, verbose=FALSE, plot=FALSE, plotlowerF=0.001,
                 plotupperF=0.999, ...)
```

Arguments

pdf	A probability density function from **lmomco**.
lowerF	The lower bounds of nonexceedance probability for the numerical integration.
upperF	The upper bounds of nonexceedance probability for the numerical integration.
para	The parameters of the distribution.
eps	An error term expressing allowable error (deviation) of the numerical integration from unity. (If that is the objective of the call to the check.pdf function.)
verbose	Is verbose output desired?
plot	Should a plot (polygon) of the pdf integration be produce?

`plotlowerF`	Alternative lower limit for the generation of the curve depicting the `pdf` function.
`plotupperF`	Alternative upper limit for the generation of the curve depicting the `pdf` function.
`...`	Additional arguments that are passed onto the R function `integration` function.

Value

An R `list` structure is returned

`isunity`	Given the `eps` is `F` close enough.
`F`	The numerical integration of `pdf` from `lowerF` to `upperF`.

Author(s)

W.H. Asquith

Examples

```
lmrg <- vec2lmom(c( 100, 40, 0.1)) # Arbitrary L-moments
lmrw <- vec2lmom(c(-100, 40,-0.1)) # Reversed Arbitrary L-moments
gev  <- pargev(lmrg) # parameters of Generalized Extreme Value distribution
wei  <- parwei(lmrw) # parameters of Weibull distribution

# The Weibull is a reversed GEV and plots in the following examples show this.
# Two examples that should integrate to "unity" given default parameters.
layout(matrix(c(1,2), 2, 2, byrow = TRUE), respect = TRUE)
check.pdf(pdfgev,gev,plot=TRUE)
check.pdf(pdfwei,wei,plot=TRUE)
```

`claudeprecip` *Annual Maximum Precipitation Data for Claude, Texas*

Description

Annual maximum precipitation data for Claude, Texas

Usage

```
data(claudeprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(claudeprecip)
summary(claudeprecip)
```

---

clearforkporosity   *Porosity Data*

Description

Porosity (fraction of void space) from neutron-density, well log for 5,350–5,400 feet below land surface for Permian Age Clear Fork formation, Ector County, Texas.

Usage

```
data(clearforkporosity)
```

Format

A data frame with

POROSITY The pre-sorted porosity data.

Details

Although the porosity data was collected at about 1-foot intervals, these intervals are not provided in the data frame. Further, the porosity data has been sorted to disrupt the specific depth to porosity relation to remove the proprietary nature of the original data.

---

cmlmomco   *Conditional Mean Residual Quantile Function of the Distributions*

Description

This function computes the Conditional Mean Residual Quantile Function for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair et al. (2013, p. 68) as

$$\mu(u) = \frac{1}{1-u}\int_u^1 x(p)\;\mathrm{d}p,$$

where $\mu(u)$ is the conditional mean for nonexceedance probability u. The $\mu(u)$ is the expectation $\mathrm{E}[X|X > x]$. The $\mu(u)$ also is known as the *vitality function*. Details can be found in Nair et al. (2013, p. 68) and Kupka and Loo (1989). Mathematically, the vitality function simply is

$$\mu(u) = M(u) + x(u),$$

where $M(u)$ is the mean residual quantile function (rmlmomco), $x(u)$ is a constant for $x(F = u)$.

Usage

```
cmlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from lmom2par or vec2par.

Value

Conditional mean residual value for F or conditional mean life for F.

Author(s)

W.H. Asquith

References

Kupka, J., and Loo, S., 1989, The hazard and vitality measures of ageing: Journal of Applied Probability, v. 26, pp. 532–542.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rmlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
qlmomco(0.5, A)  # The median lifetime = 1261 days
rmlmomco(0.5, A) # The average remaining life given survival to the median = 861 days
cmlmomco(0.5, A) # The average total life given survival to the median = 2122 days

# Now create with a nonzero origin
A <- vec2par(c(100, 2649, 2.11), type="gov") # so set lower bounds = 0.0
qlmomco(0.5, A)  # The median lifetime = 1361 days
rmlmomco(0.5, A) # The average remaining life given survival to the median = 861 days
cmlmomco(0.5, A) # The average total life given survival to the median = 2222 days

# Mean life (mu), which shows up in several expressions listed under rmlmomco.
mu1 <- cmlmomco(0,A)
mu2 <- par2lmom(A)$lambdas[1]
mu3 <- reslife.lmoms(0,A)$lambdas[1]
# Each mu is 1289.051 days.
```

`cvm.test.lmomco` *Cramér–von Mises Test for Goodness-of-Fit*

Description

The Cramér–von Mises test for goodness-of-fit is implemented for the order statistics $x_{1:n} \leq x_{i:n} \leq x_{n:n}$ of a sample of size n. Define the test statistic (Csörgő and Faraway, 1996) as

$$\omega^2 = \frac{1}{12n} + \sum_{i=1}^{n}\left[\frac{2i-1}{2n} - F_\theta(x_i)\right],$$

where $F_\theta(x)$ is the cumulative distribution function (continuous) for some distribution having parameters θ. If the value for ω^2 is larger than some critical value, reject the null hypothesis. The null hypothesis is that F is the function specified by θ, while the alternative hypothesis is that F is some other function.

Usage

```
cvm.test.lmomco(x, para1, ...)
```

Arguments

`x`	A vector of data values.
`para1`	The parameters of the distribution.
`...`	Additional arguments to pass to `par2cdf`.

Details

The above definition for ω^2 as the Cramér–von Mises test statistic is consistent with the notation in Csörgő and Faraway (1996) as well as that in package **goftest**. Depending on how the null distribution is defined by other authors and attendant notation, the Cramér–von Mises statistic can be branded as $T = n\omega^2$. The null distribution herein requires just ω^2 and the sample size is delivered separately into the cumulative distribution function:

```
goftest::pCvM(omega.sq, n=n, lower.tail=FALSE)
```

Value

An R `list` is returned.

`null.dist`	The null distribution, which is an echoing of the `para` argument, which recall for **lmomco** that is contains the distribution abbreviation.
`text`	The string "Cramer–von Mises test of goodness-of-fit".
`statistic`	The ω^2 as defined above (see **Note**).
`p.value`	The p-value computed from the `pCvM()` function from the **goftest** package for the null distribution of the test statistic.
`source`	An attribute identifying the computational source of the L-moments: "cvm.test.lmomco".

Note

An example of coverage probabilities demonstrating the differences in what the p-values mean on whether the parent is known or the "parent" is coming from the sample. The p-values are quite different and inference has subtle differences. In ensemble, comparing the test statistic amongst distribution choices might be more informative than a focus on p-values being below a critical alpha.

```
parent <- vec2par(c(20, 120), type="gam"); nsim <- 10000
pp <- nn <- ee <- rep(NA,nsim)
for(i in 1:nsim) {
  x <- rlmomco(56, parent); lmr <- lmoms(x)
  pp[i] <- cvm.test.lmomco(x,          parent          )$p.value
  nn[i] <- cvm.test.lmomco(x, lmom2par(lmr, type="nor"))$p.value
  ee[i] <- cvm.test.lmomco(x, lmom2par(lmr, type="exp"))$p.value
}
message("GAMMA PARENT KNOWN     'rejection rate'=", sum(pp < 0.05)/nsim)
message("ESTIMATED NORMAL       'rejection rate'=", sum(nn < 0.05)/nsim)
message("ESTIMATED EXPONENTIAL  'rejection rate'=", sum(ee < 0.05)/nsim)
```

The rejection rate for the Gamma is about 5 percent, which matches the 0.05 specified in the conditional. The Normal is about zero, and the Exponential is about 21 percent. The fitted Normal almost always passes for the real parent, though Gamma, for the sample size and amount of L-skewness involved. The Exponential does not. This illustrates that the p-value can be misleading in the single-sample version of this test. Thus, when fit by parameters from the sample, the test statistic is nearly always smaller than the one for a prespecified set of parameters. The significance level will be smaller than intended.

Author(s)

W.H. Asquith

References

Csörgő, S., and Faraway, J.J., 1996, The exact and asymptotic distributions of Cramér–von Mises statistics: Journal of the Royal Statistical Society, Series B, v. 58, pp. 221–234.

See Also

lmrdia

Examples

```
# An example in which the test is conducted on a sample but the parent is known.
# This will lead to more precise inference than if the sample parameters are used.
mu <- 120; sd <- 25; para <- vec2par(c(120, 25), type="nor")
x <- rnorm(56, mean=mu, sd=sd)
T1 <- cvm.test.lmomco(x, para)$statistic
T2 <- goftest::cvm.test(x, null="pnorm", mean=mu, sd=sd)$statistic
message("Cramer--von Mises: T1=", round(T1, digits=6), " and T2=", round(T2, digits=6))
```

dat2bernqua — *Observed Data to Empirical Quantiles through Bernstein or Kantorovich Polynomials*

Description

The empirical quantile function can be "smoothed" (Hernández-Maldonado and others, 2012, p. 114) through the Kantorovich polynomial (Muñoz-Pérez and Fernández-Palacín, 1987) for the sample order statistics $x_{k:n}$ for a sample of size n by

$$\tilde{X}_n(F) = \frac{1}{2}\sum_{k=0}^{n}(x_{k:n} + x_{(k+1):n})\binom{n}{k}F^k(1-F)^{n-k},$$

where F is nonexceedance probability, and $(n\ k)$ are the binomial coefficients from the R function `choose()`, and the special situations for $k = 0$ and $k = n$ are described within the Note section. The form for the Bernstein polynomial is

$$\tilde{X}_n(F) = \sum_{k=0}^{n+1}(x_{k:n})\binom{n+1}{k}F^k(1-F)^{n+1-k}.$$

There are subtle differences between the two and `dat2bernqua` function supports each. Readers are also directed to the *Special Attention* section.

Turnbull and Ghosh (2014) consider through the direction of a referee and recommendation of $p = 0.05$ by that referee (and credit to ideas by de Carvalho [2012]) that the support of the probability density function for the Turnbull and Ghosh (2014) study of Bernstein polynomials can be computed letting $\alpha = (1-p)^{-2} - 1$ by

$$\left(x_{1:n} - (x_{2:n} - x_{1:n})/\alpha,\ x_{n:n} + (x_{n:n} - x_{n-1:n})/\alpha\right),$$

for the minimum and maximum, respectively. Evidently, the original support considered by Turnbull and Ghosh (2014) was

$$\left(x_{1:n} - \lambda_2\sqrt{\pi/n},\ x_{n:n} + \lambda_2\sqrt{\pi/n}\right),$$

for the minimum and maximum, respectively and where the standard deviation is estimated in the function using the 2nd L-moment as $s = \lambda\sqrt{\pi}$.

The p is referred to by this author as the "p-factor" this value has great influence in the estimated support of the distribution and therefore distal-tail estimation or performance is sensitive to the value for p. General exploratory analysis suggests that the p can be optimized based on information external or internal to the data for shape restrained smoothing. For example, an analyst might have external information as to the expected L-skew of the phenomenon being studied or could use the sample L-skew of the data (internal information) for shape restraint (see `pfactor.bernstein`).

An alternative formula for smoothing is by Cheng (1995) and is

$$\tilde{X}_n^{\mathrm{Cheng}}(F) = \sum_{k=1}^{n} x_{k:n}\binom{n-1}{k-1}F^{k-1}(1-F)^{n-k}.$$

Usage

```
dat2bernqua(f, x, bern.control=NULL,
                  poly.type=c("Bernstein", "Kantorovich", "Cheng", "Parzen",
                              "bernstein", "kantorovich", "cheng", "parzen"),
                  bound.type=c("none", "sd", "Carv", "either", "carv"),
                  fix.lower=NULL, fix.upper=NULL, p=0.05, listem=FALSE)
```

Arguments

`f`	A vector of nonexceedance probabilities F.
`x`	A vector of data values.
`bern.control`	A `list` that holds `poly.type`, `bound.type`, `fix.lower`, and `fix.upper`. And this list will supersede the respective values provided as separate arguments.
`poly.type`	The Bernstein or Kantorovich polynomial will be used. The two are quite closely related. Or the formula by Cheng (1995) will be used and `bound.type`, `fix.lower`, `fix.upper`, and `p` are not applicable. Or the formula credited by Nair et al. (2013, p. 17) to Parzen (1979) will be used.
`bound.type`	Triggers to the not involve alternative supports (`"none"`) then the minimum and maximum are used unless already provided by the `fix.lower` or `fix.upper`, the support based `"sd"` on the standard deviation, the support `"Carv"` based on the arguments of de Carvalho (2012), or `"either"` method.
`fix.lower`	For $k = 0$, either the known lower bounds is used if provided as non `NULL` or the observed minimum of the data. If the minimum of the data is less than the `fix.lower`, a warning is triggered and `fix.lower` is set to the minimum. Following Turnbull and Ghosh (2014) to avoid bounds that are extremely lower than the data, it will use the estimated lower bounds by the method `"sd"`, `"Carv"`, or `"either"` if these bounds are larger than the provided `fix.lower`.
`fix.upper`	For $k = n$, either the known upper bounds is used if provided as non `NULL` or the observed maximum of the data; If the maximum of the data is less than the `fix.upper`, a warning is triggered and `fix.upper` is set to the maximum.
`p`	A small probability value to serve as the p in the `"Carv"` support computation. The default is recommended as mentioned above. The program will return `NA` if $10^{-6} < p \geq (1 - 10^{-6})$ is not met. The value `p` is the "p-factor" p.
`listem`	A logical controlling whether (1) a vector of $\tilde{X}_n(F)$ is returned or (2) a list containing $\tilde{X}_n(F)$, the `f`, original sample size n of the data, the de Carvalho probability `p` (whether actually used internally or not), and both `fix.lower` and `fix.upper` as computed within the function or provided (less likely) by the function arguments.

Details

Yet another alternative formula for smoothing if by Parzen (1979) and known as the "Parzen weighting method" is

$$\tilde{X}_n^{\mathrm{Parzen}}(F) = n\left(\frac{r}{n} - F\right)x_{r-1:n} + n\left(F - \frac{r-1}{n}\right)x_{r:n},$$

where $(r-1)/n \le F \le (r/n)$ for $r = 1, 2, \ldots, n$ and $x_{0:n}$ is taken as either the minimum of the data $(\min(x))$ or the lower bounds `fix.lower` as externally set by the user. For protection, the minimum of $(\min(x)$, `fix.lower`$)$ is formally used. If the Parzen method is used, the only arguments considered are `poly.type` and `fix.lower`; all others are ignored including the `f` (see Value section). The user does not actually have to provide `f` in the arguments but a place holder such as `f=NULL` is required; internally the Parzen method takes over full control. The Parzen method in general is not smooth and not recommended like the others that rely on a polynomial basis function. Further the Parzen method has implicit asymmetry in the estimated F. The method has $F = 0$ and $F < 1$ on output, but if the data are reversed, then the method has $F > 0$ and $F = 1$. Data reversal is made in `-X` as this example illustrates:

```
X <- sort(rexp(30))
P <- dat2bernqua(f=NULL,  X, poly.type="Parzen")
R <- dat2bernqua(f=NULL, -X, poly.type="Parzen")
plot(pp(X, a=0.5), X, xlim=c(0, 1)) # Hazen plotting position to
lines(  P$f,  P$x, col="red" )      # basically split the horizontal
lines(1-R$f, -R$x, col="blue")      # differences between blue and red.
```

Value

An R `vector` is returned unless the Parzen weighting method is used and in that case an R `list` is returned with elements `f` and `x`, which respectively are the F values as shown in the formula and the $\tilde{X}_n^{\mathrm{Parzen}}(F)$.

Special Attention

The limiting properties of the Bernstein and Kantorovich polynomials differ. The Kantorovich polynomial uses the average of the largest (smallest) value and the respective outer order statistics ($x_{n+1:n}$ or $x_{0:n}$) unlike the Bernstein polynomial whose $F = 0$ or $F = 1$ are purely a function of the outer order statistics. Thus, the Bernstein polynomial can attain the `fix.lower` and(or) `fix.upper` whereas the Kantorovich fundamentally can not. For a final comment, the function `dat2bernquaf` is an inverse of `dat2bernqua`.

Implentation Note

The function makes use of R functions `lchoose` and `exp` and logarithmic expressions, such as $(1-F)^{(n-k)} \rightarrow (n-k)\log(1-F)$, for numerical stability for large sample sizes.

Note

Muñoz-Pérez and Fernández-Palacín (1987, p. 391) describe what to do with the condition of $k = 0$ but seemingly do not comment on the condition of $k = n$. There is no 0th-order statistic nor is there a $k > n$ order statistic. Muñoz-Pérez and Fernández-Palacín (1987) bring up the notion of a natural minimum for the data (for example, data that must be positive, `fix.lower = 0` could be set). Logic dictates that a similar argument must be made for the maximum to keep a critical error from occurring if one tries to access the not plausible `x[n+1]`-order statistic. Lastly, the argument names `bound.type`, `fix.lower`, and `fix.upper` mimic, as revisions were made to this function in December 2013, the nomenclature of software for probability density function smoothing by Turnbull and Ghosh (2014). The `dat2bernqua` function was originally added to **lmomco** in May 2013 prior to the author learning about Turnbull and Ghosh (2014).

Lastly, there can be many practical situations in which transformation is desired. Because of the logic structure related to how `fix.lower` and `fix.upper` are determined or provided by the user, it is highly recommended that this function not internally handle transformation and detransformation. See the second example for use of logarithms.

Author(s)

W.H. Asquith

References

Cheng, C., 1995, The Bernstein polynomial estimator of a smooth quantile function: Statistics and Probability Letters, v. 24, pp. 321–330.

de Carvalho, M., 2012, A generalization of the Solis-Wets method: Journal of Statistical Planning and Inference, v. 142, no. 3, pp. 633–644.

Hernández-Maldonado, V., Díaz-Viera, M., and Erdely, A., 2012, A joint stochastic simulation method using the Bernstein copula as a flexible tool for modeling nonlinear dependence structures between petrophysical properties: Journal of Petroleum Science and Engineering, v. 90–91, pp. 112–123.

Muñoz-Pérez, J., and Fernández-Palacín, A., 1987, Estimating the quantile function by Bernstein polynomials: Computational Statistics and Data Analysis, v. 5, pp. 391–397.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

Turnbull, B.C., and Ghosh, S.K., 2014, Unimodal density estimation using Bernstein polynomials: Computational Statistics and Data Analysis, v. 72, pp. 13–29.

Parzen, E., 1979, Nonparametric statistical data modeling: Journal American Statistical Association, v. 75, pp. 105–122.

See Also

`lmoms.bernstein`, `pfactor.bernstein`, `dat2bernquaf`

Examples

```
# Compute smoothed extremes, quartiles, and median
# The smoothing seems to extend to F=0 and F=1.
set.seed(1); X <- exp(rnorm(20)); F <- c(0, .25, .50, .75, 1)
dat2bernqua(F, X, bound.type="none",   listem=TRUE)$x
dat2bernqua(F, X, bound.type="Carv",   listem=TRUE)$x
dat2bernqua(F, X, bound.type="sd",     listem=TRUE)$x
dat2bernqua(F, X, bound.type="either", listem=TRUE)$x
dat2bernqua(F, X, bound.type="sd",     listem=TRUE, fix.lower=0)$x

## Not run:
X <- sort(10^rnorm(20)); F <- nonexceeds(f01=TRUE)
plot(qnorm(pp(X)), X, xaxt="n", xlab="", ylab="QUANTILE", log="y")
add.lmomco.axis(las=2, tcl=0.5, side.type="NPP", twoside=TRUE)
lines(qnorm(F),     dat2bernqua(F,    X,  bound.type="sd"), col="red", lwd=2)
lines(qnorm(F), exp(dat2bernqua(F,log(X), bound.type="sd"))) #
```

```
## End(Not run)

## Not run:
X <- exp(rnorm(20)); F <- seq(0.001, 0.999, by=.001)
dat2bernqua(0.9, X, poly.type="Bernstein",   listem=TRUE)$x
dat2bernqua(0.9, X, poly.type="Kantorovich", listem=TRUE)$x
dat2bernqua(0.9, X, poly.type="Cheng",       listem=TRUE)$x
plot(pp(X), sort(X), log="y", xlim=range(F))
lines(F, dat2bernqua(F, X, poly.type="Bernstein"  ), col="red"  )
lines(F, dat2bernqua(F, X, poly.type="Kantorovich"), col="green")
lines(F, dat2bernqua(F, X, poly.type="Cheng"      ), col="blue" ) #
## End(Not run)

## Not run:
X <- exp(rnorm(20)); F <- nonexceeds()
plot(pp(X), sort(X))
lines(F, dat2bernqua(F,X, bound.type="sd", poly.type="Bernstein"))
lines(F, dat2bernqua(F,X, bound.type="sd", poly.type="Kantorovich"), col=2) #
## End(Not run)

## Not run:
X <- rnorm(25); F <- nonexceeds()
Q <- dat2bernqua(F, X) # the Bernstein estimates
plot( F, dat2bernqua(F, X, bound.type="Carv"), type="l"   )
lines(F, dat2bernqua(F, X, bound.type="sd"),   col="red"  )
lines(F, dat2bernqua(F, X, bound.type="none"), col="green")
points(pp(X),      sort(X), pch=16, cex=.75,   col="blue" ) #
## End(Not run)

## Not run:
set.seed(13)
par <- parkap(vec2lmom(c(1, .5, .4, .2)))
F <- seq(0.001, 0.999, by=0.001)
X <- sort(rlmomco(100, par))
pp <- pp(X)
pdf("lmomco_example_dat2bernqua.pdf")
plot(qnorm(pp(X)), dat2bernqua(pp, X), col="blue", pch=1,
     ylim=c(0,qlmomco(0.9999, par)))
lines(qnorm(F), dat2bernqua(F, sort(X)), col="blue")
lines(qnorm(F),     qlmomco(F,     par), col="red" )
sampar  <- parkap(lmoms(X))
sampar2 <- parkap(lmoms(dat2bernqua(pp, X)))
lines( qnorm(pp(F)), qlmomco(F, sampar ), col="black")
lines( qnorm(pp(F)), qlmomco(F, sampar2), col="blue", lty=2)
points(qnorm(pp(X)), X, col="black", pch=16)
dev.off() #
## End(Not run)
```

`dat2bernquaf` *Equivalent Nonexceedance Probability for a Given Value through Observed Data to Empirical Quantiles through Bernstein or Kantorovich Polynomials*

Description

This function computes an equivalent nonexceedance probability F of a single value x for the sample data set ($\hat{X}$) through inversion of the empricial quantile function as computable through Bernstein or Kantorovich Polynomials by the `dat2bernqua` function.

Usage

```
dat2bernquaf(x, data, interval=NA, ...)
```

Arguments

`x`	A scalar value for which the equivalent nonexceedance probability F through the function `dat2bernqua` is to be computed.
`data`	A vector of data values that directly correspond to the argument `x` in the function `dat2bernqua`.
`interval`	The search interval. If `NA`, then $[1/(n+1), 1-1/(n+1)]$ is used. If `interval` is a single value a, then the interval is computed as $[a, 1-a]$.
`...`	Additional arguments passed to `dat2bernqua` through the `uniroot()` function in R.

Details

The basic logic is thus. The $\hat{X}$ in conjunction with the settings for the polynomials provides the empirical quantile function (EQF). The `dat2bernquaf` function then takes the EQF (through dynamic recomputation) and seeks a root for the single value also given.

The critical piece likely is the search interval, which can be modified by the `interval` argument if the internal defaults are not sufficient. The default interval is determined as the first and last Weibull plotting positions of $\hat{X}$ having a sample size n: $[1/(n+1), 1-1/(n+1)]$. Because the `dat2bernqua` function has a substantial set of options that control how the empirical curve is (might be) extrapolated beyond the range of $\hat{X}$, it is difficult to determine an always suitable interval for the rooting. However, it should be considered obvious that the result is more of an interpolation if $F(x)$ is within $F \in [1/(n+1), 1-1/(n+1)]$ and increasingly becomes an accurate interpolation as $F(x) \rightarrow 1/2$ (the median).

If the value x is too far beyond the data or if the search interval is not sufficient then the following error will be triggered:

```
Error in uniroot(afunc, interval, ...) :
  f() values at end points not of opposite sign
```

The Examples section explores this aspect.

Value

An R `list` is returned.

`x`	An echoing of the x value via the `x` argument.
`f`	The equivalent nonexceedance probability $F(x\|\hat{X})$.
`interval`	The search interval of F used.
`afunc.root`	Corresponds to the `f.root` element returned by the `uniroot()` function.
`iter`	Corresponds to the `iter` element returned by the `uniroot()` function.
`estim.prec`	Corresponds to the `estim.prec` element returned by the `uniroot()` function.
`source`	An attribute identifying the computational source: "dat2bernquaf".

Author(s)

W.H. Asquith

See Also

`dat2bernqua`

Examples

```
dat2bernquaf(6, c(2,10)) # median 1/2 of 2 and 10 is 6 (trivial and fast)
## Not run:
set.seed(5135)
lmr <- vec2lmom(c(1000, 400, 0.2, 0.3, 0.045))
par <- lmom2par(lmr, type="wak")
Q   <- rlmomco(83, par) # n = 83 and extremely non-Normal data
lgQ <- max(Q) # 5551.052 by theory
dat2bernquaf(median(Q), Q)$f  # returns 0.5100523 (nearly 1/2)
dat2bernquaf(lgQ,   Q)$f                   # unable to root
dat2bernquaf(lgQ,   Q, bound.type="sd")$f  # unable to root
itf <- c(0.5, 0.99999)
f <- dat2bernquaf(lgQ, Q, interval=itf, bound.type="sd")$f
print(f) # F=0.9961118
qlmomco(f, par) # 5045.784 for the estimate F=0.9961118
# If we were not using the maximum and something more near the center of the
# distribution then that estimate would be closer to qlmomco(f, par).
# You might consider lqQ <- qlmomco(0.99, Q) # theoretical 99th percentile and
# let the random seed wander and see the various results.
## End(Not run)
```

disfitgovloc *Fit a Govindarajulu Distribution to Bounds and Location*

Description

Fits a *Govindarajulu* distribution to specified lower and upper bounds and a given location measure (either mean and median). Fitting occurs through 3-dimensional minimization using the `optim` function. Objective function forms are either root mean-square error (RMSE) or mean absolute deviation (MAD), and the objective functions are expected to result in slightly different estimates of distribution parameters. The RMSE form (σ_RMSE) is defined as

$$\sigma_\mathrm{RMSE} = \left[\frac{1}{3}\sum_{i=1}^{3}\bigl[x_i - \hat{x}_i\bigr]^2\right]^{1/2},$$

where x_i is a vector of the targeted lower bounds (`lwr` argument), location measure (`loc` argument), and upper bounds (`upr` argument), and $\hat{x}_i$ is a similar vector of Govindarajulu properties for "current" iteration of the optimization. Similarly, the MAD form (σ_MAD) is defined as

$$\sigma_\mathrm{MAD} = \frac{1}{3}\sum_{i=1}^{3} \mid x_i - \hat{x}_i \mid .$$

The premise of this function is that situations might exist in practical applications wherein the user has an understanding or commitment to certain bounding conditions of a distribution. The user also has knowledge of a particular location measure (the mean or median) of a distribution. The bounded nature of the Govindarajulu might be particularly of interest because the quantile function (`quagov`) is explicit. The curvatures that the distribution can attain also provide it more flexibility to fitting to a given location measure than say the *Triangular* distribution (`quatri`).

Usage

```
disfitgovloc(x=NULL, loc=NULL, lwr=0, upr=NA, init.para=NULL,
             loctype=c("mean", "median"), objfun=c("rmse", "mad"),
             ptransf=function(p) return(log(p)),
             pretransf=function(p) return(exp(p)),
             silent=TRUE, verbose=FALSE, ...)
```

Arguments

- `x` Optional vector to help guide the initial parameter estimates for the optimization, if given and if `loc=NULL`, then `loc` by `loctype` will be computed from the `x`.
- `loc` Optional value for the location statistic, which if not given will be computed from mean or median of the `x`. The `loc` however can also be given if an `x` is given and at which point the user's setting prevails.
- `lwr` Lower bounds for the distribution with default supposing that most often positive domain bounds might be of interest.

`upr`	Upper bounds for the distribution, which must be specified.
`init.para`	Optional initial values for the parameters used for starting values for the `optim` function. If this argument is not set nor is `x`, then an unrigorous attempt is made to guess at the initial parameters using heuristics and the triangular quantile function (because the triangle is trivial and also bounded) (see sources).
`loctype`	The type of location measure constraint.
`objfun`	The form of the objective function as previously described.
`ptransf`	The parameter transformation function that is useful to guide the optimization run. The distribution requires its second and third parameters to be nonzero without constraint on the first parameter; however, the default treats the first parameter as also nonzero. This is potentially suboptimal for some situations (see **Examples**).
`pretransf`	The parameter retransformation function that is useful to guide the optimization run. The distribution requires its second and third parameters to be nonzero without constraint on the first parameter; however, the default treats the first parameter as also nonzero. This is potentially suboptimal for some situations (see **Examples**).
`silent`	A logical to silence the `try()` function wrapping the `optim()` function.
`verbose`	A logical to trigger verbose output within the objective function.
`...`	Additional arguments to pass to the `optim` function.

Details

Support of the Govindarajulu for the optimized parameter set is computed by internally and reported as part of the returned values. This enhances the documentation a bit more—the computed parameters might not always have full convergence and result in slightly difference bounds than targeted. Finally, this function was developed using some heredity to `disfitqua`.

Value

An R `list` is returned. This list should contain at least the following items.

`type`	The type of distribution in three character (minimum) format.
`para`	The parameters of the Govindarajulu distribution.
`source`	Attribute specifying source of the parameters.
`supdist`	A list of confirming the distribution support from `quagov(c(0,1), gov)` where `gov` are the final computed parameters before return.
`init.para`	A vector of the initial parameters actually passed to the `optim` function to serve only as a reminder.
`optim`	The returned `list` of the `optim()` function.
`message`	Helpful messages on the computations.

Author(s)

W.H. Asquith

See Also

disfitqua, quagov

Examples

```
# EXAMPLE 1 --- Example of strictly positive domain.
disfitgovloc(loc=125, lwr=99, upr=175, loctype="mean")$para
#        xi     alpha      beta
# 99.000000 76.000000  3.846154
# These parameters have a lmomgov()$lambdas[1] mean of 124.9999999.

# EXAMPLE 2 --- Operations spanning zero and revision to the default parameter
# transform functions. Testing indicates that these, ideally align to need of
# the Govindarajulu, such do not work for all strictly positive domain, which
# led to a decision to have the defaults different than this example.
disfitgovloc(loc=100, lwr=-99, upr=175, loctype="median",
               ptransf=function(p) c(p[1], log(p[2:3])),
             pretransf=function(p) c(p[1], exp(p[2:3])))$para
#         xi        alpha         beta
# -99.000002   274.000004   1.08815151

## Not run:
  # EXTENDED EXAMPLE 3
  r <- function(r) round(r, 1)
  X <- c(8751, 14507, 4061, 22056, 6330, 3130, 5180, 6700, 22409, 3380, 17902,
         8956,  4523, 1604,  4460, 4239, 3010, 9155, 5107, 4821,  5221, 20700)
  mu  <-   mean(X); med <- median(X)
  for(objfun in c("rmse", "mad")) {
    gov <- disfitgovloc(x=X,  loc=mu,  upr=41000, objfun=objfun, loctype="mean"    )
    message(objfun, ": seek   mean=", r(mu),
                    ", GOV   mean=",  r(lmomgov(gov)$lambdas[1]))
    gov <- disfitgovloc(x=X, loc=med,  upr=41000, objfun=objfun, loctype="median"  )
    message(objfun, ": seek median=", r(med),
                    ", GOV median=",  r(quagov(0.5, gov)))
  }
  for(objfun in c("rmse", "mad")) {
    gov <- disfitgovloc(x=NULL,  loc=mu,  upr=41000, objfun=objfun, loctype="mean"  )
    message(objfun, ": seek   mean=", r(mu),
                    ", GOV   mean=",  r(lmomgov(gov)$lambdas[1]) )
    gov <- disfitgovloc(x=NULL, loc=med,  upr=41000, objfun=objfun, loctype="median")
    message(objfun, ": seek median=", r(med),
                    ", GOV median=",  r(quagov(0.5, gov)))
  } # end of loop
  # *** That last message() : mad: seek median=5200.5, GOV median=5226.2
  print(gov$para) # 64.521326, 40935.479117, 4.740232 # last parameters in prior loop
  ngv <- vec2par( c(64.521326, 40935.479117, 4.740232), type="gov") # for reuse
  # We see (at least in testing) that the last message in the sequence shows that
  # the median is not recovered via the guessed at initial parameters, let us turn
  # the gov parameters back into disfitgovloc() as the initial parameters.
  mgv <- disfitgovloc(init.para=ngv, loc=med, upr=41000, objfun=objfun,loctype="median")
  message(objfun, ": seek median=", r(med),
                  ", GOV median=", r(quagov(0.5, mgv)))
```

```
  # *** BETTER FIT mad: seek median=5200.5, GOV median=5200.5
  print(mgv$para) # 1.227568, 40998.903644, 4.729768 # last parameters
  # So, conveniently in this example, we can see that there are cases wherein an
  # apparent convergence can be made even better. But, need to be aware that
  # feed fack a very good solution can in turn cause optim() itself to NULL out.
## End(Not run)

## Not run:
  # EXTENDED EXAMPLE 4 --- Continuing from the previous example
  FF    <- seq(0.001, 0.999, by=0.001)
  maxes <- as.integer(10^(seq(4, 5, by=0.02))); n <- length(maxes)
  for(max in maxes) {
    govA <- disfitgovloc(x=X,  loc=mu,     upr=max, loctype="mean"  , lwr=0)
    govB <- disfitgovloc(x=X,  loc=median, upr=max, loctype="median", lwr=0)
    plot( FF, quagov(FF, govA), col="red",  lwd=2, type="l", ylim=c(0, maxes[n]),
         xlab="Nonexceedance probability", ylab="Quantile of Govindarajulu",
         main=paste0("Maximum = ", max))
    lines(FF, quagov(FF, govB), col="blue", lwd=2); quagov(0.5, govB)
    legend("topleft", c("Govindarajulu constrained given mean (dashed red)",
                        "Govindarajulu constrained given median (dashed blue)",
                        "disfitgovloc() computed mean (red dot)",
                        "disfitgovloc() computed median (blue dot)"),
                   lwd=c( 2,  2, NA, NA), col=c("red", "blue"), inset=0.02,
                   pch=c(NA, NA, 16, 16), pt.cex=1.5, cex=0.9)
    abline(h=mu,  lty=2, col="red" ); abline(h=med, lty=2, col="blue")
    tmu <- lmomgov(govA)$lambdas[1]
    points(cdfgov(tmu, govA), tmu, cex=1.5, pch=16, col="red" )
    points(0.5, quagov(0.5, govB), cex=1.5, pch=16, col="blue")
  } # end of loop
## End(Not run)
```

`disfitqua` *Fit a Distribution using Minimization of Available Quantiles*

Description

This function fits a distribution to available quantiles (or irregular quantiles) through n-dimensional minimization using the `optim` function. Objective function forms are either root mean-square error (RMSE) or mean absolute deviation (MAD), and the objective functions are expected to result in slightly different estimates of distribution parameters. The RMSE form (σ_RMSE) is defined as

$$\sigma_\mathrm{RMSE} = \biggl[\frac{1}{m}\sum_{i=1}^{m}\bigl[x_o(f_i) - \hat{x}(f_i)\bigr]^2\biggr]^{1/2}\mathrm{,}$$

where m is the length of the vector of *o*bserved quantiles $x_o(f_i)$ for nonexceedance probability f_i for $i \in 1, 2, \cdots, m$, and $\hat{x}(f_i)$ for $i \in 1, 2, \cdots, m$ are quantile estimates based on the "current" iteration of the parameters for the selected distribution having n parameters for $n \le m$. Similarly, the MAD form (σ_MAD) is defined as

$$\sigma_\mathrm{MAD} = \frac{1}{m}\sum_{i=1}^{m} \mid x_o(f_i) - \hat{x}(f_i) \mid \mathrm{.}$$

The `disfitqua` function is not intended to be an implementation of the *method of percentiles* but rather is intended for circumstances in which the available quantiles are restricted to either the left or right tails of the distribution. It is evident that a form of the method of percentiles however could be pursued by `disfitqua` when the length of $x(f)$ is equal to the number of distribution parameters ($n = m$). The situation of $n < m$ however is thought to be the most common application.

The right-tail restriction is the general case in flood-peak hydrology in which the median and select quantiles greater than the median can be available from empirical studies (e.g. Asquith and Roussel, 2009) or rainfall-runoff models. The available quantiles suit engineering needs and thus left-tail quantiles simply are not available. This circumstance might appear quite unusual to users from most statistical disciplines but quantile estimates can exist from regional study of observed data. The **Examples** section provides further motivation and discussion.

Usage

```
disfitqua(x, f, objfun=c("rmse", "mad"),
                init.lmr=NULL, init.para=NULL, type=NA,
                ptransf=  function(t) return(t),
                pretransf=function(t) return(t), verbose=FALSE, ... )
```

Arguments

`x`	The quantiles $x_o(f)$ for the nonexceedance probabilities in `f`.
`f`	The nonexceedance probabilities f of the quantiles $x_o(f)$ in `x`.
`objfun`	The form of the objective function as previously described.
`init.lmr`	Optional initial values for the L-moments from which the initial starting parameters for the optimization will be determined. The optimizations by this function are not performed on the L-moments during the optimization. The form of `init.lmr` is that of an L-moment object from the **lmomco** package (e.g. `lmoms`).
`init.para`	Optional initial values for the parameters used for starting values for the `optim` function. If this argument is not set nor is `init.lmr`, then unrigorous estimates of the mean λ_1 and L-scale λ_2 are made from the available quantiles, higher L-moment ratios τ_r for $r \geq 3$ are set to zero, and the L-moments converted to the initial parameters.
`type`	The distribution type specified by the abbreviations listed under `dist.list`.
`ptransf`	An optional parameter transformation function (see **Examples**) that is useful to guide the optimization run. For example, suppose the first parameter of a three parameter distribution resides in the positive domain, then `ptransf(t) = function(t) c(log(t[1]), t[2], t[3])`.
`pretransf`	An optional parameter retransformation function (see **Examples**) that is useful to guide the optimization run. For example, suppose the first parameter of a three parameter distribution resides in the positive domain, then `pretransf(t) = function(t) c(exp(t[1]), t[2], t[3])`.
`verbose`	A logical switch on the verbosity of output.
`...`	Additional arguments to pass to the `optim` function.

Value

An R `list` is returned, and this `list` contains at least the following items:

`type`	The type of distribution in character format (see `dist.list`).
`para`	The parameters of the distribution.
`source`	Attribute specifying source of the parameters—"disfitqua".
`init.para`	A vector of the initial parameters actually passed to the `optim` function to serve only as a reminder.
`disfitqua`	The returned `list` from the `optim` function. This `list` contains a repeat of the parameters, the value of the objective function (σ_{RMSE} or σ_{MAD}), the interation count, and convergence status.

Note

The `disfitqua` function is likely more difficult to apply for $n > 3$ (high parameter) distributions because of the inherent complexity of the mathematics of such distributions and their applicable parameter (and thus valid L-moment ranges). The complex interplay between parameters and L-moments can make identification of suitable initial parameters `init.para` or initial L-moments `init.lmr` more difficult than is the case for $n \leq 3$ distributions. The default initial parameters are computed from an assumed condition that the L-moments ratios $\tau_r = 0$ for $r \geq 3$. This is not ideal, however, and the **Examples** show how to move into high parameter distributions using the results from a previous fit.

Author(s)

W.H. Asquith

References

Asquith, W.H., and Roussel, M.C., 2009, Regression equations for estimation of annual peak-streamflow frequency for undeveloped watersheds in Texas using an L-moment-based, PRESS-minimized, residual-adjusted approach: U.S. Geological Survey Scientific Investigations Report 2009–5087, 48 p., doi:10.3133/sir20095087.

See Also

`dist.list`, `lmoms`, `lmom2vec`, `par2lmom`, `par2qua`, `vec2lmom`, `vec2par`

Examples

```
# Suppose the following quantiles are estimated using eight equations provided by
# Asquith and Roussel (2009) for some watershed in Texas:
Q <- c(1480, 3230, 4670, 6750, 8700, 11000, 13600, 17500)
# These are real estimates from a suite of watershed properties; the watershed
# itself and location are not germane to demonstrate this function.
LQ <- log10(Q) # transform to logarithms of cubic feet per second
# Convert the averge annual return periods for the quantiles into probability
P <- T2prob(c(2, 5, 10, 25, 50, 100, 200, 500)); qP <- qnorm(P) # std norm variates
# The log-Pearson type III (LPIII) is immensely popular for flood-risk computations.
```

```
# Let us compute LPIII parameters to the available quantiles and probabilities for
# the watershed. The log-Pearson type III is "pe3" in the lmomco with logarithms.
par1 <- disfitqua(LQ, P, type="pe3", objfun="rmse") # root mean square error
par2 <- disfitqua(LQ, P, type="pe3", objfun="mad" ) # mean absolute deviation
# Now express the fitted distributions in forms of an LPIII.
LQfit1 <- qlmomco(P, par1); LQfit2 <- qlmomco(P, par2)

plot( qP, LQ, pch=5, xlab="STANDARD NORMAL VARIATES",
                     ylab="FLOOD QUANTILES, CUBIC FEET PER SECOND")
lines(qP, LQfit1, col=2); lines(qP, LQfit2, col=4) # red and blue lines

## Not run:
# Now demonstrate how a Wakeby distribution can be fit. This is an example of how a
# three parameter distribution might be fit, and then the general L-moments secured for
# an alternative fit using a far more complicated distribution. The Wakeby for the
# above situation does not fit out of the box.
lmr1 <- theoLmoms(par1) # We need five L-moments but lmompe3() only gives four,
# therefore must compute the L-moment by numerical integration provided by theoLmoms().
par3 <- disfitqua(LQ, P, type="wak", objfun="rmse", init.lmr=lmr1)
lines(qP, par2qua(P, par3), col=6, lty=2) # dashed line, par2qua alternative to qlmomco

# Finally, the initial L-moment equivalents and then the L-moments of the fitted
# distribution can be computed and compared.
par2lmom(vec2par(par3$init.para, type="wak"))$ratios # initial L-moments
par2lmom(vec2par(par3$para,      type="wak"))$ratios # final   L-moments
## End(Not run)
```

---

dist.list *List of Distribution Names*

Description

Return a list of the three character syntax identifying distributions supported within the **lmomco** package. The distributions are aep4, cau, emu, exp, gam, gep, gev, gld, glo, gno, gov, gpa, gum, kap, kmu, kur, lap, lmrq, ln3, nor, pdq3, pdq4, pe3, ray, revgum, rice, sla, smd, st3, texp, tri, wak, and wei. These abbreviations and only these are used in routing logic within **lmomco**. There is no provision for fuzzy matching. The full distributions names are available in prettydist.

Usage

```
dist.list(type=NULL)
```

Arguments

type	If type is not NULL and is one of the abbreviations shown above, then the number of parameters of that distribution are returned or a warning message is issued. This subtle feature might be useful for developers.

Value

A vector of distribution identifiers as listed above or the number of parameters for a given distribution type.

Author(s)

W.H. Asquith

See Also

prettydist

Examples

```
dist.list("gpa")

## Not run:
# Build an L-moment object
LM <- vec2lmom(c(10000, 1500, 0.3, 0.1, 0.04))
lm2 <- lmorph(LM)  # convert to vectored format
lm1 <- lmorph(lm2) # and back to named format
dist <- dist.list()
# Demonstrate that lmom2par internally converts to needed L-moment object
for(i in 1:length(dist)) {
  # Skip Cauchy and Slash (need TL-moments).
  # Skip AEP4, Kumaraswamy, LMRQ, Student t (3-parameter), Truncated Exponential
  # are skipped because each is inapplicable to the given L-moments.
  # The Eta-Mu and Kappa-Mu are skipped for speed.
  if(dist[i] == 'aep4' | dist[i] == 'cau' | dist[i] == 'emu'  | dist[i] == 'gep' |
     dist[i] == 'kmu'  | dist[i] == 'kur' | dist[i] == 'lmrq' | dist[i] == 'tri' |
     dist[i] == 'sla'  | dist[i] == 'st3' | dist[i] == 'texp') next
  message(dist[i], " parameters : ",
          paste(round(lmom2par(lm1, type=dist[i])$para, digits=4), collapse=", "))
  message(dist[i], " parameters : ",
          paste(round(lmom2par(lm2, type=dist[i])$para, digits=4), collapse=", "))
} #
## End(Not run)
```

---

dlmomco *Probability Density Function of the Distributions*

Description

This function acts as an alternative front end to par2pdf. The nomenclature of the dlmomco function is to mimic that of built-in R functions that interface with distributions.

Usage

```
dlmomco(x, para)
```

Arguments

x	A real value vector.
para	The parameters from lmom2par or similar.

Value

Probability density for x.

Author(s)

W.H. Asquith

See Also

plmomco, qlmomco, rlmomco, slmomco

Examples

```
para <- vec2par(c(0,1),type="nor") # standard normal parameters
nonexceed <- dlmomco(1,para) # percentile of one standard deviation
```

DrillBitLifetime *Lifetime of Drill Bits*

Description

Hamada (1995, table 9.3) provides a table of lifetime to breakage measured in cycles for drill bits used for producing small holes in printed circuit boards. The data were collected under various control and noise factors to perform reliability assessment to maximize bit reliability with minimization of hole diameter. Smaller holes permit higher density of placed circuitry, and are thus economically attractive. The testing was completed at 3,000 cycles—the right censoring threhold.

Usage

```
data(DrillBitLifetime)
```

Format

A data frame with

LIFETIME Measured in cycles.

References

Hamada, M., 1995, Analysis of experiments for reliability improvement and robust reliability: in Balakrishnan, N. (ed.) Recent Advances in Life-Testing and Reliability: Boca Raton, Fla., CRC Press, ISBN 0–8493–8972–0, pp. 155–172.

Examples

```
data(DrillBitLifetime)
summary(DrillBitLifetime)
## Not run:
data(DrillBitLifetime)
X     <- DrillBitLifetime$LIFETIME
lmr   <- lmoms(X); par <- lmom2par(lmr,  type="gpa")
pwm   <- pwmRC(X, threshold=3000); zeta <- pwm$zeta
lmrrc <- pwm2lmom(pwm$Bbetas)
rcpar <- pargpaRC(lmrrc, zeta=zeta)
XBAR  <- lmomgpa(rcpar)$lambdas[1]
F <- nonexceeds(); P <- 100*F; x <- seq(min(X), max(X))
plot(sort(X), 100*pp(X), xlab="LIFETIME", ylab="PERCENT", xlim=c(1,10000))
rug(X, col=rgb(0,0,0,0.5))
lines(c(XBAR, XBAR), range(P), lty=2) # mean (expectation of life)
lines(cmlmomco(F, rcpar),  P,  lty=2) # conditional mean
points(XBAR, 0, pch=16)
lines(x, 100*plmomco(x, par),   lwd=2, col=8) # fitted dist.
lines(x, 100*plmomco(x, rcpar), lwd=3, col=1) # fitted dist.

lines( rmlmomco(F, rcpar), P,   col=4) # residual mean life
lines(rrmlmomco(F, rcpar), P,   col=4, lty=2) # rev. residual mean life
lines(x, 1E4*hlmomco(x, rcpar), col=2) # hazard function
lines(x, 1E2*lrzlmomco(plmomco(x, rcpar), rcpar), col=3) # Lorenz func.
legend(4000, 40,
       c("Mean (vertical) or conditional mean (dot at intersect.)",
         "Fitted GPA naively to all data",
         "Fitted GPA to right-censoring PWMs",
         "Residual mean life", "Reversed residual mean life",
         "Hazard function x 1E4", "Lorenz curve x 100"
        ), cex=0.75,
       lwd=c(1, 2, 3, 1, 1, 1, 1), col=c(1, 8, 1, 4, 4, 2, 3),
       lty=c(2, 1, 1, 1, 2, 1, 1), pch=rep(NA, 8))

## End(Not run)
```

`expect.max.ostat` *Compute the Expectation of a Maximum (or Minimum and others) Order Statistic*

Description

The maximum (or minimum) expectation of an order statistic can be directly used for L-moment computation through either of the following two equations (Hosking, 2006) as dictated by using the maximum ($\mathrm{E}[X_{k:k}]$, `expect.max.ostat`) or minimum ($\mathrm{E}[X_{1:k}]$, `expect.min.ostat`):

$$\lambda_r = (-1)^{r-1}\sum_{k=1}^{r}(-1)^{r-k}k^{-1}\binom{r-1}{k-1}\binom{r+k-2}{k-1}\mathrm{E}[X_{1:k}],$$

and

$$\lambda_r = \sum_{k=1}^{r} (-1)^{r-k} k^{-1} \binom{r-1}{k-1} \binom{r+k-2}{k-1} \mathrm{E}[X_{k:k}].$$

In terms of the quantile function qlmomco, the expectation of an order statistic (Asquith, 2011, p. 49) is

$$\mathrm{E}[X_{j:n}] = n \binom{n-1}{j-1} \int_0^1 x(F) \times F^{j-1} \times (1-F)^{n-j}\, \mathrm{d}F,$$

where $x(F)$ is the quantile function, F is nonexceedance probability, n is sample size, and j is the jth order statistic.

In terms of the probability density function (PDF) dlmomco and cumulative density function (CDF) plmomco, the expectation of an order statistic (Asquith, 2011, p. 50) is

$$\mathrm{E}[X_{j:n}] = \frac{1}{\mathrm{B}(j, n-j+1)} \int_{-\infty}^{\infty} [F(x)]^{j-1} [1-F(x)]^{n-j} x\, f(x)\, \mathrm{d}x,$$

where $F(x)$ is the CDF, $f(x)$ is the PDF, and $\mathrm{B}(j, n-j+1)$ is the complete Beta function, which in R is `beta` with the same argument order as shown above.

Usage

```
expect.max.ostat(n, para=NULL, cdf=NULL, pdf=NULL, qua=NULL,
                 j=NULL, lower=-Inf, upper=Inf, aslist=FALSE, ...)
```

Arguments

`n`	The sample size.
`para`	A distribution parameter list from a function such as vec2par or lmom2par.
`cdf`	cumulative distribution function of the distribution.
`pdf`	probability density function of the distribution.
`qua`	quantile function of the distribution. If this is defined, then `cdf` and `pdf` are ignored.
`j`	The jth value of the order statistic, which defaults to *n=j* (the maximum order statistic) if `j=NULL`.
`lower`	The lower limit for integration.
`upper`	The upper limit for integration.
`aslist`	A logically triggering whether an R `list` is returned instead of just the expection.
`...`	Additional arguments to pass to the three distribution functions.

Details

If `qua != NULL`, then the first order-statistic expectation equation above is used, and any function that might have been set in `cdf` and `pdf` is *ignored*. If the limits are infinite (default), then the limits of the integration will be set to $F \downarrow = 0$ and $F \uparrow = 1$. The user can replace these by setting the limits to something "near" zero and(or) "near" 1. Please consult the **Note** below concerning more information about the limits of integration.

If qua == NULL, then the second order-statistic expectation equation above is used and cdf and pdf must be set. The default $\pm\infty$ limits are used unless the user *knows* otherwise for the distribution or through supervision provides their meaning of *small* and *large*.

This function requires the user to provide either the qua or the cdf and pdf functions, which is somewhat divergent from the typical flow of logic of **lmomco**. This has been done so that expect.max.ostat can be used readily for experimental distribution functions. It is suggested that the parameter object be left in the **lmomco** style (see vec2par) even if the user is providing their own distribution functions.

Last comments: This function is built around the idea that either (1) the cdf and pdf ensemble or (2) qua exist in some clean analytical form and therefore the qua=NULL is the trigger on which order statistic expectation integral is used. This precludes an attempt to compute the support of the distribution internally, and thus providing possibly superior (more refined) lower and upper limits. Here is a suggested re-implementation using the support of the Generalized Extreme Value distribution:

```
para <- vec2par(c(100, 23, -0.5), type="gev")
lo <- quagev(0, para) # The value 54
hi <- quagev(1, para) # Infinity
E22 <- expect.max.ostat(2, para=para,cdf=cdfgev, pdf=pdfgev,
                           lower=lo, upper=hi)
E21 <- expect.min.ostat(2, para=para,cdf=cdfgev, pdf=pdfgev,
                           lower=lo, upper=hi)
L2 <- (E22 - E21)/2 # definition of L-scale
cat("L-scale: ", L2, "(integration)",
    lmomgev(para)$lambdas[2], "(theory)\n")
# The results show 33.77202 as L-scale.
```

The design intent makes it possible for some arbitrary and(or) new quantile function with difficult cdf and pdf expressions (or numerical approximations) to not be needed as the L-moments are explored. Contrarily, perhaps some new pdf exists and simple integration of it is made to get the cdf but the qua would need more elaborate numerics to invert the cdf. The user could then still explore the L-moments with supervision on the integration limits or foreknowledge of the support of the distribution.

Value

The expectation of the maximum order statistic, unless j is specified and then the expectation of that order statistic is returned. This similarly holds if the expect.min.ostat function is used except "maximum" becomes the "minimum".

Alternatively, an R list is returned.

type	The type of approach used: "bypdfcdf" means the PDF and CDF of the distribution were used, and alternatively "byqua" means that the quantile function was used.
value	See previous discussion of value.

`abs.error`	Estimate of the modulus of the absolute error from R function `integrate`.
`subdivisions`	The number of subintervals produced in the subdivision process from R function `integrate`.
`message`	"OK" or a character string giving the error message.

Note

A function such as this might be helpful for computations involving distribution mixtures. Mixtures are readily made using the algebra of quantile functions (Gilchrist, 2000; Asquith, 2011, sec. 2.1.5 "The Algebra of Quantile Functions").

Last comments: Internally, judicious use of logarithms and exponents for the terms involving the F and $1 - F$ and the quantities to the left of the intergrals shown above are made in an attempt to maximize stability of the function without the user having to become too invested in the `lower` and `upper` limits. For example, $(1 - F)^{n-j} \rightarrow \exp([n - j]\log(1 - F))$. Testing indicates that this coding practice is quite useful. But there will undoubtedly be times for which the user needs to be informed enough about the expected value on return to identify that tweaking to the integration limits is needed. Also use of R functions `lbeta` and `lchoose` is made to maximize operations in logarithmic space.

For **lmomco** v.2.1.+: Because of the extensive use of exponents and logarithms as described, enhanced deep tail estimation of the extrema for large n and large or small j results. This has come at the expense that expectations can be computed when the expectations actually do not exist. An error in the integration no longer occurs in **lmomco**. For example, the Cauchy distribution has infinite extrema but this function (for least for a selected parameter set and `n=10`) provides apparent values for the $\mathrm{E}[X_{1:n}]$ and $\mathrm{E}[X_{n:n}]$ when the `cdf` and `pdf` are used but not when the `qua` is used. Users are cautioned to not rely on `expect.max.ostat` "knowing" that a given distribution has undefined order statistic extrema. Now for the Cauchy case just described, the extrema for $j = [1, n]$ are hugely(!) greater in magnitude than for $j = [2, (n - 1)]$, so some resemblance of *infinity* remains.

The alias `eostat` is a shorter name dispatching to `expect.max.ostat` all of the arguments.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Hosking, J.R.M., 2006, On the characterization of distributions by their L-moments: Journal of Statistical Planning and Inference, v. 136, no. 1, pp. 193–198.

See Also

`theoLmoms.max.ostat`, `expect.min.ostat`, `eostat`

Examples

```
para <- vec2par(c(10, 100), type="nor")
n <- 12
# The three outputted values from should be similar:
# (1) theoretical, (2) theoretical, and (3) simulation
expect.max.ostat(n, para=para, cdf=cdfnor, pdf=pdfnor)
expect.max.ostat(n, para=para, qua=quanor)
mean(sapply(seq_len(1000), function(x) { max(rlmomco(n, para))}))

eostat(8, j=5, qua=quagum, para=vec2par(c(1670, 1000), type="gum"))

## Not run:
para <- vec2par(c(1280, 800), type="nor")
expect.max.ostat(10, j=9, para, qua=quanor)
[1] 2081.086      # SUCCESS --------------------------
expect.max.ostat(10, j=9, para, pdf=pdfnor, cdf=cdfnor,
                                lower=-1E3, upper=1E6)
[1] 1.662701e-06  # FAILURE --------------------------
expect.max.ostat(10, j=9, para, pdf=pdfnor, cdf=cdfnor,
                                lower=-1E3, upper=1E5)
[1] 2081.086      # SUCCESS --------------------------
## End(Not run)
```

---

f2f   *Subsetting of Nonexceedance Probabilities Related to Conditional Probability Adjustment*

Description

This function subsetting nonexceedance probability according to

$$F(x) < -F(x|F(x)[>, \geq]p),$$

where F is nonexceedance probability for x and pp is the probability of a threshold. In R logic, this is simply `f <- f[f > pp]` for `type == "gt"` or `f <- f[f >= pp]` for `type == "ge"`.

This function is particularly useful to shorten a commonly needed code logic related such as `FF[FF >= XloALL$pp]`, which would be needed in conditional probability adjustements and `XloALL` is from `x2xlo`. This could be replaced by syntax such as `f2f(FF, xlo=XloALL)`. This function is very similar to `f2flo` with the only exception that the conditional probability adjustment is not made.

Usage

```
f2f(f, pp=NA, xlo=NULL, type=c("ge", "gt"))
```

Arguments

f	A vector of nonexceedance probabilities.
pp	The plotting position of the left-hand threshold and recommended to come from `x2xlo`.

xlo An optional result from x2xlo from which the pp will be take instead of from the argument pp.

type The type of the logical construction gt means greater than the pp and ge means greater than or equal to the pp for the computations. There can be subtle variations in conceptualization of the truncation need or purpose and hence this argument is provided for flexibility.

Value

A vector of conditional nonexceedance probabilities.

Author(s)

W.H. Asquith

See Also

x2xlo, xlo2qua, f2flo, f2f

Examples

```
# See examples for x2xlo().
```

---

f2flo *Conversion of Annual Nonexceedance Probability to Conditional Probability Nonexceedance Probabilities*

Description

This function converts the cumulative distribution function of $F(x)$ to a conditional cumulative distribution function $P(x)$ based on the probability level of the left-hand threshold. It is recommended that this threshold (as expressed as a probability) be that value returned from x2xlo in element pp. The conversion is simple

$$P(x) < -(F(x) - pp)/(1 - pp),$$

where the term pp corresponds to the estimated probability or plotting position of the left-hand threshold.

This function is particularly useful for applications in which zero values in the data set require truncation so that logarithms of the data may be used. But also this function contributes to the isolation of the right-hand tail of the distribution for analysis. Finally, `f <- f[f >= pp]` for `type="ge"` or `f <- f[f > pp]` for `type="gt"` is used internally for probability subsetting, so the user does not have to do that with the nonexceedance probability before calling this function. The function f2f does similar subsetting without converting $F(x)$ to $P(x)$. Users are directed to **Examples** under par2qua2lo and carefully note how `f2flo` and f2f are used.

Usage

```
f2flo(f, pp=NA, xlo=NULL, type=c("ge", "gt"))
```

Arguments

- `f` A vector of nonexceedance probabilities.
- `pp` The plotting position of the left-hand threshold and recommended to come from `x2xlo`.
- `xlo` An optional result from `x2xlo` from which the `pp` will be take instead of from the argument `pp`.
- `type` The type of the logical construction `gt` means greater than the `pp` and `ge` means greater than or equal to the `pp` for the computations. There can be subtle variations in conceptualization of the truncation need or purpose and hence this argument is provided for flexibility.

Value

A vector of conditional nonexceedance probabilities.

Author(s)

W.H. Asquith

See Also

`x2xlo`, `flo2f`, `f2f`, `xlo2qua`

Examples

```
# See examples for x2xlo().
```

---

f2fpds — *Conversion of Annual Nonexceedance Probability to Partial Duration Nonexceedance Probability*

Description

This function takes an annual exceedance probability and converts it to a "partial-duration series" (a term in Hydrology) nonexceedance probability through a simple assumption that the Poisson distribution is appropriate for arrive modeling. The relation between the cumulative distribution function $G(x)$ for the partial-duration series is related to the cumulative distribution function $F(x)$ of the annual series (data on an annual basis and quite common in Hydrology) by

$$G(x) = [\log(F(x)) + \eta]/\eta.$$

The core assumption is that successive events in the partial-duration series can be considered as *independent*. The η term is the arrival rate of the events. For example, suppose that 21 events have occurred in 15 years, then $\eta = 21/15 = 1.4$ events per year.

A comprehensive demonstration is shown in the example for `fpds2f`. That function performs the opposite conversion. Lastly, the cross reference to `x2xlo` is made because the example contained therein provides another demonstration of partial-duration and annual series frequency analysis.

Usage

```
f2fpds(f, rate=NA)
```

Arguments

`f`	A vector of annual nonexceedance probabilities.
`rate`	The number of events per year.

Value

A vector of converted nonexceedance probabilities.

Author(s)

W.H. Asquith

References

Stedinger, J.R., Vogel, R.M., Foufoula-Georgiou, E., 1993, Frequency analysis of extreme events: *in* Handbook of Hydrology, ed. Maidment, D.R., McGraw-Hill, Section 18.6 Partial duration series, mixtures, and censored data, pp. 18.37–18.39.

See Also

`fpds2f`, `x2xlo`, `f2flo`, `flo2f`

Examples

```
# See examples for fpds2f().
```

`fliplmoms` *Flip L-moments by Flip Attribute in L-moment Vector*

Description

This function flips the L-moments by a flip attribute within an L-moment object such as that returned by `lmomsRCmark`. The function will attempt to identify the L-moment object and `lmorph` as necessary, but this support is not guaranteed. The flipping process is used to support left-tail censoring using the right-tail censoring alogrithms of **lmomco**. The odd order (`seq(3,n,by2)`) λ_r and τ_r are negated. The mean $\hat{\lambda}_1$ is computed by subtracting the λ_1 from the *lmom* argument from the flip M: $\hat{\lambda}_1 = M - \lambda_1$ and the τ_2 is subsequently adjusted by the new mean. This function is written to provide a convenient method to re-transform or back flip the L-moments computed by `lmomsRCmark`. Detailed review of the example problem listed here is recommended.

Usage

```
fliplmoms(lmom, flip=NULL, checklmom=TRUE)
```

Arguments

lmom
: A L-moment object created by `lmomsRCmark` or other vectorize L-moment list.

flip
: `lmomsRCmark` provides the flip, but for other vectorized L-moment list support, the flip can be set by this argument.

checklmom
: Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.

Value

An R `list` is returned that matches the structure of the *lmom* argument (unless an `lmorph` was attempted). The structure is intended to match that coming from `lmomsRCmark`.

Author(s)

W.H. Asquith

References

Wang, Dongliang, Hutson, A.D., and Miecznikowski, J.C., 2010, L-moment estimation for parametric survival models given censored data: Statistical Methodology, v. 7, no. 6, pp. 655–667.

Helsel, D.R., 2005, Nondetects and data analysis—Statistics for censored environmental data: Hoboken, New Jersey, John Wiley, 250 p.

See Also

`lmomsRCmark`

Examples

```
# Create some data with **multiple detection limits**
# This is a left-tail censoring problem, and flipping will be required.
fakedat1 <- rnorm(50, mean=16, sd=5)
fake1.left.censor.indicator <- fakedat1 <  14
fakedat1[fake1.left.censor.indicator]   <- 14

fakedat2 <- rnorm(50, mean=16, sd=5)
fake2.left.censor.indicator <- fakedat2 <  10
fakedat2[fake2.left.censor.indicator]   <- 10

# combine the data sets
fakedat <- c(fakedat1, fakedat2);
fake.left.censor.indicator <- c(fake1.left.censor.indicator,
                                fake2.left.censor.indicator)
ix <- order(fakedat)
fakedat <- fakedat[ix]
fake.left.censor.indicator <- fake.left.censor.indicator[ix]
```

```
lmr.usual       <- lmoms(fakedat)
lmr.flipped     <- lmomsRCmark(fakedat, flip=TRUE,
                               rcmark=fake.left.censor.indicator)
lmr.backflipped <- fliplmoms(lmr.flipped); # re-transform
pch <- as.numeric(fake.left.censor.indicator)*15 + 1
F <- nonexceeds()
plot(pp(fakedat), sort(fakedat), pch=pch,
     xlab="NONEXCEEDANCE PROBABILITY", ylab="DATA VALUE")
lines(F, qlmomco(F, parnor(lmr.backflipped)), lwd=2)
lines(F, qlmomco(F, parnor(lmr.usual)), lty=2)
legend(0,20, c("Uncensored", "Left-tail censored"), pch=c(1,16))
# The solid line represented the Normal distribution fit by
# censoring indicator on the multiple left-tail detection limits.
## Not run:
# see example in pwmRC
H <- c(3,4,5,6,6,7,8,8,9,9,9,10,10,11,11,11,13,13,13,13,13,
       17,19,19,25,29,33,42,42,51.9999,52,52,52)
# 51.9999 was really 52, a real (noncensored) data point.
flip <- 100
F <- flip - H #
RCpwm <- pwmRC(H, threshold=52)
lmorph(pwm2lmom(vec2pwm(RCpwm$Bbetas))) # OUTPUT1 STARTS HERE

LCpwm <- pwmLC(F, threshold=(flip - 52))
LClmr <- pwm2lmom(vec2pwm(LCpwm$Bbetas))
LClmr <- lmorph(LClmr)
#LClmr$flip <- 100; fliplmoms(LClmr) # would also work
fliplmoms(LClmr, flip=flip) # OUTPUT2 STARTS HERE

# The two outputs are the same showing how the flip argument works
## End(Not run)
```

`flo2f` *Conversion of Conditional Nonexceedance Probability to Annual Nonexceedance Probability*

Description

This function converts the conditional cumulative distribution function of $P(x)$ to a cumulative distribution function $F(x)$ based on the probability level of the left-hand threshold. It is recommended that this threshold (as expressed as a probability) be that value returned from x2xlo in attribute pp. The conversion is simple

$$F(x) = pp + (1 - pp)P(x),$$

where the term pp corresponds to the estimated probability or plotting position of the left-hand threshold.

This function is particularly useful for applications in which zero values in the data set require truncation so that logarithms of the data may be used. But also this function contributes to the isolation of the right-hand tail of the distribution for analysis by conditionally trimming out the left-hand tail at the analyst's discretion.

Usage

```
flo2f(f, pp=NA, xlo=NULL)
```

Arguments

`f`	A vector of nonexceedance probabilities.
`pp`	The plotting position of the left-hand threshold and recommended to come from `x2xlo`.
`xlo`	An optional result from `x2xlo` from which the `pp` will be take instead of from the argument `pp`.

Value

A vector of converted nonexceedance probabilities.

Author(s)

W.H. Asquith

See Also

`x2xlo`, `f2flo`

Examples

```
flo2f(f2flo(.73,pp=.1),pp=.1)
# Also see examples for x2xlo().
```

---

fpds2f *Conversion of Partial-Duration Nonexceedance Probability to Annual Nonexceedance Probability*

Description

This function takes partial duration series nonexceedance probability and converts it to a an annual exceedance probability through a simple assumption that the Poisson distribution is appropriate. The relation between the cumulative distribution function $F(x)$ for the annual series is related to the cumulative distribution function $G(x)$ of the partial-duration series by

$$F(x) = \exp(-\eta[1 - G(x)]).$$

The core assumption is that successive events in the partial-duration series can be considered as *independent*. The η term is the arrival rate of the events. For example, suppose that 21 events have occurred in 15 years, then $\eta = 21/15 = 1.4$ events per year.

The example documented here provides a comprehensive demonstration of the function along with a partnering function `f2fpds`. That function performs the opposite conversion. Lastly, the cross reference to `x2xlo` is made because the example contained therein provides another demonstration of partial-duration and annual series frequency analysis.

Usage

```
fpds2f(fpds, rate=NA)
```

Arguments

fpds	A vector of partial-duration nonexceedance probabilities.
rate	The number of events per year.

Value

A vector of converted nonexceedance probabilities.

Author(s)

W.H. Asquith

References

Stedinger, J.R., Vogel, R.M., Foufoula-Georgiou, E., 1993, Frequency analysis of extreme events: *in* Handbook of Hydrology, ed. Maidment, D.R., McGraw-Hill, Section 18.6 Partial duration series, mixtures, and censored data, pp. 18.37–18.39.

See Also

f2fpds, x2xlo, f2flo, flo2f

Examples

```
## Not run:
stream <- "A Stream in West Texas"
Qpds    <- c(61.8, 122, 47.3, 71.1, 211, 139, 244, 111, 233, 102)
Qann <- c(61.8, 122, 71.1, 211, 244, 0, 233)
years  <- length(Qann)  # gage has operated for about 7 years
visits <- 27  # number of visits or "events"
rate   <- visits/years
Z <- rep(0, visits-length(Qpds))
Qpds <- c(Qpds,Z) # The creation of a partial duration series
# that will contain numerous zero values.

Fs <- seq(0.001,.999, by=.005) # used to generate curves

type <- "pe3" # The Pearson type III distribution
PPpds <- pp(Qpds); Qpds <- sort(Qpds) # plotting positions (partials)
PPann <- pp(Qann); Qann <- sort(Qann) # plotting positions (annuals)
parann <- lmom2par(lmoms(Qann), type=type) # parameter estimation (annuals)
parpsd <- lmom2par(lmoms(Qpds), type=type) # parameter estimation (partials)

Fsplot    <- qnorm(Fs) # in order to produce normal probability paper
PPpdsplot <- qnorm(fpds2f(PPpds, rate=rate)) # ditto
PPannplot <- qnorm(PPann) # ditto
```

```
# There are many zero values in this particular data set that require leaving
# them out in order to achieve appropriate curvature of the Pearson type III
# distribution. Conditional probability adjustments will be used.
Qlo <- x2xlo(Qpds) # Create a left out object with an implied threshold of zero
parlo <- lmom2par(lmoms(Qlo$xin), type=type) # parameter estimation for the
# partial duration series values that are greater than the threshold, which
# defaults to zero.

plot(PPpdsplot, Qpds, type="n", ylim=c(0,400), xlim=qnorm(c(.01,.99)),
     xlab="STANDARD NORMAL VARIATE", ylab="DISCHARGE, IN CUBIC FEET PER SECOND")
mtext(stream)
points(PPannplot, Qann, col=3, cex=2, lwd=2, pch=0)
points(qnorm(fpds2f(PPpds, rate=rate)), Qpds, pch=16, cex=0.5 )
points(qnorm(fpds2f(flo2f(pp(Qlo$xin), pp=Qlo$pp), rate=rate)),
       sort(Qlo$xin), col=2, lwd=2, cex=1.5, pch=1)
points(qnorm(fpds2f(Qlo$ppout, rate=rate)),
       Qlo$xout, pch=4, col=4)

lines(qnorm(fpds2f(Fs, rate=rate)),
      qlmomco(Fs, parpsd), lwd=1, lty=2)
lines(Fsplot, qlmomco(Fs, parann), col=3, lwd=2)
lines(qnorm(fpds2f(flo2f(Fs, pp=Qlo$pp), rate=rate)),
      qlmomco(Fs, parlo), col=2, lwd=3)

# The following represents a subtle application of the probability transform
# functions. The show how one starts with annual recurrence intervals
# converts into conventional annual nonexceedance probabilities as well as
# converting these to the partial duration nonexceedance probabilities.
Tann <- c(2, 5, 10, 25, 50, 100)
Fann <- T2prob(Tann); Gpds <- f2fpds(Fann, rate=rate)
FFpds <- qlmomco(f2flo(Gpds, pp=Qlo$pp), parlo)
FFann <- qlmomco(Fann, parann)
points(qnorm(Fann), FFpds, col=2, pch=16)
points(qnorm(Fann), FFann, col=3, pch=16)

legend(-2.4,400, c("True annual series (with one zero year)",
                "Partial duration series (including 'visits' as 'events')",
                "Partial duration series (after conditional adjustment)",
                "Left-out values (<= zero) (trigger of conditional probability)",
                "PE3 partial-duration frequency curve (PE3-PDS)",
                "PE3 annual-series frequency curve (PE3-ANN)",
                "PE3 partial-duration frequency curve (zeros removed) (PE3-PDSz)",
                "PE3-ANN  T-year event: 2, 5, 10, 25, 50, 100 years",
                "PE3-PDSz T-year event: 2, 5, 10, 25, 50, 100 years"),
       bty="n", cex=.75,
       pch=c(0,  16, 1, 4, NA, NA, NA, 16, 16),
       col=c(3,  1, 2,  4,  1,  3,  2,  3, 2),
       pt.lwd=c(2,1,2,1), pt.cex=c(2, 0.5, 1.5, 1, NA, NA, NA, 1, 1),
       lwd=c(0,0,0,0,1,2,3), lty=c(0,0,0,0,2,1,1))

## End(Not run)
```

freq.curve.all *Compute Frequency Curve for Almost All Distributions*

Description

This function is dispatcher on top of a select suite of quaCCC functions that compute frequency curves for the L-moments. The term "frequency curves" is common in hydrology and is a renaming of the more widely known by statisticians term the "quantile function." The notation CCC represents the character notation for the distribution: exp, gam, gev, gld, glo, gno, gpa, gum, kap, nor, pe3, wak, and wei. The nonexceedance probabilities to construct the curves are derived from nonexceeds.

Usage

```
freq.curve.all(lmom, aslog10=FALSE, asprob=TRUE,
                     no2para=FALSE, no3para=FALSE,
                     no4para=FALSE, no5para=FALSE,
                     step=FALSE, show=FALSE,
                     xmin=NULL, xmax=NULL, xlim=NULL,
                     ymin=NULL, ymax=NULL, ylim=NULL,
                     aep4=FALSE, exp=TRUE, gam=TRUE, gev=TRUE, gld=FALSE,
                     glo=TRUE, gno=TRUE, gpa=TRUE, gum=TRUE, kap=TRUE,
                     nor=TRUE, pe3=TRUE, wak=TRUE, wei=TRUE,...)
```

Arguments

lmom	A L-moment object from lmoms, lmom.ub, or vec2lmom.
aslog10	Compute log10 of quantiles—note that NaNs produced in: log(x, base) will be produced for less than zero values.
asprob	The R qnorm function is used to convert nonexceedance probabilities, which are produced by nonexceeds, to standard normal variates. The Normal distribution will plot as straight line when this argument is TRUE
no2para	If TRUE, do not run the 2-parameter distributions: exp, gam, gum, and nor.
no3para	If TRUE, do not run the 3-parameter distributions: gev, glo, gno, gpa, pe3, and wei.
no4para	If TRUE, do not run the 4-parameter distributions: kap, gld, aep4.
no5para	If TRUE, do not run the 5-parameter distributions: wak.
step	Shows incremental processing of each distribution.
show	Plots all the frequency curves in a simple (crowded) plot.
xmin	Minimum x-axis value to use instead of the automatic value determined from the nonexceedance probabilities. This argument is only used is show=TRUE.

`xmax`	Maximum x-axis value to use instead of the automatic value determined from the nonexceedance probabilities. This argument is only used is `show=TRUE`.
`xlim`	Both limits of the x-axis. This argument is only used is `show=TRUE`.
`ymin`	Minimum y-axis value to use instead of the automatic value determined from the nonexceedance probabilities. This argument is only used is `show=TRUE`.
`ymax`	Maximum y-axis value to use instead of the automatic value determined from the nonexceedance probabilities. This argument is only used is `show=TRUE`.
`ylim`	Both limits of the y-axis. This argument is only used is `show=TRUE`.
`aep4`	A logical switch on computation of corresponding distribution—default is `FALSE`.
`exp`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`gam`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`gev`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`gld`	A logical switch on computation of corresponding distribution—default is `FALSE`.
`glo`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`gno`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`gpa`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`gum`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`kap`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`nor`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`pe3`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`wak`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`wei`	A logical switch on computation of corresponding distribution—default is `TRUE`.
`...`	Additional parameters are passed to the parameter estimation routines such as `parexp`.

Value

An extensive R `data.frame` of frequency curves. The nonexceedance probability values, which are provided by `nonexceeds`, are the first item in the `data.frame` under the heading of `nonexceeds`. If a particular distribution could not be fit to the L-moments of the data; this particular function returns zeros.

Note

The distributions selected for this function represent a substantial fraction of, but not all, distributions supported by **lmomco**. The `all` and "all" in the function name and the title of this documentation is a little misleading. The selection process was made near the beginning of **lmomco** availability and distributions available in the earliest versions. Further the selected distributions are frequently encountered in hydrology and because these are also those considered in length by Hosking and Wallis (1997) and the **lmom** package.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

quaaep4, quaexp, quagam, quagev, quagld, quaglo, quagno, quagpa, quagum, quakap, quanor, quape3, quawak, and quawei.

Examples

```
L <- vec2lmom(c(35612,23593,0.48,0.21,0.11))
Qtable1 <- freq.curve.all(L, step=TRUE, no2para=TRUE, no4para=TRUE)
## Not run:
Qtable2 <- freq.curve.all(L, gld=TRUE, show=TRUE)
## End(Not run)
```

---

gen.freq.curves — *Plot Randomly Generated Frequency Curves from a Parent Distribution*

Description

This function generates random samples of specified size from a specified parent distribution. Subsequently, the type of parent distribution is fit to the L-moments of the generated sample. The fitted distribution is then plotted. It is the user's responsibility to have an active `plot` already drawn; unless the `callplot` option is `TRUE`. This function is useful to demonstration of sample size on the uncertainty of a fitted distribution—a motivation for this function is as a classroom exercise.

Usage

```
gen.freq.curves(n, para, F=NULL, nsim=10, callplot=TRUE, aslog=FALSE,
                asprob=FALSE, showsample=FALSE, showparent=FALSE,
                lowerCI=NA, upperCI=NA, FCI=NA, ...)
```

Arguments

`n`	Sample size to draw from parent as specified by `para`.
`para`	The parameters from `lmom2par` or `vec2par`.
`F`	The nonexceedance probabilities for horizontal axis—defaults to `nonexceeds` when the argument is `NULL`.
`nsim`	The number of simulations to perform (frequency curves to draw)—the default is 10.
`callplot`	Calls `plot` to acquire a graphics device—default is `TRUE`, but the called `plot` is left empty.
`aslog`	Compute `log10` of quantiles—note that

```
NaNs produced in: log(x, base)
```

will be produced for less than zero values. Otherwise this is a harmless message.

asprob
: The qnorm function is used to convert nonexceedance probabilities, which are produced by nonexceeds, to Standard Normal variates. The Normal distribution will be a straight line when this argument is TRUE and aslog=FALSE.

showsample
: Each simulated sample is drawn through plotting positions (pp).

showparent
: The curve for the parent distribution is plotted on exit from the function if TRUE. Further plotting options can not be controlled—unlike the situation with the drawing of the simulated frequency curves.

lowerCI
: An optional estimate of the lower confidence limit for the FCI nonexceedance probability.

upperCI
: An optional estimate of the upper confidence limit for the FCI nonexceedance probability.

FCI
: The nonexeedance probability of interest for the confidence limits provided in lowerCI and upperCI.

...
: Additional parameters are passed to the lines call within the function—except for the drawing of the parent distribution (see argument showparent).

Value

This function is largely used for its graphical side effects, but if estimates of the lower and upper confidence limits are known (say from genci.simple) then this function can be used to evaluate the counts of simulations at nonexceedance probability FCI outside the limits provided in lowerCI and upperCI.

Author(s)

W.H. Asquith

See Also

lmom2par, nonexceeds, rlmomco, lmoms

Examples

```
## Not run:
# 1-day rainfall Travis county, Texas
para <- vec2par(c(3.00, 1.20, -.0954), type="gev")
F <- .99 # the 100-year event
n <- 46 # sample size derived from 75th percentile of record length distribution
# for Edwards Plateau from Figure 3 of USGS WRIR98-4044 (Asquith, 1998)
# Argument for 75th percentile is that the contours of distribution parameters
# in that report represent a regionalization of the parameters and hence
# record lengths such as the median or smaller for the region seem too small
# for reasonable exploration of confidence limits of precipitation.
nsim <- 5000 # simulation size
seed <- runif(1, min=1, max=10000)
set.seed(seed)
```

```
CI <- genci.simple(para, n, F=F, nsim=nsim, edist="nor")
lo.nor <- CI$lower; hi.nor <- CI$upper

set.seed(seed)
CI <- genci.simple(para, n, F=F, nsim=nsim, edist="aep4")
lo.aep4 <- CI$lower; hi.aep4 <- CI$upper
message("NORMAL ERROR DIST: lowerCI = ",lo.nor, " and upperCI = ",hi.nor)
message("  AEP4 ERROR DIST: lowerCI = ",lo.aep4," and upperCI = ",hi.aep4)
qF <- qnorm(F)
# simulated are grey, parent is black
set.seed(seed)
counts.nor  <- gen.freq.curves(n, para, nsim=nsim,
                   asprob=TRUE, showparent=TRUE, col=rgb(0,0,1,0.025),
                   lowerCI=lo.nor, upperCI=hi.nor, FCI=F)
set.seed(seed)
counts.aep4 <- gen.freq.curves(n, para, nsim=nsim,
                   asprob=TRUE, showparent=TRUE, col=rgb(0,0,1,0.025),
                   lowerCI=lo.aep4, upperCI=hi.aep4, FCI=F)
lines( c(qF,qF), c(lo.nor, hi.nor),  lwd=2, col=2)
points(c(qF,qF), c(lo.nor, hi.nor),  pch=1, lwd=2, col=2)
lines( c(qF,qF), c(lo.aep4,hi.aep4), lwd=2, col=2)
points(c(qF,qF), c(lo.aep4,hi.aep4), pch=2, lwd=2, col=2)
percent.nor  <- (counts.nor$count.above.upperCI +
                 counts.nor$count.below.lowerCI) /
                 counts.nor$count.valid.simulations
percent.aep4 <- (counts.aep4$count.above.upperCI +
                 counts.aep4$count.below.lowerCI) /
                 counts.aep4$count.valid.simulations
percent.nor  <- 100 * percent.nor
percent.aep4 <- 100 * percent.aep4
message("NORMAL ERROR DIST: ",percent.nor)
message("  AEP4 ERROR DIST: ",percent.aep4)
# Continuing on, we are strictly focused on F being equal to 0.99
# Also we are no restricted to the example using the GEV distribution
# The vargev() function is from Handbook of Hydrology
"vargev" <-
function(para, n, F=c("F080", "F090", "F095", "F099", "F998", "F999")) {
   F <- as.character(F)
   if(! are.pargev.valid(para)) return()
   F <- match.arg(F)
   A <- para$para[2]
   K <- para$para[3]
   AS <- list(F080=c(-1.813,  3.017, -1.4010, 0.854),
              F090=c(-2.667,  4.491, -2.2070, 1.802),
              F095=c(-3.222,  5.732, -2.3670, 2.512),
              F098=c(-3.756,  7.185, -2.3140, 4.075),
              F099=c(-4.147,  8.216, -0.2033, 4.780),
              F998=c(-5.336, 10.711, -1.1930, 5.300),
              F999=c(-5.943, 11.815, -0.6300, 6.262))
   AS <- as.environment(AS); CO <- get(F, AS)
   varx <- A^2 * exp( CO[1] + CO[2]*exp(-K) + CO[3]*K^2 + CO[4]*K^3 ) / n
   names(varx) <- NULL
   return(varx)
```

```
}
sdx <- sqrt(vargev(para, n, F="F099"))
VAL  <- qlmomco(F, para)
lo.vargev <- VAL + qt(0.05, df=n) * sdx # minus covered by return of qt()
hi.vargev <- VAL + qt(0.95, df=n) * sdx

set.seed(seed)
counts.vargev <- gen.freq.curves(n, para, nsim=nsim,
                   xlim=c(0,3), ylim=c(3,15),
                   asprob=TRUE, showparent=TRUE, col=rgb(0,0,1,0.01),
                   lowerCI=lo.vargev, upperCI=hi.vargev, FCI=F)
percent.vargev  <- (counts.vargev$count.above.upperCI +
                    counts.vargev$count.below.lowerCI) /
                    counts.vargev$count.valid.simulations
percent.vargev  <- 100 * percent.vargev
lines(c(qF,qF),  range(c(lo.nor,   hi.nor,
                         lo.aep4,  hi.aep4,
                         lo.vargev,hi.vargev)), col=2)
points(c(qF,qF), c(lo.nor,      hi.nor), pch=1, lwd=2, col=2)
points(c(qF,qF), c(lo.aep4,    hi.aep4), pch=3, lwd=2, col=2)
points(c(qF,qF), c(lo.vargev,hi.vargev), pch=2, lwd=2, col=2)
message("NORMAL ERROR DIST: ",percent.nor)
message("  AEP4 ERROR DIST: ",percent.aep4)
message("VARGEV ERROR DIST: ",percent.vargev)

## End(Not run)
```

`genci.simple` *Generate (Estimate) Confidence Intervals for Quantiles of a Parent Distribution*

Description

This function estimates the lower and upper limits of a specified confidence interval for a vector of nonexceedance probabilities F of a specified parent distribution [quantile function $Q(F, \theta)$ with parameters θ] using Monte Carlo simulation. The F are specified by the user. The user also provides Θ of the parent distribution (see `lmom2par`). This function is a wrapper on `qua2ci.simple`; please consult the documentation for that function for further details of the simulations.

Usage

```
genci.simple(para, n, f=NULL, level=0.90, edist="gno", nsim=1000,
               expand=FALSE, verbose=FALSE, showpar=FALSE, quiet=FALSE)
```

Arguments

`para`	The parameters from `lmom2par` or similar.
`n`	The sample size for each Monte Carlo simulation will use.

f
: Vector of nonexceedance probabilities ($0 \leq f \leq 1$) of the quantiles for which the confidence interval are needed. If NULL, then the vector as returned by nonexceeds is used.

level
: The confidence interval ($0 \leq \texttt{level} < 1$). The interval is specified as the size of the interval. The default is 0.90 or the 90th percentile. The function will return the 5th ($(1-0.90)/2$) and 95th ($1-(1-0.90)/2$) percentile cumulative probability of the error distribution for the parent quantile as specified by the nonexceedance probability argument (f). This argument is passed unused to qua2ci.simple.

edist
: The model for the error distribution. Although the Normal (the default) commonly is assumed in error analyses, it need not be, as support for other distributions supported by **lmomco** is available. The default is the Generalized Normal so the not only is the Normal possible but asymmetry is also accomodated (lmomgno). For example, if the L-skew (τ_4) or L-kurtosis (τ_4) values depart considerably from those of the Normal ($\tau_3 = 0$ and $\tau_4 = 0.122602$), then the Generalized Normal or some alternative distribution would likely provide more reliable confidence interval estimation. This argument is passed unused to qua2ci.simple.

nsim
: The number of simulations (replications) for the sample size n to perform. Much larger simulation numbers are recommended—see discussion about qua2ci.simple. This argument is passed unused to qua2ci.simple. Users are encouraged to experiment with qua2ci.simple to get a feel for the value of edist and nsim.

expand
: Should the returned values be expanded to include information relating to the distribution type and L-moments of the distribution at the corresponding nonexceedance probabilities—in other words the information necessary to reconstruct the reported confidence interval. The default is FALSE. If expand=FALSE then a single data.frame of the lower and upper limits along with the true quantile value of the parent is returned. If expand=TRUE, then a more complicated list containing multiple data.frames is returned.

verbose
: The verbosity of the operation of the function. This argument is passed unused to qua2ci.simple.

showpar
: The parameters of the edist for each simulation for each F value passed to qua2ci.simple are printed. This argument is passed unused to qua2ci.simple.

quiet
: Suppress incremental counter for a count down of the F values.

Value

An R data.frame or list is returned (see discussion of argument expand). The following elements could be available.

nonexceed
: A vector of F values, which is returned for convenience so that post operations such as plotting are easily coded.

lwr
: The lower value of the confidence interval having nonexceedance probability equal to (1-level)/2.

true
: The true quantile value from $Q(F, \theta)$ for the corresponding F value.

upr	The upper value of the confidence interval having F equal to `1-(1-level)/2`.
lscale	The second L-moment (L-scale, λ_2) of the distribution of quantiles for the corresponding F. This value is included in the primary returned `data.frame` because it measures the fundamental sampling variability.
parent	The paraments of the parent distribution if `expand=TRUE`.
edist	The type of error distribution used to model the confidence interval if the argument `expand=TRUE` is set.
elmoms	The L-moment of the distribution of quantiles for the corresponding F if the argument `expand=TRUE` is set.
epara	An environment containing the parameter lists of the error distribution fit to the `elmoms` for each of the `f` if the argument `expand=TRUE` is set.
ifail	A failure integer.
ifailtext	Text message associated with `ifail`.

Author(s)

W.H. Asquith

See Also

genci, gen.freq.curves

Examples

```
## Not run:
# For all these examples, nsim is way too small.
mean   <- 0; sigma <- 100
parent <- vec2par(c(mean,sigma), type='nor') # make parameter object
f      <- c(0.5, 0.8, 0.9, 0.96, 0.98, 0.99) # nonexceed probabilities
# nsim is small for speed of example not accuracy.
CI     <- genci.simple(parent, n=10, f=f, nsim=20); FF <- CI$nonexceed
plot( FF, CI$true, type='l', lwd=2)
lines(FF, CI$lwr, col=2); lines(FF, CI$upr, col=3)

pdf("twoCIplots.pdf")
# The qnorm() call has been added to produce "normal probability"
# paper on the horizonal axis. The parent is heavy-tailed.
GEV  <- vec2par(c(10000,1500,-0.3), type='gev') # a GEV distribution
CI   <- genci.simple(GEV, n=20, nsim=200, edist='gno')
ymin <- log10(min(CI$lwr[! is.na(CI$lwr)]))
ymax <- log10(max(CI$upr[! is.na(CI$upr)]))
qFF  <- qnorm(CI$nonexceed)
plot( qFF, log10(CI$true), type='l', ylim=c(ymin,ymax),lwd=2)
lines(qFF, log10(CI$lwr), col=2); lines(qFF, log10(CI$upr), col=3)
# another error distribution model
CI   <- genci.simple(GEV, n=20, nsim=200, edist='aep4')
lines(qFF,log10(CI$lwr),col=2,lty=2); lines(qFF,log10(CI$upr),col=3,lty=2)
dev.off() #
## End(Not run)
```

`gini.mean.diff` *Gini Mean Difference Statistic*

Description

The Gini mean difference statistic $\mathcal{G}$ is a robust estimator of distribution scale and is closely related to the second L-moment $\lambda_2 = \mathcal{G}/2$.

$$\mathcal{G} = \frac{2}{n(n-1)}\sum_{i=1}^{n}(2i - n - 1)x_{i:n},$$

where $x_{i:n}$ are the sample order statistics.

Usage

```
gini.mean.diff(x)
```

Arguments

`x`	A vector of data values that will be reduced to non-missing values.

Value

An R `list` is returned.

`gini`	The gini mean difference $\mathcal{G}$.
`L2`	The L-scale (second L-moment) because $\lambda_2 = 0.5 \times \mathcal{G}$ (see `lmom.ub`).
`source`	An attribute identifying the computational source of the Gini's Mean Difference: "gini.mean.diff".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Jurečková, J., and Picek, J., 2006, Robust statistical methods with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 1–58488–454–1.

See Also

`lmoms`

Examples

```
fake.dat <- c(123, 34, 4, 654, 37, 78)
gini <- gini.mean.diff(fake.dat)
lmr <- lmoms(fake.dat)
str(gini)
print(abs(gini$L2 - lmr$lambdas[2]))
```

---

grv2prob *Convert a Vector of Gumbel Reduced Variates to Annual Nonexceedance Probabilities*

Description

This function converts a vector of Gumbel reduced variates (grv) to annual nonexceedance probabilities F

$$F = \exp(-\exp(-grv)),$$

where $0 \leq F \leq 1$.

Usage

```
grv2prob(grv)
```

Arguments

grv A vector of Gumbel reduced variates.

Value

A vector of annual nonexceedance probabilities.

Author(s)

W.H. Asquith

See Also

prob2grv, prob2T

Examples

```
T <- c(1, 2, 5, 10, 25, 50, 100, 250, 500); grv <- prob2grv(T2prob(T))
F <- grv2prob(grv)
```

harmonic.mean *The Harmonic Mean with Zero-Value Correction*

Description

Compute the harmonic mean of a vector with a zero-value correction.

$$\breve{\mu} = \left(\frac{\sum_{i=1}^{N_T - N_0} 1/x_i}{N_T - N_0}\right)^{-1} \times \frac{N_T - N_0}{N_T},$$

where $\breve{\mu}$ is harmonic mean, x_i is a nonzero value of the data vector, N_T is the (total) sample size, N_0 is the number of zero values.

Usage

```
harmonic.mean(x)
```

Arguments

`x`	A vector of data values that will be reduced to non-missing values.

Value

An R `list` is returned.

`harmean`	The harmonic mean with zero-value correction, $\breve{\mu}$.
`correction`	The zero-value correction, $(N_T - N_0)/N_T$.
`source`	An attribute identifying the computational source of the harmonic mean: "harmonic.mean".

Note

The harmonic mean can not be computed when zero values are present. This situation is common in surface-water hydrology. As stated in the reference below, in order to calculate water-quality-based effluent limits (WQBELs) for human health protection, a harmonic mean flow is determined for all perennial streams and for streams that are intermittent with perennial pools. Sometimes these streams have days on which measured flow is zero. Because a zero flow cannot be used in the calculation of harmonic mean flow, the second term in the harmonic mean equation is an adjustment factor used to lower the harmonic mean to compensate for days on which the flow was zero. The zero-value correction is the same correction used by the EPA computer program DFLOW.

Author(s)

W.H. Asquith

References

Texas Commission on Environmental Quality, 2003, Procedures to implement the Texas surface-water-quality standards: TCEQ RG–194, p. 47

See Also

pmoms

Examples

```
Q <- c(0,0,5,6,7)
harmonic.mean(Q)
```

---

headrick.sheng.lalpha *Sample Headrick and Sheng L-alpha*

Description

Compute the sample "Headrick and Sheng L-alpha" (α_L) (Headrick and Sheng, 2013) by

$$\alpha_L = \frac{d}{d-1}\left(1 - \frac{\sum_j \lambda_2^{(j)}}{\sum_j \lambda_2^{(j)} + \sum\sum_{j\neq j'} \lambda_2^{(jj')}}\right),$$

where $j = 1, \ldots, d$ for dimensions d, the $\sum_j \lambda_2^{(j)}$ is the summation of all the 2nd order (univariate) L-moments (L-scales, $\lambda_2^{(j)}$), and the double summation is the summation of all the 2nd-order L-comoments ($\lambda_2^{(jj')}$). In other words, the double summation is the sum total of all entries in both the lower and upper triangles (not the primary diagonal) of the L-comoment matrix (the L-scale and L-coscale [L-covariance] matrix) (Lcomoment.matrix).

The α_L is closely related in structural computation as the well-known "Cronbach alpha" (α_C). These are coefficients of reliability, which commonly ranges from 0 to 1, that provide what some methodologists portray as an overall assessment of a measure's reliability. If all of the scale items are entirely independent from one another, meaning that they are not correlated or share no covariance, then α_C is 0, and, if all of the items have high covariances, then α_C will approach 1 as the number of items in the scale approaches infinity. The higher the α_C coefficient, the more the items have shared covariance and probably measure the same underlying concept. Theoretically, there is no lower bounds for $\alpha_{C,L}$, which can add complicating nuances in bootstrap or simulation study of both α_C and α_L. Negative values are considered a sign of something potentially wrong about the measure related to items not being positively correlated with each other, or a scoring system for a question item reversed. (This paragraph in part paraphrases data.library.virginia.edu/using-and-interpreting-cronbachs-alpha/
(accessed May 21, 2023; dead link April 18, 2024) and other general sources.)

Usage

```
headrick.sheng.lalpha(x, bycovFF=FALSE, a=0.5, digits=8, ...)

lalpha(x, bycovFF=FALSE, a=0.5, digits=8, ...)
```

Arguments

- `x` — An R `data.frame` of the random observations for the d random variables X, which must be suitable for internal dispatch to the `Lcomoment.matrix` function for computation of the 2nd-order L-comoment matrix. Alternatively, `x` can be a precomputed 2nd-order L-comoment matrix (L-scale and L-coscale matrix) as shown by the following usage: `lalpha(Lcomoment.matrix(x, k=2)$matrix)`.
- `bycovFF` — A logical triggering the covariance pathway for the computation and bypassing the call to the L-comoments. The additional arguments can be used to control the `pp` function that is called internally to estimate nonexceedance probabilities and the "covariance pathway" (see **Details**). If `bycovFF` is `FALSE`, then the direct to L-comoment computation is used.
- `a` — The plotting position argument `a` to the `pp` function that is hardwired here to Hazen in contrast to the default `a=0` of `pp` (Weibull) for reasoning shown in this documentation.
- `digits` — Number of digits for rounding on the returned value(s).
- `...` — Additional arguments to pass.

Details

Headrick and Sheng (2013) propose α_L to be an alternative estimator of reliability based on L-comoments. Those authors describe its context as follows: "Consider [a statistic] alpha (α) in terms of a model that decomposes an observed score into the sum of two independent components: a true unobservable score t_i and a random error component ϵ_{ij}."

Those authors continue "The model can be summarized as $X_{ij} = t_i + \epsilon_{ij}$, where X_{ij} is the observed score associated with the ith examinee on the jth test item, and where $i = 1, ..., n$ [for sample size n]; $j = 1, \ldots, d$; and the error terms (ϵ_{ij}) are independent with a mean of zero." Those authors comment that "inspection of [this model] indicates that this particular model restricts the true score t_i to be the same across all d test items."

Those authors show empirical results for a simulation study, which indicate that α_L can be "substantially superior" to [a different formulation of α_C (Cronbach's alpha) based on product moments (the variance-covariance matrix)] in "terms of relative bias and relative standard error when distributions are heavy-tailed and sample sizes are small."

Those authors show (Headrick and Sheng, 2013, eqs. 4 and 5) the reader that the second L-comoments of X_j and $X_{j'}$ can alternatively be expressed as $\lambda_2(X_j) = 2\mathrm{Cov}(X_j, F(X_j))$ and $\lambda_2(X_{j'}) = 2\mathrm{Cov}(X_{j'}, F(X_{j'}))$. The second L-comoments of X_j toward (with respect to) $X_{j'}$ and $X_{j'}$ toward (with respect to) X_j are $\lambda_2^{(jj')} = 2\mathrm{Cov}(X_j, F(X_{j'}))$ and $\lambda_2^{(j'j)} = 2\mathrm{Cov}(X_{j'}, F(X_j))$. The respective cumulative distribution functions are denoted $F(x_j)$ (nonexceedance probabilities). Evidently, those authors present the L-moments and L-comoments this way because their first example (thanks for detailed numerics!) already contain nonexceedance probabilities.

This apparent numerical difference between the version using estimates of nonexceedance probabilities for the data (the "covariance pathway") compared to a "direct to L-comoment" pathway might be more than academic concern.

The **Examples** provide comparison and brief discussion of potential issues involved in the direct L-comoments and the covariance pathway. The discussion leads to interest in the effects of ties and their handling and the question of $F(x_j)$ estimation by plotting position (`pp`). The **Note** section of this documentation provides expanded information and insights to α_L computation.

Value

An R `list` is returned.

`alpha`	The α_L statistic.
`pitems`	The number of items (column count) in the `x`.
`n`	The sample size (row count), if applicable, to the contents of `x`.
`text`	Any pertinent messages about the computations.
`source`	An attribute identifying the computational source of the Headrick and Sheng L-alpha: "headrick.sheng.lalpha" or "lalpha.star()".

Note

Headrick and Sheng (2013) use k to represent d as used here. The change is made because `k` is an L-comoment order argument already in use by `Lcomoment.matrix`.

Demonstration of Nuances of L-alpha—Consider Headrick and Sheng (2013, tables 1 and 2) and the effect of those authors' covariance pathway to α_L:

```
X1 <- c(2, 5, 3, 6, 7, 5, 2, 4, 3, 4) # Table 1 in Headrick and Sheng (2013)
X2 <- c(4, 7, 5, 6, 7, 2, 3, 3, 5, 4)
X3 <- c(3, 7, 5, 6, 6, 6, 3, 6, 5, 5)
X  <- data.frame(X1=X1, X2=X2, X3=X3)
lcm2 <- Lcomoment.matrix(X, k=2)
print(lcm2$matrix, 3)
#       [,1]  [,2]  [,3]
# [1,] 0.989 0.567 0.722
# [2,] 0.444 1.022 0.222
# [3,] 0.644 0.378 0.733
```

Now, compare the above matrix to Headrick and Sheng (2013, table 2) where it is immediately seen that the matrices are not the same before the summations are applied to compute α_L.

```
#       [,1]  [,2]  [,3]
# [1,] 0.989 0.500 0.789
# [2,] 0.500 1.022 0.411
# [3,] 0.667 0.333 0.733
```

Now, consider how the nonexceedances in Headrick and Sheng (2013, table 1) might have been computed w/o following their citation to original sources. It can be shown with reference to the first example above that these nonexceedance probabilities match.

```
FX1 <- rank(X$X1, ties.method="average") / length(X$X1)
FX2 <- rank(X$X2, ties.method="average") / length(X$X2)
FX3 <- rank(X$X3, ties.method="average") / length(X$X3)
```

Notice in Headrick and Sheng (2013, table 1) that there is no zero probability, but there is a unity and some of the probabilities are tied. Ties have numerical ramifications. Let us now look at other L-alphas using the nonexceedance pathway and use different definitions of nonexceedance estimation and inspect the results:

```
# lmomco documentation says pp() uses ties.method="first"
lalpha(X, bycovFF=TRUE, a=0     )$alpha
# [1] 0.7448583  # unbiased probs all distributions
lalpha(X, bycovFF=TRUE, a=0.3173)$alpha
# [1] 0.7671384  # Median probs for all distributions
lalpha(X, bycovFF=TRUE, a=0.35  )$alpha
# [1] 0.7695105  # Often used with probs-weighted moments
lalpha(X, bycovFF=TRUE, a=0.375 )$alpha
# [1] 0.771334   # Blom, nearly unbiased quantiles for normal
lalpha(X, bycovFF=TRUE, a=0.40  )$alpha
# [1] 0.7731661  # Cunnane, appox quantile unbiased
lalpha(X, bycovFF=TRUE, a=0.44  )$alpha
# [1] 0.7761157  # Gringorten, optimized for Gumbel
lalpha(X, bycovFF=TRUE, a=0.5   )$alpha
# [1] 0.7805825  # Hazen, traditional choice
                 # This the plotting position (i-0.5) / n
```

This is not a particularly pleasing situation because the choice of the plotting position affects the α_L. The Hazen definition `lalpha(X[,1:3], bycovFF=FALSE)` using direct to L-comoments matches the last computation shown ($\alpha_L = 0.7805825$). A question, thus, is does this matching occur because of the nature of the ties and structure of the L-comoment algorithm itself? A note to this question involves a recognition that the answer is yes because L-comoments use a `sort()` operation and does not use `rank()` because the weights for the linear combinations are used and the covariance pathway `2*cov(x$X3, x$FX2)`, for instance.

Recognizing that the direct to L-comoments alpha equals the covariance pathway with Hazen plotting positions, let us look at L-comoments:

```
lmomco::Lcomoment.Lk12 ------> snippet
     X12 <- X1[sort(X2, decreasing = FALSE, index.return = TRUE)$ix]
     n <- length(X1)
     SUM <- sum(sapply(1:n, function(r) { Lcomoment.Wk(k, r, n) * X12[r] }))
     return(SUM/n)
```

Notice that a `ties.method` is not present but kind of implicit as ties first by the index return of the `sort()` and notice the return of a `SUM/n` though this is an L-comoment and not an nonexceedance probability.

Let us run through the tie options using a plotting position definition (i/n) matching the computations of Headrick and Sheng (2013) (`"average"`, `A=0`, `B=0` for pp) and the first computation $\alpha_L = 0.807$ matches that reported by Headrick and Sheng (2013, p. 4):

```
for(tie in c("average", "first", "last", "min", "max")) { # random left out
  Lalpha <- lalpha(X, bycovFF=TRUE,
                    a=NULL, A=0, B=0, ties.method=tie)$alpha
  message("Ties method ", stringr::str_pad(tie, 7, pad=" "),
          " provides L-alpha = ", Lalpha)
}
# Ties method average provides L-alpha = 0.80747664
# Ties method   first provides L-alpha = 0.78058252
```

```
# Ties method    last provides L-alpha = 0.83243243
# Ties method     min provides L-alpha = 0.81363468
# Ties method     max provides L-alpha = 0.80120709
```

Let us run through the tie options again using a different plotting position estimator ($(n-0.5)/(n+0.5)$):

```
for(tie in c("average", "first", "last", "min", "max")) { # random left out
  Lalpha <- lalpha(X, bycovFF=TRUE,
                      a=NULL, A=-0.5, B=0.5, ties.method=tie)$alpha
  message("Ties method ", stringr::str_pad(tie, 7, pad=" "),
          " provides L-alpha = ", Lalpha)
}
# Ties method average provides L-alpha = 0.78925733
# Ties method   first provides L-alpha = 0.76230208
# Ties method    last provides L-alpha = 0.81431215
# Ties method     min provides L-alpha = 0.79543602
# Ties method     max provides L-alpha = 0.78296931
```

We see obviously that the decision on how to treat ties has some influence on the computation involving the covariance pathway. This is not an entirely satisfactory situation, but perhaps the distinction does not matter much? The direct L-comoment pathway seems to avoid this issue because `sort()` is stable and like `ties.method="first"`. Experiments suggest that `a=0.5` (Hazen plotting positions) produces the same results as direct L-comoment (see the next section). However, as the following code set shows:

```
for(tie in c("average", "first", "last", "min", "max")) { # random left out
  Lalpha1 <- lalpha(X, bycovFF=TRUE, a=0.5, ties.method=tie)$alpha
 Lalpha2 <- lalpha(X, bycovFF=TRUE, a=NULL, A=-0.5, B=0, ties.method=tie)$alpha
 Lalpha3 <- lalpha(X, bycovFF=TRUE, a=NULL, A=-1  , B=0, ties.method=tie)$alpha
 Lalpha4 <- lalpha(X, bycovFF=TRUE, a=NULL, A=   0, B=0, ties.method=tie)$alpha
  print(c(Lalpha1, Lalpha2, Lalpha3, Lalpha4))
}
```

The α_L for a given tie setting are all equal as long as the demoninator of the plotting position ($(i + A)/(n + B)$) has `B=0`. The `a=0.5` produces Hazen, the `a=NULL, A=-0.5` produces Hazen, though `a=NULL, A=-1` (lower limit of `A`) and `a=NULL, A=0` (upper limit of `A` given `B`) also produces the same. This gives us as-implemented-proof that the sensitivity to the α_L computation is in the sorting and the denominator of the plotting position formula. The prudent default settings for when the `bycovFF` argument is true seems to have the $a = -0.5$ as nonexceedance probabilities are computed by the well-known Hazen description and with the tie method as first, the computations match direct to L-comoments.

Demonstration of Computational Times—A considerable amount of documentation and examples are provided here about the two pathways that α_L can be computed: (1) direct by L-comoments or (2) covariance pathway requiring precomputed estimates of the nonexceedance probabilities using a `ties.method="first"` (default pp). The following example shows numerical congruence between the two pathways if the so-called Hazen plotting positions (`a=0.5`, see pp) are requested with the implicit default of `ties.method="first"`. However, the computational time of the direct method is quite a bit longer because of latencies in the weight factor computations involved in the L-comoments and nested `for` loops.

```
set.seed(1)
R <- 1:10; nsam <- 1E5 # random and uncorrelated scores in this measure
Z <- data.frame( I1=sample(R, nsam, replace=TRUE),
                 I2=sample(R, nsam, replace=TRUE),
                 I3=sample(R, nsam, replace=TRUE),
                 I4=sample(R, nsam, replace=TRUE) )
system.time(AnF <- headrick.sheng.lalpha(Z, bycovFF=FALSE)$alpha)
system.time(AwF <- headrick.sheng.lalpha(Z, bycovFF=TRUE )$alpha) # Hazen
#    user  system elapsed
#  30.382   0.095  30.501    AnF ---> 0.01370302
#    user  system elapsed
#   5.054   0.030   5.092    AwF ---> 0.01370302
```

Author(s)

W.H. Asquith

References

Headrick, T.C., and Sheng, Y., 2013, An alternative to Cronbach's alpha—An L-moment-based measure of internal-consistency reliability: *in* Millsap, R.E., van der Ark, L.A., Bolt, D.M., Woods, C.M. (eds) New Developments in Quantitative Psychology, Springer Proceedings in Mathematics and Statistics, v. 66, doi:10.1007/9781461493488_2.

Headrick, T.C., and Sheng, Y., 2013, A proposed measure of internal consistency reliability—Coefficient L-alpha: Behaviormetrika, v. 40, no. 1, pp. 57–68, doi:10.2333/bhmk.40.57.

Béland, S., Cousineau, D., and Loye, N., 2017, Using the McDonald's omega coefficient instead of Cronbach's alpha [French]: McGill Journal of Education, v. 52, no. 3, pp. 791–804, doi:10.7202/1050915ar.

See Also

Lcomoment.matrix, pp

Examples

```
# Table 1 in Headrick and Sheng (2013)
TV1 <- # Observations in cols 1:3, estimated nonexceedance probabilities in cols 4:6
c(2, 4, 3, 0.15, 0.45, 0.15,       5, 7, 7, 0.75, 0.95, 1.00,
  3, 5, 5, 0.35, 0.65, 0.40,       6, 6, 6, 0.90, 0.80, 0.75,
  7, 7, 6, 1.00, 0.95, 0.75,       5, 2, 6, 0.75, 0.10, 0.75,
  2, 3, 3, 0.15, 0.25, 0.15,       4, 3, 6, 0.55, 0.25, 0.75,
  3, 5, 5, 0.35, 0.65, 0.40,       4, 4, 5, 0.55, 0.45, 0.40)
T1 <- matrix(ncol=6, nrow=10)
for(r in seq(1,length(TV1), by=6)) T1[(r/6)+1, ] <- TV1[r:(r+5)]
colnames(T1) <- c("X1", "X2", "X3", "FX1", "FX2", "FX3"); T1 <- as.data.frame(T1)

lco2 <- matrix(nrow=3, ncol=3)
lco2[1,1] <- lmoms(T1$X1)$lambdas[2]
lco2[2,2] <- lmoms(T1$X2)$lambdas[2]
lco2[3,3] <- lmoms(T1$X3)$lambdas[2]
```

```
lco2[1,2] <- 2*cov(T1$X1, T1$FX2); lco2[1,3] <- 2*cov(T1$X1, T1$FX3)
lco2[2,1] <- 2*cov(T1$X2, T1$FX1); lco2[2,3] <- 2*cov(T1$X2, T1$FX3)
lco2[3,1] <- 2*cov(T1$X3, T1$FX1); lco2[3,2] <- 2*cov(T1$X3, T1$FX2)
headrick.sheng.lalpha(lco2)$alpha     # Headrick and Sheng (2013): alpha = 0.807
# 0.8074766
headrick.sheng.lalpha(Lcomoment.matrix(T1[,1:3], k=2)$matrix)$alpha
# 0.7805825
headrick.sheng.lalpha(T1[,1:3])$alpha #              FXs not used: alpha = 0.781
# 0.7805825
headrick.sheng.lalpha(T1[,1:3], bycovFF=TRUE)$alpha  # a=0.5, Hazen by default
# 0.7805825
headrick.sheng.lalpha(T1[,1:3], bycovFF=TRUE, a=0.5)$alpha
# 0.7805825
```

`herefordprecip` *Annual Maximum Precipitation Data for Hereford, Texas*

Description

Annual maximum precipitation data for Hereford, Texas

Usage

```
data(herefordprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(herefordprecip)
summary(herefordprecip)
```

`hlmomco` *Hazard Functions of the Distributions*

Description

This function acts as a front end to `dlmomco` and `plmomco` to compute the hazard function $h(x)$ or conditional failure rate. The function is defined by

$$h(x) = \frac{f(x)}{1 - F(x)},$$

where $f(x)$ is a probability density function and $F(x)$ is the cumulative distribution function.

To help with intuitive understanding of what $h(x)$ means (Ugarte and others, 2008), let $\mathrm{d}x$ represent a small unit of measurement. Then the quantity $h(x)\,\mathrm{d}x$ can be conceptualized as the approximate probability that random variable X takes on a value in the interval $[x, x + \mathrm{d}x]$.

Ugarte and others (2008) continue by stating that $h(x)$ represents the instantaneous rate of death or failure at time x, given the survival to time x has occurred. Emphasis is needed that $h(x)$ is a rate of probability change and not a probability itself.

Usage

```
hlmomco(x,para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `lmom2par` or similar.

Value

Hazard rate for `x`.

Note

The hazard function is numerically solved for the given cumulative distribution and probability density functions and not analytical expressions for the hazard function that do exist for many distributions.

Author(s)

W.H. Asquith

References

Ugarte, M.D., Militino, A.F., and Arnholt, A.T., 2008, Probability and statistics with R: CRC Press, Boca Raton, FL.

See Also

plmomco, dlmomco

Examples

```
my.lambda <- 100
para <- vec2par(c(0,my.lambda), type="exp")

x <- seq(40:60)
hlmomco(x,para) # returns vector of 0.01
# because the exponential distribution has a constant
# failure rate equal to 1/scale or 1/100 as in this example.
```

IRSrefunds.by.state *U.S. Internal Revenue Service Refunds by State for Fiscal Year 2006*

Description

U.S. Internal Revenue Service refunds by state for fiscal year 2006.

Usage

```
data(IRSrefunds.by.state)
```

Format

A data frame with

STATE State name.

REFUNDS Dollars of refunds.

Examples

```
data(IRSrefunds.by.state)
summary(IRSrefunds.by.state)
```

is.aep4 — *Is a Distribution Parameter Object Typed as 4-Parameter Asymmetric Exponential Power*

Description

The distribution parameter object returned by functions of **lmomco** such as by `paraep4` are typed by an attribute `type`. This function checks that `type` is `aep4` for the 4-parameter Asymmetric Exponential Power distribution.

Usage

```
is.aep4(para)
```

Arguments

`para`	A parameter `list` returned from `paraep4` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `aep4`.
`FALSE`	If the `type` is not `aep4`.

Author(s)

W.H. Asquith

See Also

`paraep4`

Examples

```
para <- vec2par(c(0,1, 0.5, 4), type="aep4")
if(is.aep4(para) == TRUE) {
  Q <- quaaep4(0.55,para)
}
```

is.cau *Is a Distribution Parameter Object Typed as Cauchy*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parcau` are typed by an attribute `type`. This function checks that `type` is `cau` for the Cauchy distribution.

Usage

```
is.cau(para)
```

Arguments

`para`	A parameter `list` returned from `parcau` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `cau`.
`FALSE`	If the `type` is not `cau`.

Author(s)

W.H. Asquith

See Also

`parcau`

Examples

```
para <- vec2par(c(12,12),type='cau')
if(is.cau(para) == TRUE) {
  Q <- quacau(0.5,para)
}
```

is.emu *Is a Distribution Parameter Object Typed as Eta-Mu*

Description

The distribution parameter object returned by functions of **lmomco** such as by `paremu` are typed by an attribute `type`. This function checks that `type` is `emu` for the Eta-Mu ($\eta : \mu$) distribution.

Usage

```
is.emu(para)
```

Arguments

`para`	A parameter `list` returned from `paremu` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `emu`.
`FALSE`	If the `type` is not `emu`.

Author(s)

W.H. Asquith

See Also

`paremu`

Examples

```
## Not run:
para <- vec2par(c(0.25, 1.4), type='emu')
if(is.emu(para)) Q <- quaemu(0.5,para) #
## End(Not run)
```

---

is.exp — *Is a Distribution Parameter Object Typed as Exponential*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parexp` are typed by an attribute `type`. This function checks that `type` is `exp` for the Exponential distribution.

Usage

```
is.exp(para)
```

Arguments

`para`	A parameter `list` returned from `parexp` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `exp`.
`FALSE`	If the `type` is not `exp`.

Author(s)

W.H. Asquith

See Also

parexp

Examples

```
para <- parexp(lmoms(c(123,34,4,654,37,78)))
if(is.exp(para) == TRUE) {
  Q <- quaexp(0.5,para)
}
```

---

is.gam *Is a Distribution Parameter Object Typed as Gamma*

Description

The distribution parameter object returned by functions of **lmomco** such as by pargam are typed by an attribute type. This function checks that type is gam for the Gamma distribution.

Usage

```
is.gam(para)
```

Arguments

para	A parameter list returned from pargam or vec2par.

Value

TRUE	If the type attribute is gam.
FALSE	If the type is not gam.

Author(s)

W.H. Asquith

See Also

pargam

Examples

```
para <- pargam(lmoms(c(123,34,4,654,37,78)))
if(is.gam(para) == TRUE) {
  Q <- quagam(0.5,para)
}
```

is.gdd — *Is a Distribution Parameter Object Typed as Gamma Difference*

Description

The distribution parameter object returned by functions of **lmomco** such as by pargdd are typed by an attribute `type`. This function checks that `type` is gdd for the Gamma Difference distribution.

Usage

```
is.gdd(para)
```

Arguments

`para`	A parameter `list` returned from pargdd or vec2par.

Value

`TRUE`	If the `type` attribute is gdd.
`FALSE`	If the `type` is not gdd.

Author(s)

W.H. Asquith

See Also

pargdd

Examples

```
#
```

is.gep — *Is a Distribution Parameter Object Typed as Generalized Extreme Value*

Description

The distribution parameter object returned by functions of **lmomco** such as by pargep are typed by an attribute `type`. This function checks that `type` is gep for the Generalized Extreme Value distribution.

Usage

```
is.gep(para)
```

Arguments

para A parameter `list` returned from `pargep` or `vec2par`.

Value

TRUE If the `type` attribute is gep.

FALSE If the `type` is not gep.

Author(s)

W.H. Asquith

See Also

`pargep`

Examples

```
#para <- pargep(lmoms(c(123,34,4,654,37,78)))
#if(is.gep(para) == TRUE) {
#  Q <- quagep(0.5,para)
#}
```

---

is.gev *Is a Distribution Parameter Object Typed as Generalized Extreme Value*

Description

The distribution parameter object returned by functions of **lmomco** such as by `pargev` are typed by an attribute `type`. This function checks that `type` is `gev` for the Generalized Extreme Value distribution.

Usage

```
is.gev(para)
```

Arguments

para A parameter `list` returned from `pargev` or `vec2par`.

Value

TRUE If the `type` attribute is `gev`.

FALSE If the `type` is not `gev`.

Author(s)

W.H. Asquith

See Also

pargev

Examples

```
para <- pargev(lmoms(c(123,34,4,654,37,78)))
if(is.gev(para) == TRUE) {
  Q <- quagev(0.5,para)
}
```

---

is.gld — *Is a Distribution Parameter Object Typed as Generalized Lambda*

Description

The distribution parameter object returned by functions of **lmomco** such as by `pargld` are typed by an attribute `type`. This function checks that `type` is `gld` for the Generalized Lambda distribution.

Usage

```
is.gld(para)
```

Arguments

`para`	A parameter `list` returned from `pargld` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `gld`.
`FALSE`	If the `type` is not `gld`.

Author(s)

W.H. Asquith

See Also

pargld

Examples

```
## Not run:
para <- vec2par(c(123,120,3,2),type="gld")
if(is.gld(para) == TRUE) {
  Q <- quagld(0.5,para)
}

## End(Not run)
```

---

is.glo   *Is a Distribution Parameter Object Typed as Generalized Logistic*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parglo` are typed by an attribute `type`. This function checks that `type` is `glo` for the Generalized Logistic distribution.

Usage

```
is.glo(para)
```

Arguments

`para`	A parameter `list` returned from `parglo` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `glo`.
`FALSE`	If the `type` is not `glo`.

Author(s)

W.H. Asquith

See Also

`parglo`

Examples

```
para <- parglo(lmoms(c(123,34,4,654,37,78)))
if(is.glo(para) == TRUE) {
  Q <- quaglo(0.5,para)
}
```

is.gno *Is a Distribution Parameter Object Typed as Generalized Normal*

Description

The distribution parameter object returned by functions of **lmomco** such as by `pargno` are typed by an attribute `type`. This function checks that `type` is `gno` for the Generalized Normal distribution.

Usage

```
is.gno(para)
```

Arguments

`para`	A parameter `list` returned from `pargno` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `gno`.
`FALSE`	If the `type` is not `gno`.

Author(s)

W.H. Asquith

See Also

`pargno`

Examples

```
para <- pargno(lmoms(c(123,34,4,654,37,78)))
if(is.gno(para) == TRUE) {
  Q <- quagno(0.5,para)
}
```

is.gov *Is a Distribution Parameter Object Typed as Govindarajulu*

Description

The distribution parameter object returned by functions of **lmomco** such as by `pargov` are typed by an attribute `type`. This function checks that `type` is `gov` for the Govindarajulu distribution.

Usage

```
is.gov(para)
```

Arguments

para	A parameter `list` returned from `pargov` or `vec2par`.

Value

TRUE	If the `type` attribute is gov.
FALSE	If the `type` is not gov.

Author(s)

W.H. Asquith

See Also

`pargov`

Examples

```
para <- pargov(lmoms(c(123,34,4,654,37,78)))
if(is.gov(para) == TRUE) {
  Q <- quagov(0.5,para)
}
```

---

is.gpa *Is a Distribution Parameter Object Typed as Generalized Pareto*

Description

The distribution parameter object returned by functions of **lmomco** such as by `pargpa` are typed by an attribute `type`. This function checks that `type` is gpa for the Generalized Pareto distribution.

Usage

```
is.gpa(para)
```

Arguments

para	A parameter `list` returned from `pargpa` or `vec2par`.

Value

TRUE	If the `type` attribute is gpa.
FALSE	If the `type` is not gpa.

Author(s)

W.H. Asquith

See Also

pargpa

Examples

```
para <- pargpa(lmoms(c(123,34,4,654,37,78)))
if(is.gpa(para) == TRUE) {
  Q <- quagpa(0.5,para)
}
```

---

is.gum *Is a Distribution Parameter Object Typed as Gumbel*

Description

The distribution parameter object returned by functions of **lmomco** such as by `pargum` are typed by an attribute `type`. This function checks that `type` is `gum` for the Gumbel distribution.

Usage

```
is.gum(para)
```

Arguments

para	A parameter `list` returned from `pargum` or `vec2par`.

Value

TRUE	If the `type` attribute is `gum`.
FALSE	If the `type` is not `gum`.

Author(s)

W.H. Asquith

See Also

pargum

Examples

```
para <- pargum(lmoms(c(123,34,4,654,37,78)))
if(is.gum(para) == TRUE) {
  Q <- quagum(0.5,para)
}
```

is.kap — *Is a Distribution Parameter Object Typed as Kappa*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parkap` are typed by an attribute `type`. This function checks that `type` is `kap` for the Kappa distribution.

Usage

```
is.kap(para)
```

Arguments

`para`	A parameter `list` returned from `parkap` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `kap`.
`FALSE`	If the `type` is not `kap`.

Author(s)

W.H. Asquith

See Also

`parkap`

Examples

```
para <- parkap(lmoms(c(123,34,4,654,37,78)))
if(is.kap(para) == TRUE) {
  Q <- quakap(0.5,para)
}
```

is.kmu — *Is a Distribution Parameter Object Typed as Kappa-Mu*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parkmu` are typed by an attribute `type`. This function checks that `type` is `kmu` for the Kappa-Mu ($\kappa : \mu$) distribution.

Usage

```
is.kmu(para)
```

Arguments

`para`	A parameter `list` returned from `parkmu` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `kmu`.
`FALSE`	If the `type` is not `kmu`.

Author(s)

W.H. Asquith

See Also

`parkmu`

Examples

```
para <- vec2par(c(3.1, 1.4), type='kmu')
if(is.kmu(para)) {
  Q <- quakmu(0.5,para)
}
```

---

is.kur *Is a Distribution Parameter Object Typed as Kumaraswamy*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parkur` are typed by an attribute `type`. This function checks that `type` is `kur` for the Kumaraswamy distribution.

Usage

```
is.kur(para)
```

Arguments

`para`	A parameter `list` returned from `parkur` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `kur`.
`FALSE`	If the `type` is not `kur`.

Author(s)

W.H. Asquith

See Also

parkur

Examples

```
para <- parkur(lmoms(c(0.25, 0.4, 0.6, 0.65, 0.67, 0.9)))
if(is.kur(para) == TRUE) {
  Q <- quakur(0.5,para)
}
```

---

is.lap *Is a Distribution Parameter Object Typed as Laplace*

Description

The distribution parameter object returned by functions of **lmomco** such as by parlap are typed by an attribute type. This function checks that type is lap for the Laplace distribution.

Usage

```
is.lap(para)
```

Arguments

para	A parameter list returned from parlap or vec2par.

Value

TRUE	If the type attribute is lap.
FALSE	If the type is not lap.

Author(s)

W.H. Asquith

See Also

parlap

Examples

```
para <- parlap(lmoms(c(123,34,4,654,37,78)))
if(is.lap(para) == TRUE) {
  Q <- qualap(0.5,para)
}
```

is.lmrq — *Is a Distribution Parameter Object Typed as Linear Mean Residual Quantile Function*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parlmrq` are typed by an attribute `type`. This function checks that `type` is `lmrq` for the Linear Mean Residual Quantile Function distribution.

Usage

```
is.lmrq(para)
```

Arguments

`para`	A parameter `list` returned from `parlmrq` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `lmrq`.
`FALSE`	If the `type` is not `lmrq`.

Author(s)

W.H. Asquith

See Also

`parlmrq`

Examples

```
para <- parlmrq(lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2)))
if(is.lmrq(para) == TRUE) {
  Q <- qualmrq(0.5,para)
}
```

`is.ln3` *Is a Distribution Parameter Object Typed as 3-Parameter Log-Normal*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parln3` are typed by an attribute `type`. This function checks that `type` is `ln3` for the 3-parameter Log-Normal distribution.

Usage

```
is.ln3(para)
```

Arguments

`para`	A parameter `list` returned from `parln3` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `ln3`.
`FALSE`	If the `type` is not `ln3`.

Author(s)

W.H. Asquith

See Also

`parln3`

Examples

```
para <- vec2par(c(.9252, .1636, .7),type='ln3')
if(is.ln3(para)) {
  Q <- qualn3(0.5,para)
}
```

is.nor *Is a Distribution Parameter Object Typed as Normal*

Description

The distribution parameter object returned by functions of **lmomco** such as by parnor are typed by an attribute type. This function checks that type is nor for the Normal distribution.

Usage

```
is.nor(para)
```

Arguments

para	A parameter list returned from parnor or vec2par.

Value

TRUE	If the type attribute is nor.
FALSE	If the type is not nor.

Author(s)

W.H. Asquith

See Also

parnor

Examples

```
para <- parnor(lmoms(c(123,34,4,654,37,78)))
if(is.nor(para) == TRUE) {
  Q <- quanor(0.5,para)
}
```

is.pdq3 *Is a Distribution Parameter Object Typed as Polynomial Density-Quantile3*

Description

The distribution parameter object returned by functions of **lmomco** such as by parpdq3 are typed by an attribute type. This function checks that type is pdq3 for the Polynomial Density-Quantile3 distribution.

Usage

```
is.pdq3(para)
```

Arguments

para	A parameter `list` returned from parpdq3 or vec2par.

Value

TRUE	If the `type` attribute is pdq3.
FALSE	If the `type` is not pdq3.

Author(s)

W.H. Asquith

See Also

parpdq3

Examples

```
para <- parpdq3(lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52)))
if(is.pdq3(para) == TRUE) {
  Q <- quapdq3(0.5, para)
}
```

---

is.pdq4 *Is a Distribution Parameter Object Typed as Polynomial Density-Quantile4*

Description

The distribution parameter object returned by functions of **lmomco** such as by parpdq4 are typed by an attribute `type`. This function checks that `type` is pdq4 for the Polynomial Density-Quantile4 distribution.

Usage

```
is.pdq4(para)
```

Arguments

para	A parameter `list` returned from parpdq4 or vec2par.

Value

TRUE	If the `type` attribute is pdq4.
FALSE	If the `type` is not pdq4.

Author(s)

W.H. Asquith

See Also

parpdq4

Examples

```
para <- parpdq4(lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52)))
if(is.pdq4(para) == TRUE) {
  Q <- quapdq4(0.5, para)
}
```

---

is.pe3 *Is a Distribution Parameter Object Typed as Pearson Type III*

Description

The distribution parameter object returned by functions of **lmomco** such as by parpe3 are typed by an attribute type. This function checks that type is pe3 for the Pearson Type III distribution.

Usage

```
is.pe3(para)
```

Arguments

para	A parameter list returned from parpe3 or vec2par.

Value

TRUE	If the type attribute is pe3.
FALSE	If the type is not pe3.

Author(s)

W.H. Asquith

See Also

parpe3

Examples

```
para <- parpe3(lmoms(c(123,34,4,654,37,78)))
if(is.pe3(para) == TRUE) {
  Q <- quape3(0.5,para)
}
```

is.ray *Is a Distribution Parameter Object Typed as Rayleigh*

Description

The distribution parameter object returned by functions of this module such as by `parray` are typed by an attribute `type`. This function checks that `type` is `ray` for the Rayleigh distribution.

Usage

```
is.ray(para)
```

Arguments

para	A parameter `list` returned from `parray` or `vec2par`.

Value

TRUE	If the `type` attribute is `ray`.
FALSE	If the `type` is not `ray`.

Author(s)

W.H. Asquith

See Also

`parray`

Examples

```
para <- vec2par(c(.9252, .1636, .7),type='ray')
if(is.ray(para)) {
  Q <- quaray(0.5,para)
}
```

is.revgum *Is a Distribution Parameter Object Typed as Reverse Gumbel*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parrevgum` are typed by an attribute `type`. This function checks that `type` is `revgum` for the Reverse Gumbel distribution.

Usage

```
is.revgum(para)
```

Arguments

para	A parameter `list` returned from `parrevgum` or `vec2par`.

Value

TRUE	If the `type` attribute is `revgum`.
FALSE	If the `type` is not `revgum`.

Author(s)

W.H. Asquith

See Also

`parrevgum`

Examples

```
para <- vec2par(c(.9252, .1636, .7),type='revgum')
if(is.revgum(para)) {
  Q <- quarevgum(0.5,para)
}
```

---

is.rice *Is a Distribution Parameter Object Typed as Rice*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parrice` are typed by an attribute `type`. This function checks that `type` is `rice` for the Rice distribution.

Usage

```
is.rice(para)
```

Arguments

para	A parameter `list` returned from `parrice` or `vec2par`.

Value

TRUE	If the `type` attribute is `rice`.
FALSE	If the `type` is not `rice`.

Author(s)

W.H. Asquith

See Also

parrice

Examples

```
para <- vec2par(c(3, 4),type='rice')
if(is.rice(para)) {
  Q <- quarice(0.5,para)
}
```

---

is.sla *Is a Distribution Parameter Object Typed as Slash*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parsla` are typed by an attribute `type`. This function checks that `type` is `sla` for the Slash distribution.

Usage

```
is.sla(para)
```

Arguments

`para`	A parameter `list` returned from `parsla` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `sla`.
`FALSE`	If the `type` is not `sla`.

Author(s)

W.H. Asquith

See Also

parsla

Examples

```
para <- vec2par(c(12, 1.2), type="sla")
if(is.sla(para) == TRUE) {
  Q <- quasla(0.5, para)
}
```

is.smd — *Is a Distribution Parameter Object Typed as Singh–Maddala*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parsmd` are typed by an attribute `type`. This function checks that `type` is `smd` for the Singh–Maddala distribution.

Usage

```
is.smd(para)
```

Arguments

`para`	A parameter `list` returned from `parsmd` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `smd`.
`FALSE`	If the `type` is not `smd`.

Author(s)

W.H. Asquith

See Also

`parsmd`

Examples

```
para <- parsmd(lmoms(c(123, 34, 4, 654, 37, 78)))
if(is.smd(para) == TRUE) {
  Q <- quasmd(0.5, para)
}
```

is.st3 — *Is a Distribution Parameter Object Typed as 3-Parameter Student t Distribution*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parst3` are typed by an attribute `type`. This function checks that `type` is `st3` for the 3-parameter Student t distribution.

Usage

```
is.st3(para)
```

Arguments

para	A parameter list returned from parst3 or vec2par.

Value

TRUE	If the type attribute is st3.
FALSE	If the type is not st3.

Author(s)

W.H. Asquith

See Also

parst3

Examples

```
para <- vec2par(c(3, 4, 5), type='st3')
if(is.st3(para)) {
  Q <- quast3(0.25,para)
}
```

---

is.texp *Is a Distribution Parameter Object Typed as Truncated Exponential*

Description

The distribution parameter object returned by functions of **lmomco** such as by partexp are typed by an attribute type. This function checks that type is texp for the Truncated Exponential distribution.

Usage

```
is.texp(para)
```

Arguments

para	A parameter list returned from partexp or vec2par.

Value

TRUE	If the type attribute is texp.
FALSE	If the type is not texp.

Author(s)

W.H. Asquith

See Also

partexp

Examples

```
yy <- vec2par(c(123, 2.3, TRUE),  type="texp")
zz <- vec2par(c(123, 2.3, FALSE), type="texp")
if(is.texp(yy) & is.texp(zz)) {
   print(lmomtexp(yy)$lambdas)
   print(lmomtexp(zz)$lambdas)
}
```

---

is.tri *Is a Distribution Parameter Object Typed as Asymmetric Triangular*

Description

The distribution parameter object returned by functions of **lmomco** such as by `partri` are typed by an attribute `type`. This function checks that `type` is `tri` for the Asymmetric Triangular distribution.

Usage

```
is.tri(para)
```

Arguments

`para`	A parameter `list` returned from `partri` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `tri`.
`FALSE`	If the `type` is not `tri`.

Author(s)

W.H. Asquith

See Also

partri

Examples

```
para <- partri(lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52)))
if(is.tri(para) == TRUE) {
  Q <- quatri(0.5,para)
}
```

---

is.wak *Is a Distribution Parameter Object Typed as Wakeby*

Description

The distribution parameter object returned by functions of **lmomco** such as by `parwak` are typed by an attribute `type`. This function checks that `type` is `wak` for the Wakeby distribution.

Usage

```
is.wak(para)
```

Arguments

`para`	A parameter `list` returned from `parwak` or `vec2par`.

Value

`TRUE`	If the `type` attribute is `wak`.
`FALSE`	If the `type` is not `wak`.

Author(s)

W.H. Asquith

See Also

`parwak`

Examples

```
para <- parwak(lmoms(c(123,34,4,654,37,78)))
if(is.wak(para) == TRUE) {
  Q <- quawak(0.5,para)
}
```

is.wei *Is a Distribution Parameter Object Typed as Weibull*

Description

The distribution parameter object returned by functions of **lmomco** such as by parwei are typed by an attribute type. This function checks that type is wei for the Weibull distribution.

Usage

```
is.wei(para)
```

Arguments

para	A parameter list returned from parwei or vec2par.

Value

TRUE	If the type attribute is wei.
FALSE	If the type is not wei.

Author(s)

W.H. Asquith

See Also

parwei

Examples

```
para <- parwei(lmoms(c(123,34,4,654,37,78)))
if(is.wei(para) == TRUE) {
  Q <- quawei(0.5,para)
}
```

LaguerreHalf *Laguerre Polynomial (Half)*

Description

This function computes the Laguerre polynomial, which is useful in applications involving the variance of the Rice distribution (see parrice). The Laguerre polynomial is

$$L_{1/2}(x) = \exp^{x/2} \times [(1-x)I_0(-x/2) - xI_1(-x/2)],$$

where the modified Bessel function of the first kind is $I_k(x)$, which has an R implementation in besselI, and for strictly integer k is defined as

$$I_k(x) = \frac{1}{\pi}\int_0^{\pi} \exp(x\cos(\theta))\cos(k\theta)\,\mathrm{d}\theta.$$

Usage

```
LaguerreHalf(x)
```

Arguments

x	A value.

Value

The value for the Laguerre polynomial is returned.

Author(s)

W.H. Asquith

See Also

pdfrice

Examples

```
LaguerreHalf(-100^2/(2*10^2))
```

`Lcomoment.coefficients` *L-comoment Coefficient Matrix*

Description

Compute the L-comoment coefficients from an L-comoment matrix of order $k \geq 2$ and the $k = 2$ (2nd order) L-comoment matrix. However, if the first argument is 1st-order then the coefficients of L-covariation are computed. The function requires that each matrix has already been computed by the function `Lcomoment.matrix`.

Usage

```
Lcomoment.coefficients(Lk, L2)
```

Arguments

`Lk`	A $k \geq 2$ L-comoment matrix from `Lcomoment.matrix`.
`L2`	A $k = 2$ L-comoment matrix from `Lcomoment.matrix(Dataframe,k=2)`.

Details

The coefficient of L-variation is computed by `Lcomoment.coefficients(L1,L2)` where `L1` is a 1st-order L-moment matrix and `L2` is a $k = 2$ L-comoment matrix. Symbolically, the coefficient of L-covariation is

$$\hat{\tau}_{[12]} = \frac{\hat{\lambda}_{2[12]}}{\hat{\lambda}_{1[12]}}.$$

The higher L-comoment coefficients (L-coskew, L-cokurtosis, ...) are computed by the function `Lcomoment.coefficients(L3,L2)` ($k = 3$), `Lcomoment.coefficients(L4,L2)` ($k = 4$), and so on. Symbolically, the higher L-comoment coefficients for $k \geq 3$ are

$$\hat{\tau}_{k[12]} = \frac{\hat{\lambda}_{k[12]}}{\hat{\lambda}_{2[12]}}.$$

Finally, the usual univariate L-moment ratios as seen from `lmom.ub` or `lmoms` are along the diagonal. The `Lcomoment.coefficients` function does not make use of `lmom.ub` or `lmoms`.

Value

An R `list` is returned.

`type`	The type of L-comoment representation in the matrix: “Lcomoment.coefficients”.
`order`	The order of the coefficients. `order=2` L-covariation, `order=3` L-coskew, ...
`matrix`	A $k \geq 2$ L-comoment coefficient matrix.

Note

The function begins with a capital letter. This is intentionally done so that lower case namespace is preserved. By using a capital letter now, then `lcomoment.coefficients` remains an available name in future releases.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Serfling, R., and Xiao, P., 2007, A contribution to multivariate L-moments—L-comoment matrices: Journal of Multivariate Analysis, v. 98, pp. 1765–1781.

See Also

`Lcomoment.matrix`, `Lcomoment.coefficients`

Examples

```
D      <- data.frame(X1=rnorm(30), X2=rnorm(30), X3=rnorm(30))
L1     <- Lcomoment.matrix(D,k=1)
L2     <- Lcomoment.matrix(D,k=2)
L3     <- Lcomoment.matrix(D,k=3)
LkLCV  <- Lcomoment.coefficients(L1,L2)
LkTAU3 <- Lcomoment.coefficients(L3,L2)
```

Lcomoment.correlation *L-correlation Matrix (L-correlation through Sample L-comoments)*

Description

Compute the L-correlation from an L-comoment matrix of order $k = 2$. This function assumes that the 2nd order matrix is already computed by the function `Lcomoment.matrix`.

Usage

```
Lcomoment.correlation(L2)
```

Arguments

L2	A $k = 2$ L-comoment matrix from `Lcomoment.matrix(Dataframe,k=2)`.

Details

L-correlation is computed by `Lcomoment.coefficients(L2,L2)` where `L2` is second order L-comoment matrix. The usual L-scale values as seen from `lmom.ub` or `lmoms` are along the diagonal. This function does not make use of `lmom.ub` or `lmoms` and can be used to verify computation of τ (coefficient of L-variation).

Value

An R `list` is returned.

`type`	The type of L-comoment representation in the matrix: "Lcomoment.coefficients".
`order`	The order of the matrix—extracted from the first matrix in arguments.
`matrix`	A $k \geq 2$ L-comoment coefficient matrix.

Note

The function begins with a capital letter. This is intentionally done so that lower case namespace is preserved. By using a capital letter now, then `lcomoment.correlation` remains an available name in future releases.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Serfling, R., and Xiao, P., 2007, A contribution to multivariate L-moments—L-comoment matrices: Journal of Multivariate Analysis, v. 98, pp. 1765–1781.

See Also

`Lcomoment.matrix`, `Lcomoment.correlation`

Examples

```
D   <- data.frame(X1=rnorm(30), X2=rnorm(30), X3=rnorm(30))
L2  <- Lcomoment.matrix(D,k=2)
RHO <- Lcomoment.correlation(L2)
## Not run:
"SerfXiao.eq17" <-
 function(n=25, A=10, B=2, k=4,
          method=c("pearson","lcorr"), wrt=c("12", "21")) {
   method <- match.arg(method); wrt <- match.arg(wrt)
   # X1 is a linear regression on X2
   X2 <- rnorm(n); X1 <- A + B*X2 + rnorm(n)
   r12p <- cor(X1,X2) # Pearson's product moment correlation
   XX <- data.frame(X1=X1, X2=X2) # for the L-comoments
   T2 <- Lcomoment.correlation(Lcomoment.matrix(XX, k=2))$matrix
```

```
    LAMk <- Lcomoment.matrix(XX, k=k)$matrix # L-comoments of order k
    if(wrt == "12") { # is X2 the sorted variable?
       lmr <- lmoms(X1, nmom=k); Lamk <- LAMk[1,2]; Lcor <- T2[1,2]
    } else {          # no X1 is the sorted variable (21)
       lmr <- lmoms(X2, nmom=k); Lamk <- LAMk[2,1]; Lcor <- T2[2,1]
    }
    # Serfling and Xiao (2007, eq. 17) state that
    # L-comoment_k[12] = corr.coeff * Lmoment_k[1] or
    # L-comoment_k[21] = corr.coeff * Lmoment_k[2]
    # And with the X1, X2 setup above, Pearson corr. == L-corr.
    # There will be some numerical differences for any given sample.
    ifelse(method == "pearson",
              return(lmr$lambdas[k]*r12p - Lamk),
              return(lmr$lambdas[k]*Lcor - Lamk))
    # If the above returns a expected value near zero then, their eq.
    # is numerically shown to be correct and the estimators are unbiased.
}

# The means should be near zero.
nrep <- 2000; seed <- rnorm(1); set.seed(seed)
mean(replicate(n=nrep, SerfXiao.eq17(method="pearson", k=4)))
set.seed(seed)
mean(replicate(n=nrep, SerfXiao.eq17(method="lcorr", k=4)))
# The variances should nearly be equal.
seed <- rnorm(1); set.seed(seed)
var(replicate(n=nrep, SerfXiao.eq17(method="pearson", k=6)))
set.seed(seed)
var(replicate(n=nrep, SerfXiao.eq17(method="lcorr", k=6)))

## End(Not run)
```

`Lcomoment.Lk12` *Compute a Single Sample L-comoment*

Description

Compute the L-comoment ($\lambda_{k[12]}$) for a given pair of sample of n random variates $\{(X^{(1)}_i, X^{(1)}_i), 1 \leq i \leq n\}$ from a joint distribution $H(x^{(1)}, x^{(2)})$ with marginal distribution functions F_1 and F_2. When the $X^{(2)}$ are sorted to form the sample order statistics $X^{(2)}_{1:n} \leq X^{(2)}_{2:n} \leq \cdots \leq X^{(2)}_{n:n}$, then the element of $X^{(1)}$ of the unordered (at leasted expected to be) but shuffled set $\{X^{(1)}_1, \ldots, X^{(1)}_n\}$ that is paired with $X^{(2)}_{r:n}$ the *concomitant* $X^{(12)}_{[r:n]}$ of $X^{(2)}_{r:n}$. (The shuffling occurs by the sorting of $X^{(2)}$.) The $k \geq 1$-order L-comoments are defined (Serfling and Xiao, 2007, eq. 26) as

$$\hat{\lambda}_{k[12]} = \frac{1}{n}\sum_{r=1}^{n} w^{(k)}_{r:n} X^{(12)}_{[r:n]},$$

where $w^{(k)}_{r:n}$ is defined under `Lcomoment.Wk`. (The author is aware that $k \geq 1$ is $k \geq 2$ in Serfling and Xiao (2007) but $k = 1$ returns sample means. This matters only in that the **lmomco** package returns matrices for $k \geq 1$ by `Lcomoment.matrix` even though the off diagnonals are `NA`s.)

Usage

```
Lcomoment.Lk12(X1,X2,k=1)
```

Arguments

X1	A vector of random variables (a sample of random variable 1).
X2	Another vector of random variables (a sample of random variable 2).
k	The order of the L-comoment to compute. The default is 1.

Details

Now directing explanation of L-comoments with some reference heading into R code. L-comoments of random variable `X1` (a vector) are computed from the concomitants of `X2` (another vector). That is, $X2$ is sorted in ascending order to create the order statistics of `X2`. During the sorting process, `X1` is reshuffled to the order of `X2` to form the concomitants of `X2` (denoted as `X12`). So the trailing `2` is the sorted variable and the leading `1` is the variable that is shuffled. The `X12` in turn are used in a weighted summation and expectation calculation to compute the L-comoment of `X1` with respect to `X2` such as by `Lk3.12 <- Lcomoment.Lk12(X1,X2,k=3)`. The notation of `Lk12` is to read "Lambda for kth order L-comoment", where the `12` portion of the notation reflects that of Serfling and Xiao (2007) and then Asquith (2011). The weights for the computation are derived from calls made by `Lcomoment.Lk12` to the weight function `Lcomoment.Wk`. The L-comoments of `X2` are computed from the concomitants of `X1`, and the `X21` are formed by sorting `X1` in ascending order and in turn shuffling `X2` by the order of `X1`. The often asymmetrical L-comoment of `X2` with respect to `X1` is readily done (`Lk3.21 <- Lcomoment.Lk12(X2,X1,k=3)`) and is not necessarily equal to (`Lk3.12 <- Lcomoment.Lk12(X1,X2,k=3)`).

Value

A single L-comoment.

Note

The function begins with a capital letter. This is intentionally done so that lower case namespace is preserved. By using a capital letter now, then `lcomoment.Lk12` or similar remains an available name in future releases.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Serfling, R., and Xiao, P., 2007, A contribution to multivariate L-moments—L-comoment matrices: Journal of Multivariate Analysis, v. 98, pp. 1765–1781.

See Also

Lcomoment.matrix, Lcomoment.Wk

Examples

```
X1 <- rnorm(101); X2 <- rnorm(101) + X1
Lcoskew12 <- Lcomoment.Lk12(X1,X2, k=3)
Lcorr12 <- Lcomoment.Lk12(X1,X2,k=2)/Lcomoment.Lk12(X1,X1,k=2)
rhop12 <- cor(X1, X2, method="pearson")
print(Lcorr12 - rhop12) # smallish number
```

Lcomoment.matrix *Compute Sample L-comoment Matrix*

Description

Compute the L-comoments from a rectangular data.frame containing arrays of random variables. The order of the L-comoments is specified.

Usage

```
Lcomoment.matrix(DATAFRAME, k=1)
```

Arguments

DATAFRAME	A convential data.frame that is rectangular.
k	The order of the L-comoments to compute. Default is $k = 1$.

Details

L-comoments are computed for each item in the data.frame. L-comoments of order $k = 1$ are means and co-means. L-comoments of order $k = 2$ are L-scale and L-coscale values. L-comoments of order $k = 3$ are L-skew and L-coskews. L-comoments of order $k = 4$ are L-kurtosis and L-cokurtosis, and so on. The usual univariate L-moments of order k as seen from lmom.ub or lmoms are along the diagonal. This function does not make use of lmom.ub or lmoms. The function Lcomoment.matrix calls Lcomoment.Lk12 for each cell in the matrix. The L-comoment matrix for d-random variables is

$$\mathbf{\Lambda}_k = (\hat{\lambda}_{k[ij]})$$

computed over the pairs $(X^{(i)}, X^{(j)})$ where $1 \leq i \leq j \leq d$.

Value

An R list is returned.

type	The type of L-comoment representation in the matrix: “Lcomoments”.
order	The order of the matrix—specified by k in the argument list.
matrix	A kth order L-comoment matrix.

Note

The function begins with a capital letter. This is intentionally done so that lower case namespace is preserved. By using a capital letter now, then `lcomoment.matrix` remains an available name in future releases.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Serfling, R., and Xiao, P., 2007, A contribution to multivariate L-moments—L-comoment matrices: Journal of Multivariate Analysis, v. 98, pp. 1765–1781.

See Also

`Lcomoment.Lk12`, `Lcomoment.coefficients`

Examples

```
D  <- data.frame(X1=rnorm(30), X2=rnorm(30), X3=rnorm(30))
L1 <- Lcomoment.matrix(D, k=1)
L2 <- Lcomoment.matrix(D, k=2)
```

`Lcomoment.Wk` *Weighting Coefficient for Sample L-comoment*

Description

Compute the weight factors for computation of an L-comoment for order `k`, order statistic `r`, and sample size `n`.

Usage

```
Lcomoment.Wk(k,r,n)
```

Arguments

`k`	Order of L-comoment being computed by parent calls to `Lcomoment.Wk`.
`r`	Order statistic index involved.
`n`	Sample size.

Details

This function computes the weight factors needed to calculation L-comoments and is interfaced or used by `Lcomoment.Lk12`. The weight factors are

$$w^{(k)}_{r:n} = \sum_{j=0}^{min\{r-1,k-1\}} (-1)^{k-1-j} \frac{\binom{k-1}{j}\binom{k-1+j}{j}\binom{r-1}{j}}{\binom{n-1}{j}}.$$

The weight factor $w^{(k)}_{r:n}$ is the discrete Legendre polynomial. The weight factors are well illustrated in figure 6.1 of Asquith (2011). This function is not intended for end users.

Value

A single L-comoment weight factor.

Note

The function begins with a capital letter. This is intentionally done so that lower case namespace is preserved. By using a capital letter now, then `lcomoment.Wk` remains an available name in future releases.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Serfling, R., and Xiao, P., 2007, A contribution to multivariate L-moments—L-comoment matrices: Journal of Multivariate Analysis, v. 98, pp. 1765–1781.

See Also

`Lcomoment.Lk12`

Examples

```
Wk <- Lcomoment.Wk(2,3,5)
print(Wk)

## Not run:
# To compute the weight factors for L-skew and L-coskew (k=3) computation
# for a sample of size 20.
Wk <- matrix(nrow=20,ncol=1)
for(r in seq(1,20)) Wk[r] <- Lcomoment.Wk(3,r,20)
plot(seq(1,20),Wk, type="b")

## End(Not run)
```

```
# The following shows the actual weights used for computation of
# the first four L-moments. The sum of the each sample times the
# corresponding weight equals the L-moment.
fakedat <- sort(c(-10, 20, 30, 40));  n <- length(fakedat)
Wk1 <- Wk2 <- Wk3 <- Wk4 <- vector(mode="numeric", length=n);
for(i in 1:n) {
   Wk1[i] <- Lcomoment.Wk(1,i,n)/n
   Wk2[i] <- Lcomoment.Wk(2,i,n)/n
   Wk3[i] <- Lcomoment.Wk(3,i,n)/n
   Wk4[i] <- Lcomoment.Wk(4,i,n)/n
}
cat(c("Weights for mean",         round(Wk1, digits=4), "\n"))
cat(c("Weights for L-scale",      round(Wk2, digits=4), "\n"))
cat(c("Weights for 3rd L-moment", round(Wk3, digits=4), "\n"))
cat(c("Weights for 4th L-moment", round(Wk4, digits=4), "\n"))
my.lams <- c(sum(fakedat*Wk1), sum(fakedat*Wk2),
             sum(fakedat*Wk3), sum(fakedat*Wk4))
cat(c("Manual L-moments:", my.lams, "\n"))
cat(c("lmomco L-moments:", lmoms(fakedat, nmom=4)$lambdas,"\n"))
# The last two lines of output should be the same---note that lmoms()
# does not utilize Lcomoment.Wk(). So a double check is made.
```

`lcomoms2` *The Sample L-comoments for Two Variables*

Description

Compute the sample L-moments for the R two variable `data.frame`. The “2” in the function name is to refer to fact that this function operates on only two variables. The length of the variables must be greater than the number of L-comoments requested.

Usage

```
lcomoms2(DATAFRAME, nmom=3, asdiag=FALSE, opdiag=FALSE, ...)
```

Arguments

`DATAFRAME`	An R `data.frame` housing column vectors of data values.
`nmom`	The number of L-comoments to compute. Default is 3.
`asdiag`	Return the `diagonal` of the matrices. Default is `FALSE`.
`opdiag`	Return the opposing diagonal of the matrices. Default is `FALSE`. This function returns the opposing diagonal from first two to second.
`...`	Additional arguments to pass.

Value

An R `list` is returned of the first

- `L1` — Matrix or diagonals of first L-comoment.
- `L2` — Matrix or diagonals of second L-comoment.
- `T2` — Matrix or diagonals of L-comoment correlation.
- `T3` — Matrix or diagonals of L-comoment skew.
- `T4` — Matrix or diagonals of L-comoment kurtosis.
- `T5` — Matrix or diagonals of L-comoment Tau5.
- `source` — An attribute identifying the computational source of the L-comoments: "lcomoms2".

Note

This function computes the L-comoments through the generalization of the `Lcomoment.matrix` and `Lcomoment.coefficients` functions.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`Lcomoment.matrix` and `Lcomoment.coefficients`

Examples

```
## Not run:
# Random simulation of standard normal and then combine with
# a random standard exponential distribution
X <- rnorm(200); Y <- X + rexp(200)
z <- lcomoms2(data.frame(X=X, Y=Y))
print(z)

z <- lcomoms2(data.frame(X=X, Y=Y), diag=TRUE)
print(z$T3) # the L-skew values of the margins

z <- lcomoms2(data.frame(X=X, Y=Y), opdiag=TRUE)
print(z$T3) # the L-coskew values
## End(Not run)
```

`lkhlmomco` *Leimkuhler Curve of the Distributions*

Description

This function computes the Leimkuhler Curve for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair et al. (2013, p. 181) as

$$K(u) = 1 - \frac{1}{\mu}\int_0^{1-u} x(p)\; \mathrm{d}p,$$

where $K(u)$ is Leimkuhler curve for nonexceedance probability u. The Leimkuhler curve is related to the Lorenz curve ($L(u)$, lrzlmomco) by

$$K(u) = 1 - L(1-u),$$

and related to the reversed residual mean quantile function ($R(u)$, rrmlmomco) and conditional mean (μ, cmlmomco) for $u = 0$ by

$$K(u) = \frac{1}{\mu}[\mu - (1-u)(x(1-u) - R(1-u))].$$

Usage

```
lkhlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.

Value

Leimkuhler curve value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, lrzlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0

"afunc" <- function(u) { return(par2qua(u,A,paracheck=FALSE)) }
f <- 0.35 # All three computations report: Ku = 0.6413727
Ku1 <- 1 - 1/cmlmomco(f=0,A) * integrate(afunc,0,1-f)$value
Ku2 <- (cmlmomco(0,A) - (1-f)*(quagov(1-f,A) - rrmlmomco(1-f,A)))/cmlmomco(0,A)
Ku3 <- lkhlmomco(f, A)
```

---

lmom.ub *Unbiased Sample L-moments by Direct Sample Estimators*

Description

Unbiased sample L-moments are computed for a vector using the direct sample estimation method as opposed to the use of sample probability-weighted moments. The L-moments are the ordinary L-moments and not the trimmed L-moments (see TLmoms). The mean, L-scale, coefficient of L-variation (τ, LCV, L-scale/mean), L-skew (τ_3, TAU3, L3/L2), L-kurtosis (τ_4, TAU4, L4/L2), and τ_5 (TAU5, L5/L2) are computed. In conventional nomenclature, the L-moments are

$$\hat{\lambda}_1 = \mathrm{L1} = \text{mean},$$

$$\hat{\lambda}_2 = \mathrm{L2} = \text{L-scale},$$

$$\hat{\lambda}_3 = \mathrm{L3} = \text{third L-moment},$$

$$\hat{\lambda}_4 = \mathrm{L4} = \text{fourth L-moment, and}$$

$$\hat{\lambda}_5 = \mathrm{L5} = \text{fifth L-moment.}$$

The L-moment ratios are

$$\hat{\tau} = \mathrm{LCV} = \lambda_2/\lambda_1 = \text{coefficient of L-variation},$$

$$\hat{\tau}_3 = \mathrm{TAU3} = \lambda_3/\lambda_2 = \text{L-skew},$$

$$\hat{\tau}_4 = \mathrm{TAU4} = \lambda_4/\lambda_2 = \text{L-kurtosis, and}$$

$$\hat{\tau}_5 = \mathrm{TAU5} = \lambda_5/\lambda_2 = \text{not named.}$$

It is common amongst practitioners to lump the L-moment ratios into the general term “L-moments” and remain inclusive of the L-moment ratios. For example, L-skew then is referred to as the 3rd L-moment when it technically is the 3rd L-moment ratio. The first L-moment ratio has no definition; the lmoms function uses the NA of R in its vector representation of the ratios.

The mathematical expression for sample L-moment computation is shown under TLmoms. The formula jointly handles sample L-moment computation and sample TL-moment computation.

Usage

```
lmom.ub(x)
```

Arguments

x	A vector of data values.

Details

The L-moment ratios (τ, τ_3, τ_4, and τ_5) are the primary higher L-moments for application, such as for distribution parameter estimation. However, the actual L-moments (λ_3, λ_4, and λ_5) are also reported. The implementation of `lmom.ub` requires a minimum of five data points. If more or fewer L-moments are needed then use the function `lmoms`.

Value

An R `list` is returned.

L1	Arithmetic mean.
L2	L-scale—analogous to standard deviation (see also `gini.mean.diff`.
LCV	coefficient of L-variation—analogous to coe. of variation.
TAU3	The third L-moment ratio or L-skew—analogous to skew.
TAU4	The fourth L-moment ratio or L-kurtosis—analogous to kurtosis.
TAU5	The fifth L-moment ratio.
L3	The third L-moment.
L4	The fourth L-moment.
L5	The fifth L-moment.
source	An attribute identifying the computational source of the L-moments: “lmom.ub”.

Note

The `lmom.ub` function was among the first functions written for **lmomco** and actually written before **lmomco** was initiated. The ub was to be contrasted with plotting-position-based estimation methods: `pwm.pp` $\rightarrow$ `pwm2lmom`. Further, at the time of development the radical expansion of **lmomco** beyond the Hosking (1996) FORTRAN libraries was not anticipated. The author now exclusively uses `lmoms` but the numerical results should be identical. The direct sample estimator algorithm by Wang (1996) is used in `lmom.ub` and a more generalized algorithm is associated with `lmoms`.

Author(s)

W.H. Asquith

Source

The Perl code base of W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Wang, Q.J., 1996, Direct sample estimators of L-moments: Water Resources Research, v. 32, no. 12., pp. 3617–3619.

See Also

lmom2pwm, pwm.ub, pwm2lmom, lmoms, lmorph

Examples

```
lmr <- lmom.ub(c(123,34,4,654,37,78))
lmorph(lmr)
lmom.ub(rnorm(100))
```

---

lmom2par — *Convert L-moments to the Parameters of a Distribution*

Description

This function converts L-moments to the parameters of a distribution. The type of distribution is specified in the argument list: aep4, cau, emu, exp, gam, gep, gev, gld, glo, gno, gov, gpa, gum, kap, kmu, kur, lap, lmrq, ln3, nor, pe3, ray, revgum, rice, sla, st3, texp, wak, or wei.

Usage

```
lmom2par(lmom, type, ...)
lmr2par(x, type, ...)
```

Arguments

lmom	An L-moment object such as that returned by lmoms or pwm2lmom.
type	Three character (minimum) distribution type (for example, type="gev").
...	Additional arguments for the parCCC functions.
x	In the lmr2par call the L-moments are computed from the x values. This function is a parallel to mle2par and mps2par.

Value

An R list is returned. This list should contain at least the following items, but some distributions such as the revgum have extra.

type	The type of distribution in three character (minimum) format.
para	The parameters of the distribution.
source	Attribute specifying source of the parameters.

Author(s)

W.H. Asquith

See Also

par2lmom

Examples

```
lmr  <- lmoms(rnorm(20))
para <- lmom2par(lmr,type="nor")

# The lmom2par() calls will error if trim != 1.
X <- rcauchy(20)
cauchy <- lmom2par(TLmoms(X, trim=1), type="cau")
slash  <- lmom2par(TLmoms(X, trim=1), type="sla")
## Not run:
plot(pp(X), sort(X), xlab="PROBABILITY", ylab="CAUCHY")
lines(nonexceeds(), par2qua(nonexceeds(), cauchy))
lines(nonexceeds(), par2qua(nonexceeds(), slash), col=2)

## End(Not run)
```

---

lmom2pwm *L-moments to Probability-Weighted Moments*

Description

Converts the L-moments to the probability-weighted moments (PWMs) given the L-moments. The conversion is linear so procedures based on L-moments are identical to those based on PWMs. The expression linking PWMs to L-moments is

$$\lambda_{r+1} = \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} \binom{r+k}{k} \beta_k,$$

where λ_{r+1} are the L-moments, β_r are the PWMs, and $r \geq 0$.

Usage

```
lmom2pwm(lmom)
```

Arguments

`lmom`	An L-moment object created by lmoms, lmom.ub, or vec2lmom. The function also supports `lmom` as a vector of L-moments (λ_1, λ_2, τ_3, τ_4, and τ_5).

Details

PWMs are linear combinations of the L-moments and therefore contain the same statistical information of the data as the L-moments. However, the PWMs are harder to interpret as measures of probability distributions. The PWMs are included in **lmomco** for theoretical completeness and are not intended for use with the majority of the other functions implementing the various probability distributions. The relations between L-moments (λ_r) and PWMs (β_{r-1}) for $1 \le r \le 5$ order are

$$\lambda_1 = \beta_0,$$

$$\lambda_2 = 2\beta_1 - \beta_0,$$

$$\lambda_3 = 6\beta_2 - 6\beta_1 + \beta_0,$$

$$\lambda_4 = 20\beta_3 - 30\beta_2 + 12\beta_1 - \beta_0, \text{ and}$$

$$\lambda_5 = 70\beta_4 - 140\beta_3 + 90\beta_2 - 20\beta_1 + \beta_0.$$

The linearity between L-moments and PWMs means that procedures based on one are equivalent to the other. This function only accomodates the first five L-moments and PWMs. Therefore, at least five L-moments are required in the passed argument.

Value

An R `list` is returned.

`betas`	The PWMs. Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
`source`	Source of the PWMs: “pwm”.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

`lmom.ub`, `lmoms`, `pwm.ub`, `pwm2lmom`

Examples

```
pwm <- lmom2pwm(lmoms(c(123,34,4,654,37,78)))
lmom2pwm(lmom.ub(rnorm(100)))
lmom2pwm(lmoms(rnorm(100)))

lmomvec1 <- c(1000,1300,0.4,0.3,0.2,0.1)
pwmvec   <- lmom2pwm(lmomvec1)
print(pwmvec)
#$betas
#[1] 1000.0000 1150.0000 1070.0000  984.5000  911.2857
#
#$source
#[1] "lmom2pwm"

lmomvec2 <- pwm2lmom(pwmvec)
print(lmomvec2)
#$lambdas
#[1] 1000 1300  520  390  260
#
#$ratios
#[1]  NA 1.3 0.4 0.3 0.2
#
#$source
#[1] "pwm2lmom"

pwm2lmom(lmom2pwm(list(L1=25, L2=20, TAU3=.45, TAU4=0.2, TAU5=0.1)))
```

---

lmom2vec — *Convert an L-moment object to a Vector of L-moments*

Description

This function converts an L-moment object in the structure used by **lmomco** into a simple vector. The precise operation of this function is dependent on the L-moment object argument. The `lmorph` function is not used. This function is useful if one needs to use certain functions in the **lmoms** package that are built around vectors of L-moments and L-moment ratios as arguments.

Usage

```
lmom2vec(lmom, ...)
```

Arguments

`lmom`	L-moment object as from functions such as `lmoms`, `lmom.ub`, and `vec2lmom`.
`...`	Not presently used.

Value

A vector of the L-moments (λ_1, λ_2, τ_3, τ_4, τ_5, $\dots$, τ_r).

Author(s)

W.H. Asquith

See Also

lmom.ub, lmoms, lmorph, vec2lmom, pwm2vec

Examples

```
lmr <- lmoms(rnorm(40))
lmom2vec(lmr)
lmr <- vec2lmom(c(140,150,.3,.2,-.1))
lmom2vec(lmr)
```

---

lmomaep4 — *L-moments of the 4-Parameter Asymmetric Exponential Power Distribution*

Description

This function computes the L-moments of the 4-parameter Asymmetric Exponential Power distribution given the parameters (ξ, α, κ, and h) from paraep4. The first four L-moments are complex. The mean λ_1 is

$$\lambda_1 = \xi + \alpha(1/\kappa - \kappa)\frac{\Gamma(2/h)}{\Gamma(1/h)},$$

where $\Gamma(x)$ is the complete gamma function or `gamma()` in R.

The L-scale λ_2 is

$$\lambda_2 = -\frac{\alpha\kappa(1/\kappa-\kappa)^2\Gamma(2/h)}{(1+\kappa^2)\Gamma(1/h)} + 2\frac{\alpha\kappa^2(1/\kappa^3+\kappa^3)\Gamma(2/h)I_{1/2}(1/h, 2/h)}{(1+\kappa^2)^2\Gamma(1/h)},$$

where $I_{1/2}(1/h, 2/h)$ is the cumulative distribution function of the Beta distribution ($I_x(a, b)$) or `pbeta(1/2, shape1=1/h, shape2=2/h)` in R. This function is also referred to as the normalized incomplete beta function (Delicado and Goria, 2008) and defined as

$$I_x(a, b) = \frac{\int_0^x t^{a-1}(1-t)^{b-1}\, \mathrm{d}t}{\beta(a, b)},$$

where $\beta(1/h, 2/h)$ is the complete beta function or `beta(1/h, 2/h)` in R.

The third L-moment λ_3 is

$$\lambda_3 = A_1 + A_2 + A_3,$$

where the A_i are

$$A_1 = \frac{\alpha(1/\kappa-\kappa)(\kappa^4-4\kappa^2+1)\Gamma(2/h)}{(1+\kappa^2)^2\Gamma(1/h)},$$

$$A_2 = -6\frac{\alpha\kappa^3(1/\kappa-\kappa)(1/\kappa^3+\kappa^3)\Gamma(2/h)I_{1/2}(1/h, 2/h)}{(1+\kappa^2)^3\Gamma(1/h)},$$

$$A_3 = 6\frac{\alpha(1+\kappa^4)(1/\kappa-\kappa)\Gamma(2/h)\Delta}{(1+\kappa^2)^2\Gamma(1/h)},$$

and where Δ is

$$\Delta = \frac{1}{\beta(1/h,2/h)}\int_0^{1/2} t^{1/h-1}(1-t)^{2/h-1} I_{(1-t)/(2-t)}(1/h,3/h)\;\mathrm{d}t.$$

The fourth L-moment λ_4 is

$$\lambda_4 = B_1 + B_2 + B_3 + B_4,$$

where the B_i are

$$B_1 = -\frac{\alpha\kappa(1/\kappa-\kappa)^2(\kappa^4-8\kappa^2+1)\Gamma(2/h)}{(1+\kappa^2)^3\Gamma(1/h)},$$

$$B_2 = 12\frac{\alpha\kappa^2(\kappa^3+1/\kappa^3)(\kappa^4-3\kappa^2+1)\Gamma(2/h)I_{1/2}(1/h,2/h)}{(1+\kappa^2)^4\Gamma(1/h)},$$

$$B_3 = -30\frac{\alpha\kappa^3(1/\kappa-\kappa)^2(1/\kappa^2+\kappa^2)\Gamma(2/h)\Delta}{(1+\kappa^2)^3\Gamma(1/h)},$$

$$B_4 = 20\frac{\alpha\kappa^4(1/\kappa^5+\kappa^5)\Gamma(2/h)\Delta_1}{(1+\kappa^2)^4\Gamma(1/h)},$$

and where Δ_1 is

$$\Delta_1 = \frac{\int_0^{1/2}\int_0^{(1-y)/(2-y)} y^{1/h-1}(1-y)^{2/h-1}z^{1/h-1}(1-z)^{3/h-1}\;I'\;\mathrm{d}z\,\mathrm{d}y}{\beta(1/h,2/h)\beta(1/h,3/h)},$$

for which $I' = I_{(1-z)(1-y)/(1+(1-z)(1-y))}(1/h,2/h)$ is the cumulative distribution function of the beta distribution ($I_x(a,b)$) or `pbeta((1-z)(1-y)/(1+(1-z)(1-y)), shape1=1/h, shape2=2/h)` in R. Finally, if the τ_3 of the distribution is zero (symmetrical), then the distribution is known as the Exponential Power (see `lmrdia46`).

Usage

```
lmomaep4(para, paracheck=TRUE, t3t4only=FALSE)
```

Arguments

`para`	The parameters of the distribution.
`paracheck`	Should the parameters be checked for validity by the `are.paraep4.valid` function.
`t3t4only`	Return only the τ_3 and τ_4 for the parameters κ and h. The λ_1 and λ_2 are not explicitly used although numerical values for these two L-moments are required only to avoid computational errors. Care is made so that the α parameter that is in numerator of $\lambda_{2,3,4}$ is not used in the computation of τ_3 and τ_4. Hence, this option permits the computation of τ_3 and τ_4 when α is unknown. This features is largely available for research and development purposes. Mostly this feature was used for the $\{\tau_3, \tau_4\}$ trajectory for `lmrdia`

.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomaep4".

or an alternative R list is returned if `t3t4only=TRUE`

`T3`	L-skew, τ_3.
`T4`	L-kurtosis, τ_4.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Delicado, P., and Goria, M.N., 2008, A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution: Computational Statistics and Data Analysis, v. 52, no. 3, pp. 1661–1673.

See Also

`paraep4`, `cdfaep4`, `pdfaep4`, `quaaep4`

Examples

```
## Not run:
para <- vec2par(c(0, 1, 0.5, 4), type="aep4")
lmomaep4(para)

## End(Not run)
```

lmomben *L-moments of the Benford Distribution*

Description

Experimental—This function returns previously numerical estimations of the L-moments of the Benford distribution (Benford's Law) given parameters defining the number of first M-significant digits and the numeric base.

For the first significant digits ($d \in 1, \cdots, 9$) (base 10) (designate as $m = 1$), the L-moments were estimated through very large sample-size simulation and sample L-moments computed (lmoms), direct numerical integration (theoLmoms), and through numerical integration of the probability weighted moments and conversion to L-moments (pwm2lmom) as

$$\lambda_1 = 3.43908699617500524,$$

$$\lambda_2 = 1.34518434179517077,$$

$$\tau_3 = 0.24794090889493661, \text{ and}$$

$$\tau_4 = 0.01614509742647182.$$

For the first two-significant digits ($d \in 10, \cdots, 99$) (base 10) (designate as $m = 2$), the L-moments were estimated through very large sample-size simulation, direct numerical integration (theoLmoms), and through numerical integration of the probability weighted moments and conversion to L-moments (pwm2lmom) as

$$\lambda_1 = 38.59062918136093145,$$

$$\lambda_2 = 13.81767809210059283,$$

$$\tau_3 = 0.22237541787527126, \text{ and}$$

$$\tau_4 = 0.03541037418894027.$$

For the first three-significant digits ($d \in 100, \cdots, 999$) (base 10) (designate as $m = 3$), the L-moments were estimated through very large sample-size simulation, direct numerical integration (theoLmoms), and through numerical integration of the probability weighted moments and conversion to L-moments (pwm2lmom) as

$$\lambda_1 = 390.36783537821605705,$$

$$\lambda_2 = 138.21917489739223583,$$

$$\tau_3 = 0.22192482374529940, \text{ and}$$

$$\tau_4 = 0.03571514686148788.$$

Source of the L-moments—The script `inst/doc/benford/compLmomsBenford.R` in the **lmomco** package sources is the authoritative source of the computation of the L-moments shown. Three methods are used, and the arithmetic average of the three provides the L-moments: (1) Probability-weighted simulation of the probability mass function (PMF) is used in very large sample size and sample L-moments computed by lmoms, (2) direct numerical integration for the theoretical L-moments of the quantile function (quaben) of the distribution that itself is from the cumulative distribution function (cdfben) that itself is from the PMF (pmfben), and (3) direct numerical integration of the probability-weighted moments of the quantile function (quaben) and subsequent linear system of equations to compute the L-moments. Each of the aforementioned methods result in numerical differences say at about the fourth decimal. (No previous description of the L-moments of the Benford distribution appear extant in the literature in July 2024.)

Usage

```
lmomben(para=list(para=c(1, 10)), ...)
```

Arguments

`para`	The number of first M-significant digits followed by the numerical base (only base10 supported) and the list structure mimics similar uses of the **lmomco** list structure. Default are the first significant digits and hence the digits 1 through 9.
`...`	Additional arguments to pass (not likely to be needed but changes in base handling might need this).

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomben".

Note

Hypothesis Testing—Let the squared Euclidean distance of the L-moments (not the L-moment ratios) between the first four sample L-moments ($\hat\lambda_r$) and the theoretical versions (λ) provided by this function be defined as

$$D^2 = (\hat\lambda_1 - \lambda_1)^2 + (\hat\lambda_2 - \lambda_2)^2 + (\hat\lambda_3 - \lambda_3)^2 + (\hat\lambda_4 - \lambda_4)^2.$$

Let $\alpha \in 0.10, 0.05, 0.01, 0.005, 0.001$ be upper tail probability levels (statistical significance thresholds). Let m denote the number of significant digits ($m \in 1, 2, 3$) in base 10 and n denote sample size. Let $\gamma = -\log(-\log(\alpha))$ be a transformation (in the style of a Gumbel reduced variate) (prob2grv). Using extensive simulation for many sample sizes, the α values, and computing $D^2(\alpha, m; n)$, it can be shown that the critical values for the D^2 distances are

$$D^2(\alpha) = \frac{1}{n}\exp\bigl[(-2.6607150 + 4.6154937m) - 1.217283\gamma\bigr],$$

wherein linear regression was used to estimate relation between each D^2 and $n \geq 5$ and the coefficients subsequently subjected to linear regression as functions of α. The **Examples** shows an implementation of the critical values.

Author(s)

W.H. Asquith

See Also

cdfben, pmfben, quaben

Examples

```
lmomben(para=list(para=c(3, 10)))

## Not run:
  # Code suitable for study of performance of Cho and Gaines D
  # against using the first for L-moments with controls for having
  # the Benford distribution as the true parent or alternative
  # distributions fit to the the L-moments of the Benford for the
  # first significant digit.
  # https://en.wikipedia.org/wiki/Benford%27s_law#Statistical_tests
  ChoGainesD <- function(x) {
    n <- length(x)
    d <- sapply(1:9, function(d) (length(x[x == d])/n - log10(1+1/d))^2)
    return(sqrt(n * sum(d)))
  }
  CritChoGainesD <- function(alpha=c("0.1", "0.05", "0.01")) {
    alpha <- as.character(as.numeric( alpha ))
    alpha <-   as.numeric(match.arg(  alpha ))
    if(alpha == 0.10) return(1.212)
    if(alpha == 0.05) return(1.330)
    if(alpha == 0.01) return(1.569)
    return(NULL)
  }
  D2lmom <- function(x, theolmr=NULL) {
    lmr <- lmoms(x)
    sum((lmr$lambdas[1:4] - theolmr$lambdas[1:4])^2)
  }
  CritD2lmom <-
    function(m, n, alpha=c("0.1", "0.05", "0.01", "0.005", "0.001")) {
      alpha <- as.character(as.numeric( alpha ))
      alpha <-   as.numeric(match.arg(  alpha ))
      exp((-2.6607150 + 4.6154937*m) - 1.217283*(-log(-log(alpha))))/n
  }

  nsim <- 2E4; n <- 100; alpha <- 0.05
  is_Benford_parent <- FALSE

  CritCGD <- CritChoGainesD(  alpha=alpha )
  CritLMR <- CritD2lmom(1, n, alpha=alpha )
```

```
  bens <- 1:9; pmf <- log10(1 + 1/bens) # for the Benford being true
  benlmr <- lmomben(list(para=c(1, 10))); dtype <- "nor" # Normal (say)
  parent <- lmom2par(benlmr, type=dtype)

  DF <- NULL
  ix <- seq(1, n, by=2)
  for(i in 1:nsim) {
    if(is_Benford_parent) {
      x <- sample(bens, n, replace=TRUE, prob=pmf)
    } else {
      x <- rlmomco(n, parent) # simulate from the parent
      x <- unlist(strsplit(sprintf("%2.0E", x), "E"))[ix]
      x <- as.integer(x) # complete extraction of the first digit
    }
    CGD    <- ChoGainesD(x)
    LMR    <- D2lmom(x, theolmr=benlmr)
    rejCGD <- ifelse(CGD > CritCGD, TRUE, FALSE)
    rejLMR <- ifelse(LMR > CritLMR, TRUE, FALSE)
    DF <- rbind(DF, data.frame(CGD=rejCGD, LMR=rejLMR))
  }
  print(summary(DF))
  if(is_Benford_parent) { # H0 is True
    CGDpct <- 100*(sum(as.numeric(DF$CGD)) / nsim - alpha) / alpha;
    LMRpct <- 100*(sum(as.numeric(DF$LMR)) / nsim - alpha) / alpha;
    message("The ChoGainesD rejection rate for alpha=", alpha,
            " is ", sum(as.numeric(DF$CGD)) / nsim,
            " (", round(CGDpct, digits=2), " percent difference).")
    message("The   D2lmom   rejection rate for alpha=", alpha,
            " is ", sum(as.numeric(DF$LMR)) / nsim,
            " (", round(LMRpct, digits=2), " percent difference).")
  } else { # H0 is False
    acceptH0_H0false_CDG <- sum(as.numeric(! DF$CGD)) / nsim
    acceptH0_H0false_LMR <- sum(as.numeric(! DF$LMR)) / nsim
    betaCDG <- round(1 - acceptH0_H0false_CDG, digits=2)
    betaLMR <- round(1 - acceptH0_H0false_LMR, digits=2)
    message("Power of ChoGainesD = ", betaCDG, ".")
    message("Power of   D2lmom   = ", betaLMR, ".")
  } #
## End(Not run)
```

---

lmomcau *Trimmed L-moments of the Cauchy Distribution*

Description

This function estimates the trimmed L-moments of the Cauchy distribution given the parameters (ξ and α) from parcau. The trimmed L-moments in terms of the parameters are $\lambda^{(1)}_1 = \xi$, $\lambda^{(1)}_2 = 0.69782723\alpha$, $\tau^{(1)}_{3,5,\cdots} = 0$, $\tau^{(1)}_4 = 0.34280842$, and $\tau^{(1)}_6 = 0.20274358$. These TL-moments (trim=1) are symmetrical for the first L-moments defined because $\mathrm{E}[X_{1:n}]$ and $\mathrm{E}[X_{n:n}]$ undefined expectations for the Cauchy.

Usage

```
lmomcau(para)
```

Arguments

`para`	The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the trimmed L-moments. First element is $\lambda_1^{(1)}$, second element is $\lambda_2^{(1)}$, and so on.
`ratios`	Vector of the L-moment ratios. Second element is $\tau^{(1)}$, third element is $\tau_3^{(1)}$ and so on.
`trim`	Level of symmetrical trimming used in the computation, which is unity.
`leftrim`	Level of left-tail trimming used in the computation, which is unity.
`rightrim`	Level of right-tail trimming used in the computation, which is unity.
`source`	An attribute identifying the computational source of the L-moments: "lmomcau".

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

`parcau`, `cdfcau`, `pdfcau`, `quacau`

Examples

```
X1 <- rcauchy(20)
lmomcau( parcau( TLmoms(X1, trim=1) ) )

alpha <- 30
tlmr <- theoTLmoms(vec2par(c(100, alpha), type="cau"), nmom=6, trim=1)
print( c(tlmr$lambdas[2] / alpha, tlmr$ratios[c(4,6)]), 8 )
```

lmomemu *L-moments of the Eta-Mu Distribution*

Description

This function estimates the L-moments of the Eta-Mu ($\eta : \mu$) distribution given the parameters (η and μ) from paremu. The L-moments in terms of the parameters are complex. They are computed here by the α_r probability-weighted moments in terms of the Yacoub integral (see cdfemu). The linear combination relating the L-moments to the conventional β_r probability-weighted moments is

$$\lambda_{r+1} = \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} \binom{r+k}{k} \beta_k,$$

for $r \geq 0$ and the linear combination relating the less common α_r to β_r is

$$\alpha_r = \sum_{k=0}^{r} (-1)^k \binom{r}{k} \beta_k,$$

and by definition the α_r are the expectations

$$\alpha_r \equiv E\{X\,[1 - F(X)]^r\},$$

and thus

$$\alpha_r = \int_{-\infty}^{\infty} x\,[1 - F(x)]^r f(x)\;\mathrm{d}x,$$

in terms of x, the PDF $f(x)$, and the CDF $F(x)$. Lastly, the α_r for the Eta-Mu distribution with substitution of the Yacoub integral are

$$\alpha_r = \int_{-\infty}^{\infty} Y_\mu\left(\eta,\ x\sqrt{2h\mu}\right)^r x\,f(x)\;\mathrm{d}x.$$

Yacoub (2007, eq. 21) provides an expectation for the jth moment of the distribution as given by

$$\mathrm{E}(x^j) = \frac{\Gamma(2\mu + j/2)}{h^{\mu+j/2}(2\mu)^{j/2}\Gamma(2\mu)} \times {}_2F_1(\mu + j/4 + 1/2, \mu + j/4; \mu + 1/2; (H/h)^2),$$

where ${}_2F_1(a, b; c; z)$ is the Gauss hypergeometric function of Abramowitz and Stegun (1972, eq. 15.1.1) and $h = 1/(1 - \eta^2)$ (format 2 of Yacoub's paper and the format exclusively used by **lmomco**). The `lmomemu` function optionally solves for the mean ($j = 1$) using the above equation in conjunction with the mean as computed by the order statistic minimums. The ${}_2F_1(a, b; c; z)$ is defined as

$${}_2F_1(a, b; c; z) = \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_{i=0}^{\infty} \frac{\Gamma(a + i)\Gamma(b + i)}{\Gamma(c + i)} \frac{z^i}{n!}.$$

Yacoub (2007, eq. 21) is used to compute the mean.

Usage

```
lmomemu(para, nmom=5, paracheck=TRUE, tol=1E-6, maxn=100)
```

Arguments

`para`	The parameters of the distribution.
`nmom`	The number of L-moments to compute.
`paracheck`	A logical controlling whether the parameters and checked for validity.
`tol`	An absolute tolerance term for series convergence of the Gauss hypergeometric function when the Yacoub (2007) mean is to be computed.
`maxn`	The maximum number of interations in the series of the Gauss hypergeometric function when the Yacoub (2007) mean is to be computed.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomemu".
`yacoubsmean`	A list containing the mean, convergence error, and number of iterations in the series until convergence.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

`paremu`, `cdfemu`, `pdfemu`, `quaemu`

Examples

```
## Not run:
emu <- vec2par(c(.19,2.3), type="emu")
lmomemu(emu)

par <- vec2par(c(.67, .5), type="emu")
```

```
lmomemu(par)$lambdas
cdf2lmoms(par, nmom=4)$lambdas
system.time(lmomemu(par))
system.time(cdf2lmoms(par, nmom=4))

# This extensive sequence of operations provides very important
# perspective on the L-moment ratio diagram of L-skew and L-kurtosis.
# But more importantly this example demonstrates the L-moment
# domain of the Kappa-Mu and Eta-Mu distributions and their boundaries.
#
t3 <- seq(-1,1,by=.0001)
plotlmrdia(lmrdia(), xlim=c(-0.05,0.5), ylim=c(-0.05,.2))
# The following polynomials are used to define the boundaries of
# both distributions. The applicable inequalities for these
# are not provided for these polynomials as would be in deeper
# implementation---so don't worry about wild looking trajectories.
"KMUup" <- function(t3) {
             return(0.1227 - 0.004433*t3 - 2.845*t3^2 +
                    + 18.41*t3^3 - 50.08*t3^4 + 83.14*t3^5 +
                    - 81.38*t3^6 + 43.24*t3^7 - 9.600*t3^8)}

"KMUdnA" <- function(t3) {
              return(0.1226 - 0.3206*t3 - 102.4*t3^2 - 4.753E4*t3^3 +
                     - 7.605E6*t3^4 - 5.244E8*t3^5 - 1.336E10*t3^6)}

"KMUdnB" <- function(t3) {
              return(0.09328 - 1.488*t3 + 16.29*t3^2 - 205.4*t3^3 +
                     + 1545*t3^4 - 5595*t3^5 + 7726*t3^6)}

"KMUdnC" <- function(t3) {
              return(0.07245 - 0.8631*t3 + 2.031*t3^2 - 0.01952*t3^3 +
                     - 0.7532*t3^4 + 0.7093*t3^5 - 0.2156*t3^6)}

"EMUup" <- function(t3) {
              return(0.1229 - 0.03548*t3 - 0.1835*t3^2 + 2.524*t3^3 +
                     - 2.954*t3^4 + 2.001*t3^5 - 0.4746*t3^6)}

# Here, we are drawing the trajectories of the tabulated parameters
# and L-moments within the internal storage of lmomco.
lines(.lmomcohash$EMU_lmompara_byeta$T3,
      .lmomcohash$EMU_lmompara_byeta$T4,   col=7, lwd=0.5)
lines(.lmomcohash$KMU_lmompara_bykappa$T3,
      .lmomcohash$KMU_lmompara_bykappa$T4, col=8, lwd=0.5)

# Draw the polynomials
lines(t3, KMUdnA(t3), lwd=4, col=2, lty=4)
lines(t3, KMUdnB(t3), lwd=4, col=3, lty=4)
lines(t3, KMUdnC(t3), lwd=4, col=4, lty=4)
lines(t3, EMUup(t3),  lwd=4, col=5, lty=4)
lines(t3, KMUup(t3),  lwd=4, col=6, lty=4)

## End(Not run)
```

lmomexp *L-moments of the Exponential Distribution*

Description

This function estimates the L-moments of the Exponential distribution given the parameters (ξ and α) from parexp. The L-moments in terms of the parameters are $\lambda_1 = \xi + \alpha$, $\lambda_2 = \alpha/2$, $\tau_3 = 1/3$, $\tau_4 = 1/6$, and $\tau_5 = 1/10$.

Usage

```
lmomexp(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: “lmomexp”.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parexp, cdfexp, pdfexp, quaexp

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomexp(parexp(lmr))
```

`lmomgam` *L-moments of the Gamma Distribution*

Description

This function estimates the L-moments of the Gamma distribution given the parameters (α and β) from pargam. The L-moments in terms of the parameters are complicated and solved numerically. This function is adaptive to the 2-parameter and 3-parameter Gamma versions supported by this package. For legacy reasons, **lmomco** continues to use a port of Hosking's FORTRAN into R if the 2-parameter distribution is used but the 3-parameter generalized Gamma distribution calls upon theoLmoms.max.ostat. Alternatively, the theoTLmoms could be used: theoTLmoms(para) is conceptually equivalent to the internal calls to theoLmoms.max.ostat made for the lmomgam implementation.

Usage

```
lmomgam(para, ...)
```

Arguments

para	The parameters of the distribution.
...	Additional arguments to pass to theoLmoms.max.ostat.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomgam".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, p. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pargam, cdfgam, pdfgam, quagam

Examples

```
lmomgam(pargam(lmoms(c(123,34,4,654,37,78))))

## Not run:
# 3-p Generalized Gamma Distribution and comparisons of 3-p Gam parameterization.
#     1st parameter A[lmomco] = A[gamlss] =  exp(A[flexsurv])
#     2nd parameter B[lmomco] = B[gamlss] =      B[flexsurv]
#     3rd parameter C[lmomco] = C[gamlss] -->    C[flexsurv] = B[lmomco]/C[lmomco]
lmomgam(vec2par(c(7.4, 0.2, 14), type="gam"), nmom=5)$lambdas      # numerics
lmoms(gamlss.dist::rGG(50000, mu=7.4, sigma=0.2, nu=14))$lambdas   # simulation
lmoms(flexsurv::rgengamma(50000, log(7.4), 0.2, Q=0.2*14))$lambdas # simulation
#[1]  5.364557537  1.207492689 -0.110129217  0.067007941 -0.006747895
#[1]  5.366707749  1.209455502 -0.108354729  0.066360223 -0.006716783
#[1]  5.356166684  1.197942329 -0.106745364  0.069102821 -0.008293398#
## End(Not run)
```

---

lmomgdd *L-moments of the Gamma Difference Distribution*

Description

This function estimates the L-moments of the Gamma Difference distribution (Klar, 2015) given the parameters ($\alpha_1 > 0$, $\beta_1 > 0$, $\alpha_2 > 0$, $\beta_2 > 0$) from pargdd. The L-moments in terms of the parameters higher than the mean are complex and numerical methods are required. The mean is

$$\lambda_1 = \frac{\alpha_1}{\beta_1} - \frac{\alpha_2}{\beta_2}.$$

The product moments, however, have simple expressions, the variance and skewness, respectively are

$$\sigma^2 = \frac{\alpha_1}{\beta_2^2} + \frac{\alpha_2}{\beta_2^2},$$

and

$$\gamma = \frac{2\bigl(\alpha_1 \beta_2^3 + \alpha_2 \beta_2^2\bigr)}{\bigl(\alpha_2 \beta_1^2 + \alpha_2 \beta_1^2\bigr)^{3/2}}.$$

Usage

```
lmomgdd(para, nmom=6, paracheck=TRUE, silent=TRUE, ...)
```

Arguments

`para`	The parameters of the distribution.
`nmom`	The number of L-moment to numerically compute for the distribution.
`paracheck`	A logical controlling whether the parameters are checked for validity.
`silent`	The argument of `silent` for the `try()` operation wrapped on `integrate()`.
`...`	Additional argument to pass.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomgdd".

Note

Experimental Summer 2024—For a symmetrical version of the distribution, the relation between τ_4 and τ_6 and other coupling to λ_2 can be computed using the following recipe:

```
LMR <- NULL
plotlmrdia46(lmrdia46(), autolegend=TRUE, xleg="topleft")
for(i in 1:4000) {
  if(length(grep("00$", as.character(i)))) message(i)
  para <- 10^runif(2, min=-3, max=3)
  para <- list(para=c(para[1], para[2], para[1], para[2]), type="gdd")
  lmr  <- lmomgdd(para, nmom=6, subdivisions=200)
  points(lmr$ratios[4], lmr$ratios[6], pch=1, cex=0.8, lwd=0.8)
  LMR <- rbind(LMR, data.frame(A12=para$para[1], B12=para$para[2],
               L1=lmr$lambdas[1], L2=lmr$lambdas[2], T3=lmr$ratios[3],
               T4=lmr$ratios[ 4], T5=lmr$ratios[ 3], T6=lmr$ratios[6]))
}
LMR <- LMR[completed.cases(LMR), ]
LMR <- LMR[abs(  LMR$T3) < 0.01, ]
LMR <- LMR[order(LMR$T4),        ]
```

We have swept through, hopefully, a sufficiently large span of viable parameter values under a constrain of symmetry. The following recipe continues in post-processing with the goal of producing a polynomial approximation between τ_4 and τ_6 for lmrdia46.

```
plotlmrdia46(lmrdia46(), autolegend=TRUE, xleg="topleft")
points(LMR$T4, LMR$T6, pch=1, cex=0.8, lwd=0.8)
LM <- lm(T6~I(T4  ) + I(T4^2) + I(T4^3) + I(T4^4) +
            I(T4^5) + I(T4^6) + I(T4^7) + I(T4^8), data=LMR)
lines(LMR$T4, fitted.values(LM), col="blue", lwd=3)

res <- residuals(LM)
plot(fitted.values(LM), res, ylim=c(-0.02, 0.02))
abline(h=c(-0.002, 0.002), col="red")
LMRthin <- LMR[abs(res) < 0.002, ]

LM <- lm(T6~I(T4  ) + I(T4^2) + I(T4^3) + I(T4^4)+
            I(T4^5) + I(T4^6) + I(T4^7) + I(T4^8), data=LMRthin)

plot(  LMRthin$T4, fitted.values(LM), col="blue", type="l", lwd=3  )
points(LMRthin$T4, LMRthin$T6,        col="red",  cex=0.4,  lwd=0.5)

tau4 <- c(lmrdia46()$nor$tau4, 0.1227, 0.123, 0.125, seq(0.13, 1, by=0.01))
tau6 <- predict(LM, newdata=data.frame(T4=tau4))
names(tau6) <- NULL
gddsymt46   <- data.frame(tau4=tau4, tau6=tau6)

gddsymt46f <- function(t4) { # print(coefficients(LM))
  coe <- c( -0.0969112,    2.1743687, -12.8878580,  47.8931168, -108.0871549,
           156.9200440, -139.5599813,  69.3492358, -14.7052424)
  ix <- seq_len(length(coes))-1
  sapply(t4, function(t) sum(coes[ix+1]*t^ix))
} # This function is inserted into the lmrdia46() for deployment as symgdd.

plotlmrdia46(lmrdia46(), autolegend=TRUE, xleg="topleft")
lines(          tau4, gddsymt46f(tau4), lwd=3, col="deepskyblue3")
lines(gddsymt46$tau4, gddsymt46$tau6,   lwd=3, col="deepskyblue3")
legend("bottomright", "Symmetrical Gamma Difference distribution",
       bty="n", cex=0.9, lwd=3, col="deepskyblue3")
```

This is the first known derivation of the relation between these two L-moment ratios for the symmetrical version of this distribution. The quantities recorded in the LMR data frame in the recipe can be useful for additional study of the quality of numerical implementation of the distribution by **lmomco**. Next, for purposes of helping parameter estimation for $\alpha_1 = \alpha_2$ and $\beta_1 = \beta_2$ and τ_4, let us build a polynomial for α estimation from τ_4:

```
tlogit <- function(x) log(x/(1-x))
ilogit <- function(x) 1/(1+exp(-x))
A12l <-    log(LMRthin$A12)
```

```
T4l  <- tlogit(LMRthin$T4 )
A12p <- exp( approx(T4l, y=A12l, xout=tlogit(tau4))$y )

plot(  tlogit(tau4), A12p, log="y",     col="blue", type="l", lwd=3  )
points(LMRthin$T4, LMRthin$A12, col="red",  cex=0.4,  lwd=0.5)
```

Author(s)

W.H. Asquith

See Also

pargdd, cdfgdd, pdfgdd, quagdd

Examples

```
#
```

---

lmomgep *L-moments of the Generalized Exponential Poisson Distribution*

Description

This function estimates the L-moments of the Generalized Exponential Poisson (GEP) distribution given the parameters (β, κ, and h) from pargep. The L-moments in terms of the parameters are best expressed in terms of the expectations of order statistic maxima $\mathrm{E}[X_{n:n}]$ for the distribution. The fundamental relation is

$$\lambda_r = \sum_{k=1}^{r} (-1)^{r-k} k^{-1} \binom{r-1}{k-1} \binom{r+k-2}{k-1} \mathrm{E}[X_{k:k}].$$

The L-moments do not seem to have been studied for the GEP. The challenge is the solution to $\mathrm{E}[X_{n:n}]$ through an expression by Barreto-Souza and Cribari-Neto (2009) that is

$$\mathrm{E}[X_{n:n}] = \frac{\beta\, h\, \Gamma(\kappa+1)\, \Gamma(n\kappa+1)}{n\, \Gamma(n)\, (1-\exp(-h))^{n\kappa}} \sum_{j=0}^{\infty} \frac{(-1)^j \exp(-h(j+1))}{\Gamma(n\kappa - j)\, \Gamma(j+1)}\, F_{22}^{12}(h(j+1)),$$

where $F_{22}^{12}(h(j+1))$ is the Barnes Extended Hypergeometric function with arguments reflecting those needed for the GEP (see comments under BEhypergeo).

Usage

```
lmomgep(para, byqua=TRUE)
```

Arguments

para	The parameters of the distribution.
byqua	A logical triggering the theoLmoms.max.ostat instead of using the mathematics of Barreto-Souza and Cribari-Neto (2009) (see Details).

Details

The mathematics (not of L-moments but $\mathrm{E}[X_{n:n}]$) shown by Barreto-Souza and Cribari-Neto (2009) are correct but are apparently subject to considerable numerical issues even with substantial use of logarithms and exponentiation in favor of multiplication and division in the above formula for $\mathrm{E}[X_{n:n}]$. Testing indicates that numerical performance is better if the non-j-dependent terms in the infinite sum remain *inside* it. Testing also indicates that the edges of performance can be readily hit with large κ and less so with large h. It actually seems superior to not use the above equation for L-moment computation based on $\mathrm{E}[X_{n:n}]$ but instead rely on expectations of maxima order statistics (`expect.max.ostat`) from numerical integration of the quantile function (`quagep`) as is implementated in `theoLmoms.max.ostat`. This is the reason that the `byqua` argument is available and set to the shown default. Because the GEP is experimental, this function provides two approaches for λ_r computation for research purposes.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: “lmomgep”.

Author(s)

W.H. Asquith

References

Barreto-Souza, W., and Cribari-Neto, F., 2009, A generalization of the exponential-Poisson distribution: Statistics and Probability, 79, pp. 2493–2500.

See Also

`pargep`, `cdfgep`, `pdfgep`, `quagep`

Examples

```
## Not run:
gep <- vec2par(c(2, 1.5, 3), type="gep")
lmrA <- lmomgep(gep, byqua=TRUE);   print(lmrA)
lmrB <- lmomgep(gep, byqua=FALSE);  print(lmrB)

# Because the L-moments of the Generalized Exponential Poisson are computed
# strictly from the expectations of the order statistic extrema, lets us evaluate
# by theoretical integration of the quantile function and simulation:
```

```
set.seed(10); gep <- vec2par(c(2, 1.5, 3), type="gep")
lmr  <- lmomgep(gep, byqua=FALSE)
E33a <- (lmr$lambdas[3] + 3*lmr$lambdas[2] + 2*lmr$lambdas[1])/2  # 2.130797
E33b <- expect.max.ostat(3, para=gep, qua=quagep)                 # 2.137250
E33c <- mean(replicate(20000, max(quagep(runif(3), gep))))        # 2.140226
# See how the E[X_{3:3}] by the formula shown in this documentation results in
# a value that is about 0.007 too small. Now this might now seem large but it
# is a difference.  Try gep <- list(para=c(2, 1.5, 13), type="gep") or
#  gep <- list(para=c(2, .08, 21), type="gep"), which fails on byqua=TRUE
## End(Not run)
```

---

lmomgev *L-moments of the Generalized Extreme Value Distribution*

Description

This function estimates the L-moments of the Generalized Extreme Value distribution given the parameters (ξ, α, and κ) from pargev. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \frac{\alpha}{\kappa}(1 - \Gamma(1+\kappa)),$$

$$\lambda_2 = \frac{\alpha}{\kappa}(1 - 2^{-\kappa})\Gamma(1+\kappa),$$

$$\tau_3 = \frac{2(1 - 3^{-\kappa})}{1 - 2^{-\kappa}} - 3, \text{ and}$$

$$\tau_4 = \frac{5(1 - 4^{-\kappa}) - 10(1 - 3^{-\kappa}) + 6(1 - 2^{-\kappa})}{1 - 2^{-\kappa}}.$$

Usage

```
lmomgev(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomgev".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pargev, cdfgev, pdfgev, quagev

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmomgev(pargev(lmr))

## Not run:
# The Gumbel is a limiting version of the maxima regardless of parent. The GLO,
# PE3 (twice), and GPA are studied here. A giant number of events to simulate is made.
# Then numbers of events per year before the annual maxima are computed are specified.
# The Gumbel is a limiting version of the maxima regardless of parent. The GLO,
# PE3 (twice), and GPA are studied here. A giant number of events to simulate is made.
# Then numbers of events per year before the annual maxima are computed are specified.
nevents <- 100000
nev_yr <- c(1,2,3,4,5,6,10,15,20,30,50,100,200,500); n <- length(nev_yr)
pdf("Gumbel_in_the_limit.pdf", useDingbats=FALSE)
# Draw the usually L-moment ratio diagram but only show a few of the
# three parameter families.
plotlmrdia(lmrdia(), xlim=c(-.5,0.5), ylim=c(0,0.3), nopoints=TRUE,
           autolegend=TRUE, noaep4=TRUE, nogov=TRUE, xleg=0.1, yleg=0.3)
gum <- lmrdia()$gum # extract the L-skew and L-kurtosis of the Gumbel
points(gum[1], gum[2], pch=10, cex=3, col=2) # draw the Gumbel

para <- parglo(vec2lmom(c(1,.1,0))) # generalized logistic
t3 <- t4 <- rep(NA, n) # define
for(k in 1:n) { # generate GLO time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/nev_yr[k], function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=3)

para <- parglo(vec2lmom(c(1,.1,0.3))) # generalized logistic
t3 <- t4 <- rep(NA, n) # define
for(k in 1:n) { # generate GLO time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/nev_yr[k], function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
```

```
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=3)

para <- parglo(vec2lmom(c(1,.1,-0.3))) # generalized logistic
t3 <- t4 <- rep(NA, n) # define
for(k in 1:n) { # generate GLO time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/nev_yr[k], function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=3)

para <- parpe3(vec2lmom(c(1,.1,.4))) # Pearson type III
t3 <- t4 <- rep(NA, n) # reset
for(k in 1:n) { # generate PE3 time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/k, function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=6)

para <- parpe3(vec2lmom(c(1,.1,0))) # Pearson type III
t3 <- t4 <- rep(NA, n) # reset
for(k in 1:n) { # generate another PE3 time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/k, function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=6)

para <- parpe3(vec2lmom(c(1,.1,-.4))) # Pearson type III
t3 <- t4 <- rep(NA, n) # reset
for(k in 1:n) { # generate PE3 time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/k, function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=6)

para <- pargpa(vec2lmom(c(1,.1,0))) # generalized Pareto
t3 <- t4 <- rep(NA, n) # reset
for(k in 1:n) { # generate GPA time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/k, function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=4)

para <- pargpa(vec2lmom(c(1,.1,.4))) # generalized Pareto
t3 <- t4 <- rep(NA, n) # reset
for(k in 1:n) { # generate GPA time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/k, function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=4)

para <- pargpa(vec2lmom(c(1,.1,-.4))) # generalized Pareto
t3 <- t4 <- rep(NA, n) # reset
```

```
for(k in 1:n) { # generate GPA time series of annual maxima with k-events per year
   lmr <- lmoms(sapply(1:nevents/k, function(i) max(rlmomco(nev_yr[k], para))))
   t3[k] <- lmr$ratios[3]; t4[k] <- lmr$ratios[4]
}
lines(t3, t4, lwd=0.8); points(t3, t4, lwd=0.8, pch=21, bg=4)
dev.off() #
## End(Not run)
```

---

lmomgld *L-moments of the Generalized Lambda Distribution*

Description

This function estimates the L-moments of the Generalized Lambda distribution given the parameters (ξ, α, κ, and h) from vec2par. The L-moments in terms of the parameters are complicated; however, there are analytical solutions. There are no simple expressions of the parameters in terms of the L-moments. The first L-moment or the mean is

$$\lambda_1 = \xi + \alpha\left(\frac{1}{\kappa+1} - \frac{1}{h+1}\right).$$

The second L-moment or L-scale in terms of the parameters and the mean is

$$\lambda_2 = \xi + \frac{2\alpha}{(\kappa+2)} - 2\alpha\left(\frac{1}{h+1} - \frac{1}{h+2}\right) - \xi.$$

The third L-moment in terms of the parameters, the mean, and L-scale is

$$Y = 2\xi + \frac{6\alpha}{(\kappa+3)} - 3(\alpha+\xi) + \xi, \text{ and}$$

$$\lambda_3 = Y + 6\alpha\left(\frac{2}{h+2} - \frac{1}{h+3} - \frac{1}{h+1}\right).$$

The fourth L-moment in termes of the parameters and the first three L-moments is

$$Y = \frac{-3}{h+4}\left(\frac{2}{h+2} - \frac{1}{h+3} - \frac{1}{h+1}\right),$$

$$Z = \frac{20\xi}{4} + \frac{20\alpha}{(\kappa+4)} - 20Y\alpha, \text{ and}$$

$$\lambda_4 = Z - 5(\kappa + 3(\alpha+\xi) - \xi) + 6(\alpha+\xi) - \xi.$$

It is conventional to express L-moments in terms of only the parameters and not the other L-moments. Lengthy algebra and further manipulation yields such a system of equations. The L-moments are

$$\lambda_1 = \xi + \alpha\left(\frac{1}{\kappa+1} - \frac{1}{h+1}\right),$$

$$\lambda_2 = \alpha \left(\frac{\kappa}{(\kappa+2)(\kappa+1)} + \frac{h}{(h+2)(h+1)} \right),$$

$$\lambda_3 = \alpha \left(\frac{\kappa(\kappa-1)}{(\kappa+3)(\kappa+2)(\kappa+1)} - \frac{h(h-1)}{(h+3)(h+2)(h+1)} \right), \text{ and}$$

$$\lambda_4 = \alpha \left(\frac{\kappa(\kappa-2)(\kappa-1)}{(\kappa+4)(\kappa+3)(\kappa+2)(\kappa+1)} + \frac{h(h-2)(h-1)}{(h+4)(h+3)(h+2)(h+1)} \right).$$

The L-moment ratios are

$$\tau_3 = \frac{\kappa(\kappa-1)(h+3)(h+2)(h+1) - h(h-1)(\kappa+3)(\kappa+2)(\kappa+1)}{(\kappa+3)(h+3)\times[\kappa(h+2)(h+1)+h(\kappa+2)(\kappa+1)]}, \text{ and}$$

$$\tau_4 = \frac{\kappa(\kappa-2)(\kappa-1)(h+4)(h+3)(h+2)(h+1) + h(h-2)(h-1)(\kappa+4)(\kappa+3)(\kappa+2)(\kappa+1)}{(\kappa+4)(h+4)(\kappa+3)(h+3)\times[\kappa(h+2)(h+1)+h(\kappa+2)(\kappa+1)]}.$$

The pattern being established through symmetry, even higher L-moment ratios are readily obtained. Note the alternating substraction and addition of the two terms in the numerator of the L-moment ratios (τ_r). For odd $r \geq 3$ substraction is seen and for even $r \geq 3$ addition is seen. For example, the fifth L-moment ratio is

$$N1 = \kappa(\kappa-3)(\kappa-2)(\kappa-1)(h+5)(h+4)(h+3)(h+2)(h+1),$$

$$N2 = h(h-3)(h-2)(h-1)(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2)(\kappa+1),$$

$$D1 = (\kappa+5)(h+5)(\kappa+4)(h+4)(\kappa+3)(h+3),$$

$$D2 = [\kappa(h+2)(h+1)+h(\kappa+2)(\kappa+1)], \text{ and}$$

$$\tau_5 = \frac{N1 - N2}{D1 \times D2}.$$

By inspection the τ_r equations are not applicable for negative integer values $k = \{-1, -2, -3, -4, \dots\}$ and $h = \{-1, -2, -3, -4, \dots\}$ as division by zero will result. There are additional, but difficult to formulate, restrictions on the parameters both to define a valid Generalized Lambda distribution as well as valid L-moments. Verification of the parameters is conducted through `are.pargld.valid`, and verification of the L-moment validity is conducted through `are.lmom.valid`.

Usage

```
lmomgld(para)
```

Arguments

`para`	The parameters of the distribution.

Value

An R list is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomgld".

Author(s)

W.H. Asquith

Source

Derivations conducted by W.H. Asquith on February 11 and 12, 2006.

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karvanen, J., Eriksson, J., and Koivunen, V., 2002, Adaptive score functions for maximum likelihood ICA: Journal of VLSI Signal Processing, v. 32, pp. 82–92.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distibutions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

`pargld`, `cdfgld`, `pdfgld`, `quagld`

Examples

```
## Not run:
lmomgld(vec2par(c(10,10,0.4,1.3),type='gld'))

## End(Not run)

## Not run:
PARgld <- vec2par(c(0,1,1,.5), type="gld")
theoTLmoms(PARgld, nmom=6)
lmomgld(PARgld)

## End(Not run)
```

lmomglo *L-moments of the Generalized Logistic Distribution*

Description

This function estimates the L-moments of the Generalized Logistic distribution given the parameters (ξ, α, and κ) from parglo. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \alpha\left(\frac{1}{\kappa} - \frac{\pi}{\sin(\kappa\pi)}\right),$$

$$\lambda_2 = \frac{\alpha\kappa\pi}{\sin(\kappa\pi)},$$

$$\tau_3 = -\kappa, \text{ and}$$

$$\tau_4 = \frac{(1+5\tau_3^2)}{6} = \frac{(1+5\kappa^2)}{6}.$$

Usage

```
lmomglo(para)
```

Arguments

para	The parameters of the distribution.

Value

An R `list` is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is `0`.
leftrim	Level of left-tail trimming used in the computation, which is `NULL`.
rightrim	Level of right-tail trimming used in the computation, which is `NULL`.
source	An attribute identifying the computational source of the L-moments: “lmomglo”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parglo, cdfglo, pdfglo, quaglo

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomglo(parglo(lmr))
```

---

lmomgno *L-moments of the Generalized Normal Distribution*

Description

This function estimates the L-moments of the Generalized Normal (Log-Normal3) distribution given the parameters (ξ, α, and κ) from pargno. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \frac{\alpha}{\kappa}(1 - \exp(\kappa^2/2), \text{ and}$$

$$\lambda_2 = \frac{\alpha}{\kappa}(\exp(\kappa^2/2)(1 - 2\Phi(-\kappa/\sqrt{2})),$$

where Φ is the cumulative distribution of the Standard Normal distribution. There are no simple expressions for τ_3, τ_4, and τ_5. Logarthmic transformation of the data prior to fitting of the Generalized Normal distribution is not required. The distribution is algorithmically the same with subtle parameter modifications as the Log-Normal3 distribution (see lmomln3, parln3). If desired for user-level control of the lower bounds of a Log-Normal-like distribution is required, then see parln3.

Usage

```
lmomgno(para)
```

Arguments

para The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomgno".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`pargno`, `cdfgno`, `pdfgno`, `quagno`, `lmomln3`

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomgno(pargno(lmr))
```

`lmomgov` *L-moments of the Govindarajulu Distribution*

Description

This function estimates the L-moments of the Govindarajulu distribution given the parameters (ξ, α, and β) from `pargov`. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \frac{2\alpha}{\beta + 2},$$

$$\lambda_2 = \frac{2\alpha\beta}{(\beta + 2)(\beta + 3)},$$

$$\tau_3 = \frac{\beta - 2}{\beta + 4}\text{, and}$$

$$\tau_4 = \frac{(\beta - 5)(\beta - 1)}{(\beta + 4)(\beta + 5)}.$$

The limits of τ_3 are $(-1/2, 1)$ for $\beta \rightarrow 0$ and $\beta \rightarrow \infty$.

Usage

```
lmomgov(para)
```

Arguments

`para`	The parameters of the distribution.

Value

An R list is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomgov".

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Nair, N.U., Sankaran, P.G., Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

Nair, N.U., Sankaran, P.G., and Vineshkumar, B., 2012, The Govindarajulu distribution—Some Properties and applications: Communications in Statistics, Theory and Methods, v. 41, no. 24, pp. 4391–4406.

See Also

`pargov`, `cdfgov`, `pdfgov`, `quagov`

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmorph(lmr)
lmomgov(pargov(lmr))
## Not run:
Bs <- exp(seq(log(.01),log(10000),by=.05))
T3 <- (Bs-2)/(Bs+4)
T4 <- (Bs-5)*(Bs-1)/((Bs+4)*(Bs+5))
plotlmrdia(lmrdia(), autolegend=TRUE)
points(T3, T4)
T3s <- c(-0.5,T3,1)
T4s  <- c(0.25,T4,1)
the.lm <- lm(T4s~T3s+I(T3s^2)+I(T3s^3)+I(T3s^4)+I(T3s^5))
lines(T3s, predict(the.lm), col=2)
max(residuals(the.lm))
summary(the.lm)

## End(Not run)
```

---

lmomgpa — *L-moments of the Generalized Pareto Distribution*

Description

This function estimates the L-moments of the Generalized Pareto distribution given the parameters (ξ, α, and κ) from pargpa. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \frac{\alpha}{\kappa + 1},$$

$$\lambda_2 = \frac{\alpha}{(\kappa + 2)(\kappa + 1)},$$

$$\tau_3 = \frac{(1 - \kappa)}{(\kappa + 3)}, \text{ and}$$

$$\tau_4 = \frac{(1 - \kappa)(2 - \kappa)}{(\kappa + 4)(\kappa + 3)}.$$

Usage

```
lmomgpa(para)
```

Arguments

`para` The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomgpa".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`pargpa`, `cdfgpa`, `pdfgpa`, `quagpa`

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomgpa(pargpa(lmr))
```

`lmomgpaRC` *B-type L-moments of the Generalized Pareto Distribution with Right-Tail Censoring*

Description

This function computes the "B"-type L-moments of the Generalized Pareto distribution given the parameters (ξ, α, and κ) from pargpaRC and the right-tail censoring fraction ζ. The B-type L-moments in terms of the parameters are

$$\lambda_1^B = \xi + \alpha m_1,$$

$$\lambda_2^B = \alpha(m_1 - m_2),$$

$$\lambda_3^B = \alpha(m_1 - 3m_2 + 2m_3),$$

$$\lambda_4^B = \alpha(m_1 - 6m_2 + 10m_3 - 5m_4), \text{ and}$$

$$\lambda_5^B = \alpha(m_1 - 10m_2 + 30m_3 - 35m_4 + 14m_5),$$

where $m_r = \{1-(1-\zeta)^{r+\kappa}\}/(r+\kappa)$ and ζ is the right-tail censor fraction or the probability $\mathrm{Pr}\{\}$ that x is less than the quantile at ζ nonexceedance probability: $(\mathrm{Pr}\{x < X(\zeta)\})$. In other words, if $\zeta = 1$, then there is no right-tail censoring. Finally, the RC in the function name is to denote Right-tail Censoring.

Usage

```
lmomgpaRC(para)
```

Arguments

para	The parameters of the distribution. Note that if the ζ part of the parameters (see pargpaRC) is not present then zeta=1 (no right-tail censoring) is assumed.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomgpaRC".
message	For clarity, this function adds the unusual message to an L-moment object that the lambdas and ratios are B-type L-moments.
zeta	The censoring fraction. Assumed equal to unity if not present in the gpa parameter object.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

pargpa, pargpaRC, lmomgpa, cdfgpa, pdfgpa, quagpa

Examples

```
para <- vec2par(c(1500,160,.3),type="gpa") # build a GPA parameter set
lmorph(lmomgpa(para))
lmomgpaRC(para) # zeta = 1 is internally assumed if not available
# The previous two commands should output the same parameter values from
# independent code bases.
# Now assume that we have the sample parameters, but the zeta is nonunity.
para$zeta = .8
lmomgpaRC(para) # The B-type L-moments.
```

---

lmomgum *L-moments of the Gumbel Distribution*

Description

This function estimates the L-moments of the Gumbel distribution given the parameters (ξ and α) from pargum. The L-moments in terms of the parameters are $\lambda_1 = [\xi + (0.5722\dots)\alpha]$, $\lambda_2 = \alpha\log(2)$, $\tau_3 = 0.169925$, $\tau_4 = 0.150375$, and $\tau_5 = 0.055868$.

Usage

```
lmomgum(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.

`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomgum".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

pargum, cdfgum, pdfgum, quagum

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmomgum(pargum(lmr))
```

---

lmomkap *L-moments of the Kappa Distribution*

Description

This function estimates the L-moments of the Kappa distribution given the parameters (ξ, α, κ, and h) from parkap. The L-moments in terms of the parameters are complicated and are solved numerically. If the parameter $k = 0$ (is small or near zero) then let

$$d_r = \gamma + \log(-h) + \mathrm{digamma}(-r/h) \text{ for } h < 0$$

$$d_r = \gamma + \log(r) \text{ for } h = 0 \text{ (is small)}$$

$$d_r = \gamma + \log(h) + \mathrm{digamma}(1 + r/h) \text{ for } h > 0$$

or if $k > -1$ (nonzero) then let

$$g_r = \frac{\Gamma(1+k)\Gamma(-r/h - k)}{-h^k\,\Gamma(-r/h)} \text{ for } h < 0$$

$$g_r = \frac{\Gamma(1+k)}{r^k} \times (1 - 0.5hk(1+k)/r) \text{ for } h = 0 \text{ (is small)}$$

$$g_r = \frac{\Gamma(1+k)\Gamma(1+r/h)}{h^g\,\Gamma(1+k+r/h)} \text{ for } h > 0$$

where r is L-moment order, γ is Euler's constant, and for $h = 0$ the term to the right of the multiplication is not in Hosking (1994) or Hosking and Wallis (1997) for exists within Hosking's FORTRAN code base.

The probability-weighted moments (β_r; pwm2lmom) for $k = 0$ (is small or near zero) are

$$r\beta_{r-1} = \xi + (\alpha/\kappa)[1 - d_r]$$

or if $k > -1$ (nonzero) then

$$r\beta_{r-1} = \xi + (\alpha/\kappa)[1 - g_r]$$

Usage

```
lmomkap(para, nmom=5)
```

Arguments

para	The parameters of the distribution.
nmom	The number of moments to compute. Default is 5.

Value

An R `list` is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is `0`.
leftrim	Level of left-tail trimming used in the computation, which is `NULL`.
rightrim	Level of right-tail trimming used in the computation, which is `NULL`.
source	An attribute identifying the computational source of the L-moments: "lmomkap".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1994, The four-parameter kappa distribution: IBM Journal of Reserach and Development, v. 38, no. 3, pp. 251–258.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parkap, cdfkap, pdfkap, quakap

Examples

```
lmr <- lmoms(c(123, 34, 4,78, 45, 234, 65, 2, 3, 5, 76, 7, 80))
lmomkap(parkap(lmr))
```

---

lmomkmu *L-moments of the Kappa-Mu Distribution*

Description

This function estimates the L-moments of the Kappa-Mu ($\kappa : \mu$) distribution given the parameters (ν and α) from parkmu. The L-moments in terms of the parameters are complex. They are computed here by the α_r probability-weighted moments in terms of the Marcum Q-function (see cdfkmu). The linear combination relating the L-moments to the β_r probability-weighted moments is

$$\lambda_{r+1} = \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} \binom{r+k}{k} \beta_k,$$

for $r \geq 0$ and the linear combination relating α_r to β_r is

$$\alpha_r = \sum_{k=0}^{r} (-1)^k \binom{r}{k} \beta_k,$$

and by definition the α_r are the expectations

$$\alpha_r \equiv E\{X\,[1 - F(X)]^r\},$$

and thus

$$\alpha_r = \int_{-\infty}^{\infty} x\,[1 - F(x)]^r f(x)\,\mathrm{d}x,$$

in terms of x, the PDF $f(x)$, and the CDF $F(x)$. Lastly, the α_r for the Kappa-Mu distribution with substitutions of the Marcum Q-function are

$$\alpha_r = \int_{-\infty}^{\infty} Q_\mu\left(\sqrt{2\kappa\mu},\ x\sqrt{2(1+\kappa)\mu}\right)^r x\,f(x)\,\mathrm{d}x.$$

Although multiple methods for Marcum Q-function computation are in cdfkmu and discussed in that documentation, the lmomkmu presenting is built only using the “chisq” approach.

Yacoub (2007, eq. 5) provides an expectation for the jth moment of the distribution as given by

$$\mathrm{E}(x^j) = \frac{\Gamma(\mu + j/2)\exp(-\kappa\mu)}{\Gamma(\mu)[(1+\kappa)\mu]^{j/2}} \times {}_1F_1(\mu + j/2; \mu; \kappa\mu),$$

where ${}_1F_1(a; b; z)$ is the confluent hypergeometric function of Abramowitz and Stegun (1972, eq. 13.1.2). The lmomkmu function optionally solves for the mean ($j = 1$) using the above equation in conjunction with the mean as computed by the order statistic minimums. The ${}_1F_1(a; b; z)$ is defined as

$${}_1F_1(a; b; z) = \sum_{i=0}^{\infty} \frac{a^{(i)}}{b^{(i)}} \frac{z^i}{n!},$$

where the notation $a^{(n)}$ represents “rising factorials” that are defined as $a^{(0)} = 1$ and $a^{(n)} = a(a+1)(a+2)\ldots(a+n-1)$. The rising factorials are readily computed by $a^{(n)} = \Gamma(n+1)/\Gamma(n)$ without resorting to a series computation. Yacoub (2007, eq. 5) is used to compute the mean.

Usage

```
lmomkmu(para, nmom=5, paracheck=TRUE, tol=1E-6, maxn=100)
```

Arguments

`para`	The parameters of the distribution.
`nmom`	The number of moments to compute.
`paracheck`	A logical controlling whether the parameters and checked for validity.
`tol`	An absolute tolerance term for series convergence of the confluent hypergeometric function when the Yacoub (2007) mean is to be computed.
`maxn`	The maximum number of interations in the series of the confluent hypergeometric function when the Yacoub (2007) mean is to be computed.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomkmu".
`yacoubsmean`	A list containing the mean, convergence error, and number of iterations in the series until convergence.

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81.

See Also

`parkmu`, `cdfkmu`, `pdfkmu`, `quakmu`

Examples

```
kmu <- vec2par(c(1.19,2.3), type="kmu")
lmomkmu(kmu)
## Not run:
par <- vec2par(c(1.67, .5), type="kmu")
lmomkmu(par)$lambdas
cdf2lmoms(par, nmom=4)$lambdas

system.time(lmomkmu(par))
system.time(cdf2lmoms(par, nmom=4))

## End(Not run)
# See the examples under lmomemu() so visualize L-moment
# relations on the L-skew and L-kurtosis diagram
```

`lmomkur` *L-moments of the Kumaraswamy Distribution*

Description

This function estimates the L-moments of the Kumaraswamy distribution given the parameters (α and β) from parkur. The L-moments in terms of the parameters with $\eta = 1 + 1/\alpha$ are

$$\lambda_1 = \beta B(\eta, \beta),$$

$$\lambda_2 = \beta[B(\eta, \beta) - 2B(\eta, 2\beta)],$$

$$\tau_3 = \frac{B(\eta, \beta) - 6B(\eta, 2\beta) + 6B(\eta, 3\beta)}{B(\eta, \beta) - 2B(\eta, 2\beta)},$$

$$\tau_4 = \frac{B(\eta, \beta) - 12B(\eta, 2\beta) + 30B(\eta, 3\beta) - 40B(\eta, 4\beta)}{B(\eta, \beta) - 2B(\eta, 2\beta)}, \text{ and}$$

$$\tau_5 = \frac{B(\eta, \beta) - 20B(\eta, 2\beta) + 90B(\eta, 3\beta) - 140B(\eta, 4\beta) + 70B(\eta, 5\beta)}{B(\eta, \beta) - 2B(\eta, 2\beta)}.$$

where $B(a, b)$ is the complete beta function or `beta()`.

Usage

```
lmomkur(para)
```

Arguments

`para` The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomkur".

Author(s)

W.H. Asquith

References

Jones, M.C., 2009, Kumaraswamy's distribution—A beta-type distribution with some tractability advantages: Statistical Methodology, v. 6, pp. 70–81.

See Also

parkur, cdfkur, pdfkur, quakur

Examples

```
lmr <- lmoms(c(0.25, 0.4, 0.6, 0.65, 0.67, 0.9))
lmomkur(parkur(lmr))
## Not run:
A <- B <- exp(seq(-3,5, by=.05))
logA <- logB <- T3 <- T4 <- c();
i <- 0
for(a in A) {
  for(b in B) {
    i <- i + 1
    parkur <- list(para=c(a,b), type="kur");
    lmr <- lmomkur(parkur)
    logA[i] <- log(a); logB[i] <- log(b)
    T3[i] <- lmr$ratios[3]; T4[i] <- lmr$ratios[4]
  }
}
library(lattice)
contourplot(T3~logA+logB, cuts=20, lwd=0.5, label.style="align",
            xlab="LOG OF ALPHA", ylab="LOG OF BETA",
            xlim=c(-3,5), ylim=c(-3,5),
            main="L-SKEW FOR KUMARASWAMY DISTRIBUTION")
contourplot(T4~logA+logB, cuts=10, lwd=0.5, label.style="align",
            xlab="LOG OF ALPHA", ylab="LOG OF BETA",
            xlim=c(-3,5), ylim=c(-3,5),
```

```
            main="L-KURTOSIS FOR KUMARASWAMY DISTRIBUTION")

## End(Not run)
```

---

lmomlap *L-moments of the Laplace Distribution*

Description

This function estimates the L-moments of the Laplace distribution given the parameters (ξ and α) from parlap. The L-moments in terms of the parameters are $\lambda_1 = \xi$, $\lambda_2 = 3\alpha/4$, $\tau_3 = 0$, $\tau_4 = 17/22$, $\tau_5 = 0$, and $\tau_6 = 31/360$.

For r odd and $r \geq 3$, $\lambda_r = 0$, and for r even and $r \geq 4$, the L-moments using the hypergeometric function ${}_2F_1()$ are

$$\lambda_r = \frac{2\alpha}{r(r-1)}[1 - {}_2F_1(-r, r-1, 1, 1/2)],$$

where ${}_2F_1(a, b, c, z)$ is defined as

$${}_2F_1(a, b, c, z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!},$$

where $(x)_n$ is the *rising* Pochhammer symbol, which is defined by

$$(x)_n = 1 \text{ for } n = 0 \text{, and}$$

$$(x)_n = x(x+1)\cdots(x+n-1) \text{ for } n > 0.$$

Usage

```
lmomlap(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomlap".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: IBM Research Report RC12210, T.J. Watson Research Center, Yorktown Heights, New York.

See Also

parlap, cdflap, pdflap, qualap

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomlap(parlap(lmr))
```

---

lmomlmrq — *L-moments of the Linear Mean Residual Quantile Function Distribution*

Description

This function estimates the L-moments of the Linear Mean Residual Quantile Function distribution given the parameters (μ and α) from parlmrq. The first six L-moments in terms of the parameters are

$$\lambda_1 = \mu,$$

$$\lambda_2 = (\alpha + 3\mu)/6,$$

$$\lambda_3 = 0,$$

$$\lambda_4 = (\alpha + \mu)/12,$$

$$\lambda_5 = (\alpha + \mu)/20, \text{ and}$$

$$\lambda_6 = (\alpha + \mu)/30.$$

Because $\alpha + \mu > 0$, then $\tau_3 > 0$, so the distribution is positively skewed. The coefficient of L-variation is in the interval $(1/3, 2/3)$.

Usage

```
lmomlmrq(para)
```

Arguments

para	The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomlmrq".

Author(s)

W.H. Asquith

References

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

See Also

`parlmrq`, `cdflmrq`, `pdflmrq`, `qualmrq`

Examples

```
lmr <- lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2))
lmr
lmomlmrq(parlmrq(lmr))
```

`lmomln3` *L-moments of the 3-Parameter Log-Normal Distribution*

Description

This function estimates the L-moments of the Log-Normal3 distribution given the parameters (ζ, lower bounds; μ_{log}, location; and σ_{log}, scale) from `parln3`. The distribution is the same as the Generalized Normal with algebraic manipulation of the parameters, and **lmomco** does not have truly separate algorithms for the Log-Normal3 but uses those of the Generalized Normal. The discussion begins with the later distribution.

The two L-moments in terms of the Generalized Normal distribution parameters (`lmomgno`) are

$$\lambda_1 = \xi + \frac{\alpha}{\kappa}[1 - \exp(\kappa^2/2)], \text{ and}$$

$$\lambda_2 = \frac{\alpha}{\kappa}(\exp(\kappa^2/2)(1 - 2\Phi(-\kappa/\sqrt{2})),$$

where Φ is the cumulative distribution of the Standard Normal distribution. There are no simple expressions for τ_3, τ_4, and τ_5, and numerical methods are used.

Let ζ be the lower bounds (real space) for which $\zeta < \lambda_1 - \lambda_2$ (checked in `are.parln3.valid`), μ_{log} be the mean in natural logarithmic space, and σ_{log} be the standard deviation in natural logarithm space for which $\sigma_{\mathrm{log}} > 0$ (checked in `are.parln3.valid`) is obvious because this parameter has an analogy to the second product moment. Letting $\eta = \exp(\mu_{\mathrm{log}})$, the parameters of the Generalized Normal are $\zeta + \eta$, $\alpha = \eta\sigma_{\mathrm{log}}$, and $\kappa = -\sigma_{\mathrm{log}}$. At this point the L-moments can be solved for using algorithms for the Generalized Normal.

Usage

```
lmomln3(para)
```

Arguments

`para`	The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: “lmomln3”.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`parln3`, `cdfln3`, `pdfln3`, `qualn3`, `lmomgno`

Examples

```
X <- exp(rnorm(10))
pargno(lmoms(X))$para
parln3(lmoms(X))$para
```

lmomnor — *L-moments of the Normal Distribution*

Description

This function estimates the L-moments of the Normal distribution given the parameters (μ and σ) from parnor. The L-moments in terms of the parameters are $\lambda_1 = \mu$, $\lambda_2 = \sigma/\sqrt{pi}$, $\tau_3 = 0$, $\tau_4 = 0.122602$, and $\tau_5 = 0$.

Usage

```
lmomnor(para)
```

Arguments

para	The parameters of the distribution.

Value

An R `list` is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is `0`.
leftrim	Level of left-tail trimming used in the computation, which is `NULL`.
rightrim	Level of right-tail trimming used in the computation, which is `NULL`.
source	An attribute identifying the computational source of the L-moments: "lmomnor".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parnor, cdfnor, pdfnor, quanor

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
lmr
lmomnor(parnor(lmr))
```

---

lmompdq3 *L-moments of the Polynomial Density-Quantile3 Distribution*

Description

This function estimates the L-moments of the Polynomial Density-Quantile3 distribution given the parameters (ξ, α, and κ) from parpdq3. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \alpha\bigl[(1+\kappa)\log(1+\kappa) - (1-\kappa)\log(1-\kappa) - \kappa\log(4)\bigr],$$

$$\lambda_2 = \frac{\alpha(1-\kappa^2)}{(1-\kappa\tau_3)},$$

$$\tau_3 = \frac{1}{\kappa} - \frac{1}{\mathrm{arctanh}(\kappa)}, \text{ and}$$

$$\tau_4 = (5\tau_3/\kappa) - 1.$$

Usage

```
lmompdq3(para, paracheck=TRUE)
```

Arguments

para	The parameters of the distribution.
paracheck	A logical switch as to whether the validity of the parameters should be checked. Default is paracheck=TRUE.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: “lmompdq3”.

Note

Polynomial approximations for the τ_3 and τ_4 are developed here. First, the author's monograph (Asquith, 2011, table 10.1) shows five digits for such approximates for other distributions, so the code below will used the core basis, five digits. Second, an approximation means that `lmrdia` does not have the internal burden of using `uniroot()` to solve for the coordinates for the L-moment ratio diagram. The following code represents an exploration towards the definition of a helper function, `t4pdq3()`, which is repeated inside the internals of `lmrdia` in order to support the PDQ3. The trajectory of the PDQ3 resides at or above that for the generalized logistic distribution (`quaglo`) that is well known to L-moment theory. In conclusion, the 5-digit approximation provides a maximum absolute τ_4 error of about 0.00055.

```
fn <- function(k, tau3=NA) { t3 <- (1/k - 1/atanh(k))
                             if(is.nan(t3)) t3 <- 0
                             return(t3-tau3) }
t3s <- seq(-1, 1, by=0.005)
t4s <- NULL
for(t3 in t3s) {
  rt  <- uniroot(fn, interval=c(-1,1), tau3=t3)
  t4  <- ((5*t3 / rt$root) - 1) / 4 # Hosking (2007)
  t4s <- c(t4s, t4)
}
t4s[is.nan(t4s)] <- 1/6 # by distribution properties

plotlmrdia(lmrdia())
points(t3s, t4s, pch=21, cex=0.5, bg=8, col="lightgreen")
lines( t3s, t4s, col="darkgreen") # above GLO and see Hosking (2007, fig. 1)

# eight powers as in Hosking and Wallis (1997) coefficient table for
# many other distributions
pdq3 <- stats::lm(t4s~I(t3s^1)+I(t3s^2)+I(t3s^3)+I(t3s^4)+
                      I(t3s^5)+I(t3s^6)+I(t3s^7)+I(t3s^8))
lines(t3s, fitted.values(pdq3), lwd=2, col=grey(0.8))
pdq3$coefficients # Ah, see the odd coefficients are near zero, so define
# as such in a repeated linear model but with skips on the odd orders:
pdq3 <- stats::lm(t4s~I(t3s^2)+I(t3s^4)+I(t3s^6)+I(t3s^8))
lines(t3s, fitted.values(pdq3), lwd=1, col="red")
max(abs(t4s - fitted.values(pdq3))) # show the max error in Rs resolution

# we desire to compare "full resolution" to 5-digit truncation
print(pdq3$coefficients,       16)  # in the 5 in the next line, c.2022,
# we can  make new column in Asquith (2011, table 10.1) if ever needed for
print(round(pdq3$coefficients,  5)) # a second edition

t4pdq3 <- function(t3, use5digits=TRUE) { # helper to repeat within lmrdia()
  c05 <- c( 0.16688, 0, 0.98951, 0, -0.00526, 0, -0.24074, 0, 0.08906)
  c16 <- c( 0.166875136751297809, 0,  0.989506002306983601, 0,
           -0.005255434641059076, 0, -0.240744479052170501, 0,
            0.089060315246257210)
```

```
    ifelse(use5digits, myc <- c05, myc <- c16)
    t4 <- vector(mode="numeric", length(t3))
    for(i in 1:length(t3)) {
      t4[i] <- sum(sapply(2:length(myc), function(k) myc[k]*t3[i]^(k-1)))
    }
    return(t4 + myc[1]) # end with the intercept being added on
  }
  lines(t3s, t4pdq3(t3s), col="darkgreen", lty=2)
  summary(abs(t4s - t4pdq3(t3s, use5digits=TRUE )))
  summary(abs(t4s - t4pdq3(t3s, use5digits=FALSE)))
  max(    abs(t4s - t4pdq3(t3s, use5digits=TRUE )))
  max(    abs(t4s - t4pdq3(t3s, use5digits=FALSE)))
  # further comparisons as needed to understand the aforementioned operations
  plot(  t4s, t4s - t4pdq3(t3s, use5digits=TRUE ), col="red", type="l")
  lines( t4s, t4s - t4pdq3(t3s, use5digits=FALSE), col="blue")
  abline(h=0)
```

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parpdq3, cdfpdq3, pdfpdq3, quapdq3

Examples

```
## Not run:
  para <- list(para=c(20, 1, -0.5), type="pdq3")
  lmoms(quapdq3(runif(100000), para))$lambdas
  lmompdq3(para)$lambdas #
## End(Not run)

## Not run:
  para <- list(para=c(20, 1, +0.5), type="pdq3")
  lmoms(quapdq3(runif(100000), para))$lambdas
  lmompdq3(para)$lambdas #
## End(Not run)
```

`lmompdq4` *L-moments of the Polynomial Density-Quantile4 Distribution*

Description

This function estimates the L-moments of the Polynomial Density-Quantile4 distribution given the parameters (ξ, α, and κ) from `parpdq4`. The L-moments in terms of the parameters are

$$\lambda_1 = \xi,$$

$$\lambda_2 = \frac{\alpha}{\kappa}\left(1-\kappa^2\right)\mathrm{atanh}(\kappa) \text{ for } \kappa > 0,$$

$$\lambda_2 = \frac{\alpha}{\kappa}\left(1+\kappa^2\right)\mathrm{atan}(\kappa) \text{ for } \kappa < 0,$$

$$\tau_3 = 0, \text{ and}$$

$$\tau_4 = -\frac{1}{4} + \frac{5}{4\kappa}\left(\frac{1}{\kappa} - \frac{1}{\mathrm{atanh}(\kappa)}\right) \text{ for } \kappa > 0,$$

$$\tau_4 = -\frac{1}{4} - \frac{5}{4\kappa}\left(\frac{1}{\kappa} - \frac{1}{\mathrm{atan}(\kappa)}\right) \text{ for } \kappa < 0,$$

Usage

```
lmompdq4(para, paracheck=TRUE)
```

Arguments

`para`	The parameters of the distribution.
`paracheck`	A logical switch as to whether the validity of the parameters should be checked. Default is `paracheck=TRUE`.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`ifail`	A numeric field connected to the `ifailtext`; a value of 0 indicates fully successful operation of the function.
`ifailtext`	A message, instead of a warning, about the internal operations or operational limits of the function.
`source`	An attribute identifying the computational source of the L-moments: “lmompdq4”.

Note

What L-kurtosis produces the widest 95th-percentile bounds?—Study of the shapes of the PDQ4 will show that with support for τ_4 much less and even negative and much more than the $\tau_4 = 0.122602$ defined into the Normal distribution considerable variation. The widths or spreads between quantiles moderately deep into the tails might be interesting to study. Consider the code that follows that seeks the τ_4 that will produce the widest 95th-percentile bounds:

```
ofunc <- function(t4,  lscale=NA) {
  lmr <- vec2lmom(c(0, lscale, 0, t4))
  if(! are.lmom.valid(lmr)) return(-Inf)
  pdq4  <- lmomco::parpdq4(lmr, snapt4uplimit=FALSE)
  return(-diff(lmomco::quapdq4(c(0.025, 0.975), pdq4)))
}
optim(0.2, ofunc, lscale=1)$par # [1] 0.4079688
```

The code maximizes at about $\tau_4 = 0.4079688$. It is informative to visualizing the nature of the objective function. In the code below, we standardize the width by division of the $\lambda_2 = 1$ for generality and because of symmetry only the 97.5th percentile requires study:

```
lscale <- 1
tau4s  <- seq(-1/4, 0.9, by=0.01)
qua975s <- rep(NA, length(tau4s))
for(i in 1:length(tau4s)) {
  lmr <- vec2lmom(c(0, lscale, 0, tau4s[i]))
  if(! are.lmom.valid(lmr)) next
  pdq4 <- lmomco::parpdq4(lmr, snapt4uplimit=FALSE)
  quas <- lmomco::quapdq4(c(0.025, 0.975), pdq4)
  qua975s[i] <- quas[2] / lscale
}
plot(tau4s, qua975s, ylim=c(-0.1, 5), col="blue")
abline(v=0.845, lty=2) # supporting the "snaptau4uplimit" in parpdq4().
abline(v=0.4079688, col=2, lwd=2)
abline(h=qnorm(0.975, sd=sqrt(pi)), col="green", lty=3, lwd=3)
```

The figure so produces shows that the maximum at the red vertical line for τ_4 is at the crest of the blue points. The figure shows that for $\tau_4 >= 0.845$ that numerical problems manifest and contribute to an snapping limit of τ_4 in parpdq4. The figure also shows with a dotted green line that the equivalent percentile of the Normal distribution with a standard deviation equivalent to the $\lambda_2 = 1$ has two intersections on the widths of the PDQ4.

Now some further experiments on the apparent computational limits to τ_4 can be made using the code that follows. This support the threshold of $\tau_4 \leq 0.845$ embedded into parpdq4 through the use of the theoTLmoms function.

```
t4s <- seq(-1/4, 1, by=0.02)
t4s <- t4s[t4s > -1/4 & t4s < 1]
l2s_theo <- t4s_theo <- t6s_theo <- rep(NA, length(t4s))
for(i in 1:length(t4s)) {
  lmr  <- vec2lmom(c(0, 1, 0, t4s[i]))
```

```
    suppressWarnings(par <- parpdq4(lmr, snapt4uplimit=FALSE))
    tlmr <- theoTLmoms(par, nmom=6, trim=0)
    l2s_theo[i] <- tlmr$lambdas[2]
    t4s_theo[i] <- tlmr$ratios[ 4]
    t6s_theo[i] <- tlmr$ratios[ 6]
  }
  plot(  t4s_theo, l2s_theo, type="l")
  points(t4s_theo, l2s_theo)
    abline(v=0.864, lty=2) # see "snaptau4uplimit" in parpdq4()
    abline(v=0.845, lty=2) # see "snaptau4uplimit" in parpdq4()
  plot(  t4s_theo, t4s,      type="l")
  points(t4s_theo, t4s)
    abline(v=0.864, lty=2) # see "snaptau4uplimit" in parpdq4()
    abline(v=0.845, lty=2) # see "snaptau4uplimit" in parpdq4()
  plot(  t4s_theo, t6s_theo, type="l")
  points(t4s_theo, t6s_theo)
    abline(v=0.864, lty=2) # see "snaptau4uplimit" in parpdq4()
    abline(v=0.845, lty=2) # see "snaptau4uplimit" in parpdq4()
```

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

parpdq4, cdfpdq4, pdfpdq4, quapdq4

Examples

```
para <- vec2par(c(0, 1, -100), type="pdq4")
lmompdq4(  para)$ratios[4]                # -0.2421163
theoTLmoms(para, nmom=6, trim=0)$ratios[4] # -0.2421163
theoTLmoms(para, nmom=6, trim=1)$ratios[4] # -0.2022106
theoTLmoms(para, nmom=6, trim=2)$ratios[4] # -0.1697186

## Not run:
  para <- list(para=c(20, 1, -0.5), type="pdq4")
  lmoms(quapdq4(runif(100000), para))$lambdas
  lmompdq4(para)$lambdas #
## End(Not run)

## Not run:
  para <- list(para=c(20, 1, +0.5), type="pdq4")
  lmoms(quapdq4(runif(100000), para))$lambdas
  lmompdq4(para)$lambdas #
```

```
## End(Not run)

## Not run:
  K1 <- seq(-5, 0, by=0.001)
  K2 <- seq( 0, 1, by=0.001)
  suppressWarnings(mono_decrease_part1 <- -(1/4) + (5/(4*K1)) * (1/K1 - 1/atanh(K1)))
                   mono_increase_part2 <- -(1/4) - (5/(4*K1)) * (1/K1 - 1/atan( K1))
                   mono_increase_part1 <- -(1/4) + (5/(4*K2)) * (1/K2 - 1/atanh(K2))
                   mono_decrease_part2 <- -(1/4) - (5/(4*K2)) * (1/K2 - 1/atan( K2))

  plot( 0, 0, type="n", xlim=range(c(K1, K2)), ylim=c(-0.25, 1),
       xlab="Kappa shape parameter PDQ4 distribution", ylab="L-kurtosis (Tau4)")
  lines(K1, mono_decrease_part1, col=4, lwd=0.3)
  lines(K2, mono_increase_part1, col=4, lwd=3)
  lines(K2, mono_decrease_part2, col=2, lwd=0.3)
  lines(K1, mono_increase_part2, col=2, lwd=3)

  abline(h= 1/6, lty=2, lwd=0.6)
  abline(h=-1/4, lty=2, lwd=0.6)
  text(-5, -1/4, "Tau4 lower bounds", pos=4, cex=0.8)
  abline(v=0,    lty=2, lwd=0.6)
  abline(v=1,    lty=1, lwd=0.9)
  points(-0.7029, 0.1226, pch=15, col="darkgreen")

  # bigTAU4 <- 0.845 # see parpdq4.R and parpdq4.Rd
  pdq4 <- parpdq4(vec2lmom(c(0, 1, 0, 0.845)), snapt4uplimit=FALSE)
  points(pdq4$para[3], 0.845, cex=1.5, pch=17, col="blue")

  legend("topleft", c("Monotonic increasing for kappa < 0 (used for PDQ4)",
                      "Monotonic increasing for kappa > 0 (used for PDQ4)",
                      "Monotonic decreasing for kappa > 0 (not used for PDQ4)",
                      "Monotonic decreasing for kappa < 0 (not used for PDQ4)",
                      "Normal distribution (Tau4=0.122602 by definition)",
                   "Operational upper limit of Tau4 before numerical problems"), cex=0.8,
     pch=c(NA, NA, NA, NA, 15, 17), lwd=c(3,3, 0.3, 0.3, NA, NA),
     pt.cex=c(NA, NA, NA, NA, 1, 1.5), col=c(2, 4, 2, 4, "darkgreen", "blue")) #
## End(Not run)
```

---

lmompe3 *L-moments of the Pearson Type III Distribution*

Description

This function estimates the L-moments of the Pearson Type III distribution given the parameters (μ, σ, and γ) from parpe3 as the product moments: mean, standard deviation, and skew. The first three L-moments in terms of these parameters are complex and numerical methods are required. For simplier expression of the distribution functions (cdfpe3, pdfpe3, and quape3) the "moment parameters" are expressed differently.

The Pearson Type III distribution is of considerable theoretical interest because the parameters, which are estimated via the L-moments, are in fact the product moments. Although, these values fitted by the method of L-moments will not be numerically equal to the sample product moments. Further details are provided in the Examples section of the pmoms function documentation.

Usage

```
lmompe3(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmompe3".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parpe3, cdfpe3, pdfpe3, quape3

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmompe3(parpe3(lmr))
```

lmomray *L-moments of the Rayleigh Distribution*

Description

This function estimates the L-moments of the Rayleigh distribution given the parameters (ξ and α) from parray. The L-moments in terms of the parameters are

$$\lambda_1 = \xi + \alpha\sqrt{\pi/2},$$

$$\lambda_2 = \frac{1}{2}\alpha(\sqrt{2}-1)\sqrt{\pi},$$

$$\tau_3 = \frac{1 - 3/\sqrt{2} + 2/\sqrt{3}}{1 - 1/\sqrt{2}} = 0.1140, \text{ and}$$

$$\tau_4 = \frac{1 - 6/\sqrt{2} + 10/\sqrt{3} - 5\sqrt{4}}{1 - 1/\sqrt{2}} = 0.1054.$$

Usage

```
lmomray(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomray".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

See Also

parray, cdfray, pdfray, quaray

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomray(parray(lmr))
```

`lmomRCmark` *Sample L-moment for Right-Tail Censoring by a Marking Variable*

Description

Compute the sample L-moments for right-tail censored data set in which censored data values are identified by a marking variable.

Usage

```
lmomRCmark(x, rcmark=NULL, r=1, sort=TRUE)
```

Arguments

`x`	A vector of data values.
`rcmark`	The right-tail censoring (upper) marking variable for unknown threshold: 1 is uncensored, 0 is censored.
`r`	The L-moment order to return, default is the mean.
`sort`	Do the data need sorting? The availability of this option is to avoid unnecessary overhead of sorting on each call to this function by the primary higher-level function `lmomsRCmark`.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is $\hat{\lambda}_1^{(0,0)}$, second element is $\hat{\lambda}_2^{(0,0)}$, and so on.
`ratios`	Vector of the L-moment ratios. Second element is $\hat{\tau}^{(0,0)}$, third element is $\hat{\tau}_3^{(0,0)}$ and so on.
`trim`	Level of symmetrical trimming used in the computation, which will equal `NULL` if asymmetrical trimming was used.
`leftrim`	Level of left-tail trimming used in the computation.
`rightrim`	Level of right-tail trimming used in the computation.
`source`	An attribute identifying the computational source of the L-moments: “lmomsRCmark”.

Author(s)

W.H. Asquith

References

Wang, Dongliang, Hutson, A.D., Miecznikowski, J.C., 2010, L-moment estimation for parametric survival models given censored data: Statistical Methodology, v. 7, no. 6, pp. 655–667.

See Also

lmomsRCmark

Examples

```
# See example under lmomsRCmark
```

---

lmomrevgum *L-moments of the Reverse Gumbel Distribution*

Description

This function estimates the L-moments of the Reverse Gumbel distribution given the parameters (ξ and α) from parrevgum. The first two type-B L-moments in terms of the parameters are

$$\lambda^B_1 = \xi - (0.5722\dots)\alpha - \alpha\{\mathrm{Ei}(-\log(1-\zeta))\}\text{and}$$

$$\lambda^B_2 = \alpha\{\log(2) + \mathrm{Ei}(-2\log(1-\zeta)) - \mathrm{Ei}(-\log(1-\zeta))\},$$

where ζ is the right-tail censoring fraction of the sample or the nonexceedance probability of the right-tail censoring threshold, and $\mathrm{Ei}(x)$ is the exponential integral defined as

$$\mathrm{Ei}(X) = \int_X^\infty x^{-1}\exp(-x)\mathrm{d}x,$$

where $\mathrm{Ei}(-\log(1-\zeta)) \rightarrow 0$ as $\zeta \rightarrow 1$ and $\mathrm{Ei}(-\log(1-\zeta))$ can not be evaluated as $\zeta \rightarrow 0$.

Usage

```
lmomrevgum(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
zeta	Number of samples observed (noncensored) divided by the total number of samples.
source	An attribute identifying the computational source of the L-moments: "lmomrevgum".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

parrevgum, cdfrevgum, pdfrevgum, quarevgum

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
rev.para <- lmom2par(lmr,type='revgum')
lmomrevgum(rev.para)
```

---

lmomrice *L-moments of the Rice Distribution*

Description

This function estimates the L-moments of the Rice distribution given the parameters (ν and α) from parrice. The L-moments in terms of the parameters are complex. They are computed here by the system of maximum order statistic expectations from theoLmoms.max.ostat, which uses expect.max.ostat. The connection between τ_2 and ν/α and a special function (the Laguerre polynomial, LaguerreHalf) of ν^2/α^2 and additional algebraic terms is tabulated in the R data.frame located within .lmomcohash$RiceTable. The file 'SysDataBuilder01.R' provides additional details.

Usage

```
lmomrice(para, ...)
```

Arguments

para	The parameters of the distribution.
...	Additional arguments passed to theoLmoms.max.ostat.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomrice", but the exact contents of the remainder of the string might vary as limiting distributions of Normal and Rayleigh can be involved for $\nu/\alpha > 52$ (super high SNR, Normal) or $24 < \nu/\alpha \leq 52$ (high SNR, Normal) or $\nu/\alpha < 0.08$ (very low SNR, Rayleigh).

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

parrice, cdfrice, cdfrice, quarice

Examples

```
## Not run:
lmomrice(vec2par(c(65,34), type="rice"))

# Use the additional arguments to show how to avoid unnecessary overhead
# when using the Rice, which only has two parameters.
  rice <- vec2par(c(15,14), type="rice")
  system.time(lmomrice(rice, nmom=2)); system.time(lmomrice(rice, nmom=6))

  lcvs <- vector(mode="numeric"); i <- 0
```

```
  SNR  <- c(seq(7,0.25, by=-0.25), 0.1)
  for(snr in SNR) {
    i <- i + 1
    rice    <- vec2par(c(10,10/snr), type="rice")
    lcvs[i] <- lmomrice(rice, nmom=2)$ratios[2]
  }
  plot(lcvs, SNR,
       xlab="COEFFICIENT OF L-VARIATION",
       ylab="LOCAL SIGNAL TO NOISE RATIO (NU/ALPHA)")
  lines(.lmomcohash$RiceTable$LCV,
        .lmomcohash$RiceTable$SNR)
  abline(1,0, lty=2)
  mtext("Rice Distribution")
  text(0.15,0.5, "More noise than signal")
  text(0.15,1.5, "More signal than noise")

## End(Not run)
## Not run:
# A polynomial expression for the relation between L-skew and
# L-kurtosis for the Rice distribution can be readily constructed.
T3 <- .lmomcohash$RiceTable$TAU3
T4 <- .lmomcohash$RiceTable$TAU4
LM <- lm(T4~T3+I(T3^2)+I(T3^3)+I(T3^4)+
               I(T3^5)+I(T3^6)+I(T3^7)+I(T3^8))
summary(LM) # note shown
## End(Not run)
```

---

lmoms *The Sample L-moments and L-moment Ratios*

Description

Compute the sample L-moments. The mathematical expression for sample L-moment computation is shown under TLmoms. The formula jointly handles sample L-moment computation and sample TL-moment (Elamir and Seheult, 2003) computation. A description of the most common L-moments is provided under lmom.ub.

Usage

```
lmoms(x, nmom=5, no.stop=FALSE, vecit=FALSE)
```

Arguments

- `x` A vector of data values.
- `nmom` The number of moments to compute. Default is 5.
- `no.stop` A logical to return NULL instead of issuing a stop() if nmom is greater than the sample size or if all the values are equal. This is a very late change (decade+) to the foundational function in the package. Auxiliary coding to above this function to avoid the internal stop() became non-ignorable in large data mining

exercises. It was a design mistake to have the `stop()` and not a `warning()` instead.

`vecit` A logical to return the first two $\lambda_i \in 1, 2$ and then the $\tau_i \in 3, \cdots$ where the length of the returned vector is controlled by the `nmom` argument. This argument will store the trims (see `TLmoms`) as `NULL` used (see the **Example** that follows).

Value

An R `list` is returned.

`lambdas` Vector of the L-moments. First element is $\hat{\lambda}_1^{(0,0)}$, second element is $\hat{\lambda}_2^{(0,0)}$, and so on.

`ratios` Vector of the L-moment ratios. Second element is $\hat{\tau}^{(0,0)}$, third element is $\hat{\tau}_3^{(0,0)}$ and so on.

`trim` Level of symmetrical trimming used in the computation, which will equal `NULL` if asymmetrical trimming was used.

`leftrim` Level of left-tail trimming used in the computation.

`rightrim` Level of right-tail trimming used in the computation.

`source` An attribute identifying the computational source of the L-moments: "lmoms".

Note

This function computes the L-moments through the generalization of the TL-moments (`TLmoms`). In fact, this function calls the default TL-moments with no trimming of the sample. This function is equivalent to `lmom.ub`, but returns a different data structure. The `lmoms` function is preferred by the author.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational statistics and data analysis, vol. 43, pp. 299-314.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

`lmom.ub`, `TLmoms`, `lmorph`, `lmoms.bernstein`, `vec2lmom`

Examples

```
lmoms(rnorm(30),nmom=4)

vec2lmom(lmoms(rexp(30), nmom=3, vecit=TRUE)) # re-vector
```

`lmoms.bernstein` *Numerically Integrated L-moments of Smoothed Quantiles from Bernstein or Kantorovich Polynomials*

Description

Compute the L-moment by numerical integration of the smoothed quantiles from Bernstein or Kantorovich polynomials (see dat2bernqua). Letting $\tilde{X}_n(F)$ be the smoothed quantile function for nonexceedance probability F for a sample of size n, from Asquith (2011) the first five L-moments in terms of quantile function integration are

$$\lambda_1 = \int_0^1 \tilde{X}_n(F)\,\mathrm{d}F,$$

$$\lambda_2 = \int_0^1 \tilde{X}_n(F) \times (2F-1)\,\mathrm{d}F,$$

$$\lambda_3 = \int_0^1 \tilde{X}_n(F) \times (6F^2-6F+1)\,\mathrm{d}F,$$

$$\lambda_4 = \int_0^1 \tilde{X}_n(F) \times (20F^3-30F^2+12F-1)\,\mathrm{d}F, \text{ and}$$

$$\lambda_5 = \int_0^1 \tilde{X}_n(F) \times (70F^4-140F^3+90F^2-20F+1)\,\mathrm{d}F.$$

Usage

```
lmoms.bernstein(x, bern.control=NULL,
                   poly.type=c("Bernstein", "Kantorovich", "Cheng"),
                   bound.type=c("none", "sd", "Carv", "either"),
                   fix.lower=NULL, fix.upper=NULL, p=0.05)
```

Arguments

`x`	A vector of data values.
`bern.control`	A `list` that holds `poly.type`, `bound.type`, `fix.lower`, and `fix.upper`. And this list will supersede the respective values provided as separate arguments.
`poly.type`	Same argument as for dat2bernqua.
`bound.type`	Same argument as for dat2bernqua.
`fix.lower`	Same argument as for dat2bernqua.
`fix.upper`	Same argument as for dat2bernqua.
`p`	The "p-factor" is the same argument as for dat2bernqua.

Value

An R `vector` is returned.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

dat2bernqua, pfactor.bernstein, lmoms

Examples

```
## Not run:
X <- exp(rnorm(100))
lmoms.bernstein(X)$ratios
lmoms.bernstein(X, fix.lower=0)$ratios
lmoms.bernstein(X, fix.lower=0, bound.type="sd")$ratios
lmoms.bernstein(X, fix.lower=0, bound.type="Carv")$ratios
lmoms(X)$ratios

lmoms.bernstein(X, poly.type="Kantorovich")$ratios
lmoms.bernstein(X, fix.lower=0, poly.type="Kantorovich")$ratios
lmoms.bernstein(X, fix.lower=0, bound.type="sd", poly.type="Kantorovich")$ratios
lmoms.bernstein(X, fix.lower=0, bound.type="Carv", poly.type="Kantorovich")$ratios
lmoms(X)$ratios

## End(Not run)

## Not run:
lmr <- vec2lmom(c(1,.2,.3))
par <- lmom2par(lmr, type="gev")
lmr <- lmorph(par2lmom(par))
lmT <- c(lmr$lambdas[1:2], lmr$ratios[3:5])
ns  <- 200; nsim <- 1000; empty <- rep(NA, nsim)

sink("ChengLmomentTest.txt")
cat(c("N errmeanA  errlscaleA  errtau3A  errtau4A  errtau5A",
        "errmeanB  errlscaleB  errtau3B  errtau4B  errtau5B\n"))
for(n in 1:ns) {
   message(n);
   SIM <- data.frame(errmeanA=empty, errlscaleA=empty,   errtau3A=empty, errtau4A=empty,
                     errtau5A=empty,   errmeanB=empty, errlscaleB=empty, errtau3B=empty,
                     errtau4B=empty,   errtau5B=empty)
   for(i in 1:nsim) {
      X <- rlmomco(30, par)
      lmrA <- lmoms(X)
      lmA <- c(lmrA$lambdas[1:2], lmrA$ratios[3:5])
      lmrB <- lmoms.bernstein(X, poly.type="Cheng")
      lmB <- c(lmrB$lambdas[1:2], lmrB$ratios[3:5])
      EA <- lmA - lmT; EB <- lmB - lmT
```

```
        SIM[i,] <- c(EA,EB)
    }
    MeanErr <- sapply(1:length(SIM[1,]), function(x) { return(mean(SIM[,x])) })
    line <- paste(c(n, round(MeanErr, digits=6), "\n"), sep=" ")
    cat(line)
}
sink()

## End(Not run)
```

`lmoms.bootbarvar` *Exact Bootstrap Mean and Variance of L-moments*

Description

This function computes the exact bootstrap mean and variance of L-moments using the exact analytical expressions for the bootstrap mean and variance of any L-estimator described by Hutson and Ernst (2000). The approach by those authors is to use the bootstrap distribution of the single order statistic in conjunction with the joint distribution of two order statistics. The key component is the bootstrap mean vector as well as the variance-covariance matrix of all the order statistics and then performing specific linear combinations of a basic L-estimator combined with the proportion weights used in the computation of L-moments (`Lcomoment.Wk`, see those examples and division by n). Reasonably complex algorithms are used; however, what makes those authors' contribution so interesting is that neither simulation, resampling, or numerical methods are needed as long as the sample size is not too large.

This function provides a uniquely independent method to compute the L-moments of a sample from the vector of exact bootstrap order statistics. It is anticipated that several of the intermediate computations of this function would be of interest in further computations or graphical visualization. Therefore, this function returns many more numerical values than other L-moment functions of **lmomco**. The variance-covariance matrix for large samples requires considerable CPU time; as the matrix is filled, status output is generated.

The example section of this function contains the verification of the implementation as well as provides to additional computations of variance through resampling with replacement and simulation from the parent distribution that generated the sample vector shown in the example.

Usage

```
lmoms.bootbarvar(x, nmom=6, covarinverse=TRUE, verbose=TRUE,
                    force.exact=FALSE, nohatSIGMA=FALSE, nsim=500, bign=40, ...)
```

Arguments

`x`	A vector of data values.
`nmom`	The number of moments to compute. Default is 6 and can not be less than 3.
`covarinverse`	Logical on computation of the matrix inversions: `inverse.varcovar.tau23`, `inverse.varcovar.tau34`, and `inverse.varcovar.tau46`.

`verbose`	A logical switch on the verbosity of the construction of the variance-covariance matrix of the order statisitics. This operation is the most time consuming of those inside the function and is provided at default of `verbose=TRUE` to make a general user comfortable.
`force.exact`	A logical switch to attempt a *forced exact bootstrap computation* (empirical bootstrap controlled by `nsim` thus is *not* used) even if the sample size is too large as controlled by `bign`. See messages during the execution for guidance.
`nohatSIGMA`	A logical to bypass most of the interesting matrix functions and results. If `TRUE`, then only `lambdas`, `ratios`, and `bootstrap.orderstatistics` are populated. This feature is useful if a user is only interested in get the bootstrap estimates of the order statistics.
`nsim`	Simulation size in case simulations and not the exact bootstrap are used.
`bign`	A sample size threshold that triggers simulation using `nsim` replications for estimation by empirical bootstrap. Some of the "exact" operations are extremely expensive and numerical problems in the matrices are known for non-normal data.
`...`	Additional arguments but not implemented.

Value

An R `list` is returned.

`lambdas`	Vector of the exact bootstrap L-moments. First element is $\hat{\lambda}_1$, second element is $\hat{\lambda}_2$, and so on. This vector is from equation 1.3 and 2.4 of Hutson and Ernst (2000).
`ratios`	Vector of the exact bootstrap L-moment ratios. Second element is $\hat{\tau}$, third element is $\hat{\tau}_3$ and so on.
`lambdavars`	The exact bootstrap variances of the L-moments from equation 1.4 of Hutson and Ernst (2000) via `crossprod` matrix operations.
`ratiovars`	The exact bootstrap variances of the L-moment ratios with `NA` inserted for $r = 1, 2$ because $r = 1$ is the mean and $r = 2$ for L-CV is unknown to this author.
`varcovar.lambdas`	The variance-covariance matrix of the L-moments from which the diagonal are the values `lambdavars`.
`varcovar.lambdas.and.ratios`	The variance-covariance matrix of the first two L-moments and for the L-moment ratios (if `nmom`$>= 3$) from which select diagonal are the values `ratiovars`.
`bootstrap.orderstatistics`	The exact bootstrap estimate of the order statistics from equation 2.2 of Hutson and Ernst (2000).
`varcovar.orderstatistics`	The variance-covariance matrix of the order statistics from equations 3.1 and 3.2 of Hutson and Ernst (2000). The diagonal of this matrix represents the variances of each order statistic.

`inverse.varcovar.tau23`
: The inversion of the variance-covariance matrix of τ_2 and τ_3 by Cholesky decomposition. This matrix may be used to estimate a joint confidence region of (τ_2, τ_3) based on asymptotic normality of L-moments.

`inverse.varcovar.tau34`
: The inversion of the variance-covariance matrix of τ_3 and τ_4 by Cholesky decomposition. This matrix may be used to estimate a joint confidence region of (τ_3, τ_4) based on asymptotic normality of L-moments; these two L-moment ratios likely represent the most common ratios used in general L-moment ratio diagrams.

`inverse.varcovar.tau46`
: The inversion of the variance-covariance matrix of τ_4 and τ_6 by Cholesky decomposition. This matrix may be used to estimate a joint confidence region of (τ_4, τ_6) based on asymptotic normality of L-moments; these two L-moment ratios represent those ratios used in L-moment ratio diagrams of symmetrical distributions.

`source`
: An attribute identifying the computational source of the results: "lmoms.bootbarvar".

Note

This function internally defines several functions that provide a direct nomenclature connection to Hutson and Ernst (2000). Interested users are invited to adapt these functions as they might see fit. A reminder is made to sort the data vector as needed; the vector is only sorted once within the `lmoms.bootbarvar` function.

The $100(1-\alpha)$ percent confidence region of the vector $\boldsymbol{\eta} = (\tau_3, \tau_4)$ (for example) based on the sample L-skew and L-kurtosis of the vector $\hat{\boldsymbol{\eta}} = (\hat{\tau}_3, \hat{\tau}_4)$ is expressed as

$$(\boldsymbol{\eta} - \hat{\boldsymbol{\eta}})' \hat{\boldsymbol{P}}^{-1}_{(3,4)} (\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}) \le \chi^2_{2,\alpha}$$

where $\hat{\boldsymbol{P}}_{(3,4)}$ is the variance-covariance matrix of these L-moment ratios subselected from the resulting matrix titled `varcovar.lambdas.and.ratios` but extracted and inverted in the resulting matrix titled `inverse.varcovar.tau34`, which is $\hat{\boldsymbol{P}}^{-1}_{(3,4)}$. The value $\chi^2_{2,\alpha}$ is the upper quantile of the Chi-squared distribution. The inequality represents a standard equal probable ellipse from a Bivariate Normal distribution.

Author(s)

W.H. Asquith

References

Hutson, A.D., and Ernst, M.D., 2000, The exact bootstrap mean and variance of an L-estimator: Journal Royal Statistical Society B, v. 62, part 1, pp. 89–94.

Wang, D., and Hutson, A.D., 2013, Joint confidence region estimation of L-moments with an extension to right censored data: Journal of Applied Statistics, v. 40, no. 2, pp. 368–379.

See Also

lmoms

Examples

```
## Not run:
   para <- vec2par(c(0,1), type="gum") # Parameters of Gumbel
   n <- 10; nmom <- 6; nsim <- 2000
   # X <- rlmomco(n, para) # This is commented out because
   # the sample below is from the Gumbel distribution as in para.
   # However, the seed for the random number generator was not recorded.
   X <- c( -1.4572506, -0.7864515, -0.5226538,  0.1756959,  0.2424514,
            0.5302202,  0.5741403,  0.7708819,  1.9804254,  2.1535666)
   EXACT.BOOTLMR <- lmoms.bootbarvar(X, nmom=nmom)
   LA <- EXACT.BOOTLMR$lambdavars
   LB <- LC <- rep(NA, length(LA))
   set.seed(n)
   for(i in 1:length(LB)) {
     LB[i] <- var(replicate(nsim,
                  lmoms(sample(X, n, replace=TRUE), nmom=nmom)$lambdas[i]))
   }
   set.seed(n)
   for(i in 1:length(LC)) {
     LC[i] <- var(replicate(nsim,
                  lmoms(rlmomco(n, para), nmom=nmom)$lambdas[i]))
   }
   print(LA) # The exact bootstrap variances of the L-moments.
   print(LB) # Bootstrap variances of the L-moments by actual resampling.
   print(LC) # Simulation of the variances from the parent distribution.

   # The variances for this example are as follows:
   #> print(LA)
   #[1] 0.115295563 0.018541395 0.007922893 0.010726508 0.016459913 0.029079202
   #> print(LB)
   #[1] 0.117719198 0.018945827 0.007414461 0.010218291 0.016290100 0.028338396
   #> print(LC)
   #[1] 0.17348653 0.04113861 0.02156847 0.01443939 0.01723750 0.02512031
   # The variances, when using simulation of parent distribution,
   # appear to be generally larger than those based only on resampling
   # of the available sample of only 10 values.

   # Interested users may inspect the exact bootstrap estimates of the
   # order statistics and the variance-covariance matrix.
   # print(EXACT.BOOTLMR$bootstrap.orderstatistics)
   # print(EXACT.BOOTLMR$varcovar.orderstatistics)

   # The output for these two print functions is not shown, but what follows
   # are the numerical confirmations from A.D. Hutson (personnal commun., 2012)
   # using his personnal algorithms (outside of R).
   # Date: Jul 2012, From: ahutson, To: Asquith
   # expected values the same
   # -1.174615143125091, -0.7537760316881618, -0.3595651823632459,
```

```
   # -0.028951905838698,  0.2360931764028858,  0.4614289985084462,
   #  0.713957210869635,  1.0724040932920058,  1.5368435379648948,
   #  1.957207045977329
   # and the first two values on the first row of the matrix are
   # 0.1755400544274771,  0.1306634198810892

## End(Not run)
## Not run:
# Wang and Hutson (2013): Attempt to reproduce first entry of
# row 9 (n=35) in Table 1 of the reference, which is 0.878.
Xsq  <- qchisq(1-0.05, 2); n <- 35; nmom <- 4; nsim <- 1000
para <- vec2par(c(0,1), type="gum") # Parameters of Gumbel
eta  <- as.vector(lmorph(par2lmom(para))$ratios[3:4])
h <- 0
for(i in 1:nsim) {
   X <- rlmomco(n,para); message(i)
   EB <- lmoms.bootbarvar(X, nmom=nmom, verbose=FALSE)
   lmr    <- lmoms(X); etahat <- as.vector(lmr$ratios[c(3,4)])
   Pinv   <- EB$inverse.varcovar.tau34
   deta   <- (eta - etahat)
   LHS <- t(deta)
   if(LHS > Xsq) { # Comparison to Chi-squared distribution
      h <- h + 1 # increment because outside ellipse
      message("Outside: ",i, " ", h, " ", round(h/i, digits=3))
   }
}
message("Empirical Coverage Probability with Alpha=0.05 is ",
        round(1 - h/nsim, digits=3), " and count is", h)
# I have run this loop and recorded an h=123 for the above settings. I compute a
# coverage probability of 0.877, which agrees with Wang and Hutson (2013) within 0.001.
# Hence "very down the line" computations of lmoms.bootbarvar appear to be verified.

## End(Not run)
```

lmoms.cov	*Distribution-Free Variance-Covariance Structure of Sample L-moments*

Description

Compute the distribution-free, variance-covariance matrix ($\widehat{\mathrm{var}}(\lambda)$) of the sample L-moments ($\hat{\lambda}_r$) or alternatively the sample probability-weighted moments ($\hat{\beta}_k$, Elamir and Seheult, 2004, sec. 5). The $\widehat{\mathrm{var}}(\lambda)$ is defined by the matrix product

$$\widehat{\mathrm{var}}(\lambda) = \mathbf{C}\,\hat{\mathbf{\Theta}}\,\mathbf{C}^{\mathrm{T}},$$

where the $r \times r$ matrix $\mathbf{C}$ for number of moments r represents the coefficients of the linear combinations converting β_k to λ_r and the rth row in the matrix is defined as

$$\mathbf{C}[r,]_{k=0:(r-1)} = (-1)^{(r-1-k)}\binom{r-1}{k}\binom{r-1+k}{k},$$

where the row is padded from the right with zeros for $k < r$ to form the required lower triangular structure. Elamir and Seheult (2004) list the $\mathbf{C}$ matrix for $r = 4$.

Letting the *falling factorial* be defined (matching Elamir and Seheult's nomenclature) as

$$a^{(b)} = \Gamma(b+1)\binom{a}{b},$$

and letting an entry in the $\hat{\mathbf{\Theta}}$ matrix denoted as $\hat{\theta}_{kl}$ be defined as

$$\hat{\theta}_{kl} = \hat{\beta}_k\hat{\beta}_l - \frac{A}{n^{(k+l+2)}},$$

where $\hat{\beta}_k$ are again the sample probability-weighted moments and are computed by pwm, and finally A is defined as

$$A = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left[(i-1)^{(k)}(j-k-2)^{(l)} + (i-1)^{(l)}(i-l-2)^{(k)}\right]X_{i:n}X_{j:n},$$

where $X_{i:n}$ are the sample order statistics for a sample of size n.

Incidentally, the matrix $\hat{\mathbf{\Theta}}$ is the variance-covariance structure ($\widehat{\mathrm{var}}$) of the $\hat{\beta}$, thus $\widehat{\mathrm{var}}(\beta) = \hat{\mathbf{\Theta}}$, which can be returned by a logical function argument (`as.pwm=TRUE`) instead of $\widehat{\mathrm{var}}(\lambda)$. The last example in **Examples** provides a demonstration.

Usage

```
lmoms.cov(x, nmom=5, as.pwm=FALSE, showC=FALSE,
             se=c("NA", "lamse", "lmrse", "pwmse"), ...)
```

Arguments

`x`	A vector of data values.
`nmom`	The number of moments to compute. Default is 5.
`as.pwm`	A logical controlling whether the distribution-free, variance-covariance of sample probability-weighted moments ($\hat{\mathbf{\Theta}}$) is returned instead.
`showC`	A logical controlling whether the matrix $\mathbf{C}$ is printed during function operation, and this matrix is not returned as a presumed safety feature.
`se`	Compute standard errors (SE) for the respective moments. The default of `"NA"` retains the return of either $\widehat{\mathrm{var}}(\beta)$ or $\widehat{\mathrm{var}}(\lambda)$ depending on setting of `as.pwm`. The `"lamse"` returns the square root of the diagonal of $\widehat{\mathrm{var}}(\lambda)$, and notationally these are λ_r^{SE}. Similarly, `"pwmse"` returns the square root of the diagonal of $\widehat{\mathrm{var}}(\beta)$ by internally setting `as.pwm` to `TRUE`, and notationally these are β_{r-1}^{SE}. (Remember that $\beta_0 \equiv \lambda_1$—the indexing of the former starts at 0 and at the later at 1). The `"lmrse"` returns the square root of the first two terms of the $\widehat{\mathrm{var}}(\lambda)$ diagonal ($\lambda_{1,2}^{SE}$) but computes SE for the L-moment ratios (τ_r^{SE}) for $r \geq 3$ using

the Taylor-series-based approximation (see **Note**) shown by Elamir and Seheult (2004, p. 348). (Remember that L-moment ratios are $\tau_r = \lambda_r/\lambda_2$ for $r \ge 3$ and that $\tau_2 = \lambda_2/\lambda_1$ [coefficient of L-variation].)

... Other arguments to pass should they be needed (none were at first implementation).

Value

An R `matrix` is returned. In small samples and substantially sized r, one or more $\hat{\theta}_{kl}$ will be `NaN` starting from the lower right corner of the matrix. The function does not test for this nor reduce the number of moments declared in `nmom` itself. To reiterate, the square roots along the $\widehat{\mathrm{var}}(\lambda)$ diagonal are SE for the respective L-moments.

Note

Function `lmoms.cov` was developed as a double check on the evidently separately developed $r \le 4$ (`nmom`) implementations of $\widehat{\mathrm{var}}(\lambda)$ in packages **Lmoments** and **nsRFA**. Also the internal structure closely matches the symbolic mathematics by Elamir and Seheult (2004), but this practice comes at the expense of more than an order of magnitude slower execution times than say either of the functions `Lmomcov()` (package **Lmoments**) or `varLmoments()` (package **nsRFA**). For a high speed and recommended implementation, please use the **Lmoments** package by Karvanen (2016)—Karvanen extended this implementation to larger r for the **lmomco** package.

For `se="lmrse"`, the Taylor-series-based approximation is suggested by Elamir and Seheult (2004, p. 348) to estimate the variance of an L-moment ratio (τ_r for $r \ge 3$) is based on structure of the variance of the ratio of two uniform variables in which the numerator is the rth L-moment and the denominator is λ_2:

$$\mathrm{var}(\tau_r) \cong \left[\frac{\mathrm{var}(\lambda_r)}{\mathrm{E}(\lambda_r)^2} + \frac{\mathrm{var}(\lambda_2)}{\mathrm{E}(\lambda_2)^2} - \frac{2\mathrm{cov}(\lambda_r, \lambda_2)}{\mathrm{E}(\lambda_r)\mathrm{E}(\lambda_2)}\right]\left[\frac{\mathrm{E}(\lambda_r)}{\mathrm{E}(\lambda_2)}\right]^2,$$

where $\mathrm{var}(\cdots)$ are the along the diagonal of $\widehat{\mathrm{var}}(\lambda)$ and $\mathrm{cov}(\cdots)$ are the off-diagonal covariances. The expectations $\mathrm{E}(\cdots)$ are replaced with the sample estimates. Only for `se="lmrse"` the SE of the coefficient of L-variation (τ_2^{SE}) is computed but retained as an attribute (`attr()` function) of the returned vector and not housed within the vector—the λ_2^{SE} continues to be held in the 2nd position of the returned vector.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2004, Exact variance structure of sample L-moments: Journal of Statistical Planning and Inference, v. 124, pp. 337–359.

Karvanen, Juha, 2016, Lmoments—L-moments and quantile mixtures: R package version 1.2-3, accessed February 22, 2016 at https://cran.r-project.org/web/packages/Lmoments/index.html

See Also

`lmoms`, `pwm`

Examples

```
## Not run:
nsim <- 1000; n <- 10 # Let us compute variance of lambda_3
VL3sample <- mean(replicate(nsim, { zz <- lmoms.cov(rexp(n),nmom=3); zz[3,3] }))
falling.factorial <- function(a, b) gamma(b+1)*choose(a,b)
VL3exact  <- ((4*n^2 - 3*n - 2)/30)/falling.factorial (10, 3) # Exact variance is from
print(c(VL3sample, VL3exact)) # Elamir and Seheult (2004, table 1, line 8)
#[1] 0.01755058 0.01703704  # the values obviously are consistent
## End(Not run)
## Not run:
# Data considered by Elamir and Seheult (2004, p. 348)
library(MASS); data(michelson); Light <- michelson$Speed
lmoms(Light, nmom=4)$lambdas # 852.4, 44.3, 0.83, 6.5 # matches those authors
lmoms.cov(Light) # [1, ] ==> 62.4267, 0.7116, 2.5912, -3.9847 # again matches
# The authors report standard error of L-kurtosis as 0.03695, which matches
lmoms.cov(Light, se="lmrse")[4] # 0.03695004
## End(Not run)
## Not run:
D <- rnorm(100) # Check results of Lmoments package.
lmoms.cov(D, rmax=5)[,5]
#        lam1         lam2         lam3         lam4         lam5
#3.662721e-04 3.118812e-05 5.769509e-05 6.574662e-05 1.603578e-04
Lmoments::Lmomcov(D, rmax=5)[,5]
#          L1           L2           L3           L4           L5
#3.662721e-04 3.118812e-05 5.769509e-05 6.574662e-05 1.603578e-04
## End(Not run)
```

---

lmomsla *Trimmed L-moments of the Slash Distribution*

Description

This function estimates the trimmed L-moments of the Slash distribution given the parameters (ξ and α) from parsla. The relation between the TL-moments (trim=1) and the parameters have been numerically determined and are $\lambda_1^{(1)} = \xi$, $\lambda_2^{(1)} = 0.93686275\alpha$, $\tau_3^{(1)} = 0$, $\tau_4^{(1)} = 0.30420472$, $\tau_5^{(1)} = 0$, and $\tau_6^{(1)} = 0.18900723$. These TL-moments (trim=1) are symmetrical for the first L-moments defined because $\mathrm{E}[X_{1:n}]$ and $\mathrm{E}[X_{n:n}]$ are undefined expectations for the Slash.

Usage

```
lmomsla(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the trimmed L-moments. First element is $\lambda_1^{(1)}$, second element is $\lambda_2^{(1)}$, and so on.
ratios	Vector of the L-moment ratios. Second element is $\tau^{(1)}$, third element is $\tau_3^{(1)}$ and so on.
trim	Level of symmetrical trimming used in the computation, which is 1.
leftrim	Level of left-tail trimming used in the computation, which is 1.
rightrim	Level of right-tail trimming used in the computation, which is 1.
source	An attribute identifying the computational source of the L-moments: "lmomsla"
trim	Level of symmetrical trimming used.

Author(s)

W.H. Asquith

References

Rogers, W.H., and Tukey, J.W., 1972, Understanding some long-tailed symmetrical distributions: Statistica Neerlandica, v. 26, no. 3, pp. 211–226.

See Also

parsla, cdfsla, pdfsla, quasla

Examples

```
## Not run:
# This example was used to numerically back into the TL-moments and the
# relation between \alpha and \lambda_2.
"lmomtrim1" <- function(para) {
    bigF <- 0.9999
    minX <- para$para[1] - para$para[2]*qnorm(1 - bigF) / qunif(1 - bigF)
    maxX <- para$para[1] + para$para[2]*qnorm(    bigF) / qunif(1 - bigF)
    minF <- cdfsla(minX, para); maxF <- cdfsla(maxX, para)
    lmr <- theoTLmoms(para, nmom = 6, leftrim = 1, rightrim = 1)
}

U <- -10; i <- 0
As <- seq(.1,abs(10),by=.2)
L1s <- L2s <- T3s <- T4s <- T5s <- T6s <- vector(mode="numeric", length=length(As))
for(A in As) {
   i <- i + 1
   lmr <- lmomtrim1(vec2par(c(U, A), type="sla"))
   L1s[i] <- lmr$lambdas[1]; L2s[i] <- lmr$lambdas[2]
   T3s[i] <- lmr$ratios[3];  T4s[i] <- lmr$ratios[4]
   T5s[i] <- lmr$ratios[5];  T6s[i] <- lmr$ratios[6]
}
```

```
print(summary(lm(L2s~As-1))$coe)
print(mean(T4s))
print(mean(T6s)) #
## End(Not run)

## Not run:
  alpha <- 30
  tlmr <- theoTLmoms(vec2par(c(100, alpha), type="cau"), nmom=6, trim=1)
  print( c(tlmr$lambdas[2] / alpha, tlmr$ratios[c(4,6)]), 8 ) #
## End(Not run)
```

---

lmomsmd *L-moments of the Singh–Maddala Distribution*

Description

This function computes the L-moments of the Singh–Maddala (Burr Type XII) distribution given the parameters (ξ, a, b, and q) from parsmd. The first L-moment (λ_1) for $b' = 1/b$ and $R = a\Gamma(1+b')$ is

$$\lambda_1 = R \times \left[\frac{a\Gamma(1q - b')}{\Gamma(1q)}\right] + \xi.$$

The second L-moment (λ_2) is

$$\lambda_2 = R \times \left[\frac{1\Gamma(1q - b')}{\Gamma(1q)} - \frac{1\Gamma(2q - b')}{\Gamma(2q)}\right].$$

The third L-moment (λ_3) is

$$\lambda_3 = R \times \left[\frac{1\Gamma(1q - b')}{\Gamma(1q)} - \frac{3\Gamma(2q - b')}{\Gamma(2q)} + \frac{2\Gamma(3q - b')}{\Gamma(3q)}\right].$$

The fourth L-moment (λ_4) is

$$\lambda_4 = R \times \left[\frac{1\Gamma(1q - b')}{\Gamma(1q)} - \frac{6\Gamma(2q - b')}{\Gamma(2q)} + \frac{10\Gamma(3q - b')}{\Gamma(3q)} - \frac{5\Gamma(4q - b')}{\Gamma(4q)}\right].$$

The fifth L-moment (λ_5) (unique to **lmomco** development) is

$$\lambda_5 = R \times \left[\frac{1\Gamma(1q - b')}{\Gamma(1q)} - \frac{10\Gamma(2q - b')}{\Gamma(2q)} + \frac{30\Gamma(3q - b')}{\Gamma(3q)} - \frac{35\Gamma(4q - b')}{\Gamma(4q)} + \frac{14\Gamma(5q - b')}{\Gamma(5q)}\right].$$

The sixth L-moment (λ_6) (unique to **lmomco** development) is

$$\lambda_6 = R \times \left[\frac{1\Gamma(1q - b')}{\Gamma(1q)} - \frac{15\Gamma(2q - b')}{\Gamma(2q)} + \frac{70\Gamma(3q - b')}{\Gamma(3q)} - \frac{140\Gamma(4q - b')}{\Gamma(4q)} + \right.$$

$$\left. \frac{126\Gamma(5q - b')}{\Gamma(5q)} - \frac{42\Gamma(6q - b')}{\Gamma(6q)}\right].$$

Usage

```
lmomsmd(para)
```

Arguments

para	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: "lmomsmd".

Author(s)

W.H. Asquith

References

Bhatti, F.A., Hamedani, G.G., Korkmaz, M.C., and Munir Ahmad, M., 2019, New modified Singh–Maddala distribution—Development, properties, characterizations, and applications: Journal of Data Science, v. 17, no. 3, pp. 551–574, doi:10.6339/JDS.201907_17(3).0006.

Shahzad, M.N., and Zahid, A., 2013, Parameter estimation of Singh Maddala distribution by moments: International Journal of Advanced Statistics and Probability, v. 1, no. 3, pp. 121–131, doi:10.14419/ijasp.v1i3.1206.

See Also

parsmd, cdfsmd, pdfsmd, quasmd

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78), nmom=6)
lmr$source <- lmr$trim <- lmr$leftrim <- lmr$rightrim <-NULL
# The parsmd() reports Tau4 is too big and snaps it to an empirical boundary.
# "Tau4(~Tau3) snapped to upper limit, Tau4=0.65483 for Tau3=0.75126"
bmr <- lmomsmd(parsmd(lmr, snap.tau4=TRUE))
dmr <- data.frame(bmr$lambdas, bmr$ratios)
cbind(as.data.frame(lmr), dmr) # See in table that row 4 has different Tau4s
#  lambdas    ratios bmr.lambdas bmr.ratios
# 1   155.0        NA   155.00000         NA
# 2   118.6 0.7651613   118.60000  0.7651613
```

```
# 3    89.1 0.7512648    89.18739  0.7520016
# 4    82.1 0.6922428    77.59904  0.6542921 # see different Tau4s (snapping)
# 5    69.5 0.5860034    68.40150  0.5767411 # We are not fitting to these
# 6   102.5 0.8642496    62.58792  0.5277228 # higher L-moments.

# T3 and T4 of the Gumbel distribution, which is inside the SMD domain.
gumt3t4 <- c(log(9/8)/log(2), (16 * log(2) - 10 * log(3))/log(2))
lmr <- theoLmoms(pargum(vec2lmom(c(155, 118.6, gumt3t4))), nmom=6)
lmr$source <- lmr$trim <- lmr$leftrim <- lmr$rightrim <-NULL
bmr <- lmomsmd(parsmd(lmr, snap.tau4=TRUE))
dmr <- data.frame(bmr$lambdas, bmr$ratios)
cbind(as.data.frame(lmr), dmr)
#      lambdas     ratios bmr.lambdas bmr.ratios
# 1 155.000000         NA  155.000000         NA
# 2 118.600005 0.76516132  118.600005  0.7651613
# 3  20.153103 0.16992498   20.153104  0.1699250
# 4  17.834464 0.15037490   17.834464  0.1503749 # see same Tau4s (no snapping)
# 5   6.625972 0.05586823    7.688957  0.0648310 # We are not fitting to these
# 6   6.891842 0.05810997    7.213039  0.0608182 # higher L-moments.

## Not run:
  # T3 and T4 of the Gumbel distribution, which is inside the SMD domain.
  gumt3t4 <- c(log(9/8)/log(2), (16 * log(2) - 10 * log(3))/log(2))
  FF <- nonexceeds(); qFF <- qnorm(FF)
  gumx <- qlmomco(FF, pargum(vec2lmom(c(155, 118.6, gumt3t4))))
  smdx <- qlmomco(FF, parsmd(lmr, snap.tau4=TRUE))
  plot( qFF, gumx, col="blue", type="l",
       xlab="Standard normal variate", ylab="Quantile")
  lines(qFF, smdx, col="red") #
## End(Not run)
```

`lmomsRCmark` *Sample L-moments Moments for Right-Tail Censoring by a Marking Variable*

Description

Compute the sample L-moments for right-tail censored data set in which censored data values are identified by a marking variable. Extension of left-tail censoring can be made using `fliplmoms` and the example therein.

Usage

```
lmomsRCmark(x, rcmark=NULL, nmom=5, flip=NA, flipfactor=1.1)
```

Arguments

`x`	A vector of data values.
`rcmark`	The right-tail censoring (upper) marking variable for unknown threshold: 0 is uncensored, 1 is censored.

nmom — Number of L-moments to return.

flip — Do the data require flipping so that left-censored data can be processed as such. If the flip is a logical and TRUE, then flipfactor $\times \max(x)$ (the maximum of x) is used. If the flip is a numeric, then it is used as the flip.

flipfactor — The value that is greater than 1, which is multiplied on the maximum of x to determine the flip, if the flip is not otherwise provided.

Value

An R list is returned.

lambdas — Vector of the L-moments. First element is $\hat{\lambda}_1^{(0,0)}$, second element is $\hat{\lambda}_2^{(0,0)}$, and so on. *The returned mean is NOT unflipped.*

ratios — Vector of the L-moment ratios. Second element is $\hat{\tau}^{(0,0)}$, third element is $\hat{\tau}_3^{(0,0)}$ and so on.

trim — Level of symmetrical trimming used in the computation, which will equal NULL if asymmetrical trimming was used. This is not currently implemented as no one has done the derivations.

leftrim — Level of left-tail trimming used in the computation. This is not currently implemented as no one has done the derivations.

rightrim — Level of right-tail trimming used in the computation. This is not currently implemented as no one has done the derivations.

n — The complete sample size.

n.cen — The number of right-censored data values.

flip — The flip used in the computations for support of left-tail censoring.

source — An attribute identifying the computational source of the L-moments: "lmomsRCmark".

Author(s)

W.H. Asquith

References

Wang, Dongliang, Hutson, A.D., Miecznikowski, J.C., 2010, L-moment estimation for parametric survival models given censored data: Statistical Methodology, v. 7, no. 6, pp. 655–667.

Helsel, D.R., 2005, Nondetects and data analysis—Statistics for censored environmental data: Hoboken, New Jersey, John Wiley, 250 p.

See Also

lmomRCmark, fliplmoms

Examples

```
# Efron, B., 1988, Logistic regression, survival analysis, and the
# Kaplan-Meier curve: Journal of the American Statistical Association,
# v. 83, no. 402, pp.414--425
# Survival time measured in days for 51 patients with a marking
# variable in the "time,mark" ensemble. If marking variable is 1,
# then the time is right-censored by an unknown censoring threshold.
Efron <-
c(7,0,  34,0,  42,0,  63,0,  64,0,  74,1,  83,0,  84,0,  91,0,
108,0,  112,0,  129,0,  133,0,  133,0,  139,0,  140,0,  140,0,
146,0,  149,0,  154,0,  157,0,  160,0,  160,0,  165,0,  173,0,
176,0,  185,1,  218,0,  225,0,  241,0,  248,0,  273,0,  277,0,
279,1,  297,0,  319,1,  405,0,  417,0,  420,0,  440,0,  523,1,
523,0,  583,0,  594,0,  1101,0,  1116,1,  1146,0,  1226,1,
1349,1,  1412,1, 1417,1);

# Break up the ensembles into to vectors
ix <- seq(1,length(Efron),by=2)
T  <- Efron[ix]
Efron.data <- T;
Efron.rcmark <- Efron[(ix+1)]

lmr.RC <- lmomsRCmark(Efron.data, rcmark=Efron.rcmark)
lmr.ub <- lmoms(Efron.data)
lmr.noRC <- lmomsRCmark(Efron.data)
PP <- pp(Efron.data)
plot(PP, Efron.data, col=(Efron.rcmark+1), ylab="DATA")
lines(PP, qlmomco(PP, lmom2par(lmr.noRC, type="kap")), lwd=3, col=8)
lines(PP, qlmomco(PP, lmom2par(lmr.ub, type="kap")))
lines(PP, qlmomco(PP, lmom2par(lmr.RC, type="kap")), lwd=2, col=2)
legend(0,1000,c("uncensored L-moments by indicator (Kappa distribution)",
                "unbiased L-moments (Kappa)",
           "right-censored L-moments by indicator (Kappa distribution)"),
                lwd=c(3,1,2), col=c(8,1,2))

########
ZF <- 5 # discharge of undetection of streamflow
Q <- c(rep(ZF,8), 116, 34, 56, 78, 909, 12, 56, 45, 560, 300, 2500)
Qc <- Q == ZF; Qc <- as.numeric(Qc)
lmr     <- lmoms(Q)
lmr.cen <- lmomsRCmark(Q, rcmark=Qc, flip=TRUE)
flip <- lmr.cen$flip
fit  <- pargev(lmr);                    fit.cen <- pargev(lmr.cen)
F <- seq(0.001, 0.999, by=0.001)
Qfit     <-        qlmomco(    F, fit    )
Qfit.cen <- flip - qlmomco(1 - F, fit.cen) # remember to reverse qdf
plot(pp(Q),sort(Q), log="y", xlab="NONEXCEED PROB.", ylab="QUANTILE")
lines(F, Qfit);   lines(F, Qfit.cen,col=2)
```

`lmomst3` *L-moments of the 3-Parameter Student t Distribution*

Description

This function estimates the first six L-moments of the 3-parameter Student t distribution given the parameters (ξ, α, ν) from parst3. The L-moments in terms of the parameters are

$$\lambda_1 = \xi,$$

$$\lambda_2 = 2^{6-4\nu}\pi\alpha\nu^{1/2}\,\Gamma(2\nu-2)/[\Gamma(\frac{1}{2}\nu)]^4 \text{ and}$$

$$\tau_4 = \frac{15}{2}\frac{\Gamma(\nu)}{\Gamma(\frac{1}{2})\Gamma(\nu-\frac{1}{2})}\int_0^1 \frac{(1-x)^{\nu-3/2}[I_x(\frac{1}{2},\frac{1}{2}\nu)]^2}{\sqrt{x}}\,\mathrm{d}x - \frac{3}{2},$$

where $I_x(\frac{1}{2},\frac{1}{2}\nu)$ is the cumulative distribution function of the Beta distribution. The distribution is symmetrical so that $\tau_r = 0$ for odd values of $r : r \geq 3$.

Numerical integration of is made to estimate τ_4. The other two parameters are readily solved for when ν is available. A polynomial approximation is used to estimate the τ_6 as a function of τ_4; the polynomial was based on the theoLmoms estimating τ_4 and τ_6. The τ_6 polynomial has nine coefficients with a maximum absolute residual value of 2.065e-06 for 4,000 degrees of freedom (see `inst/doc/t4t6/studyST3.R`).

Usage

```
lmomst3(para, ...)
```

Arguments

`para`	The parameters of the distribution.
`...`	Additional arguments to pass.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: “lmomst3”.

Author(s)

W.H. Asquith with A.R. Biessen

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

parst3, cdfst3, pdfst3, quast3

Examples

```
lmomst3(vec2par(c(1124, 12.123, 10), type="st3"))
```

---

lmomtexp *L-moments of the Truncated Exponential Distribution*

Description

This function estimates the L-moments of the Truncated Exponential distribution. The parameter ψ is the right truncation of the distribution and α is a scale parameter, letting $\beta = 1/\alpha$ to match nomenclature of Vogel and others (2008), the L-moments in terms of the parameters, letting $\eta = \exp(-\alpha\psi)$, are

$$\lambda_1 = \frac{1}{\beta} - \frac{\psi\eta}{1-\eta},$$

$$\lambda_2 = \frac{1}{1-\eta}\left[\frac{1+\eta}{2\beta} - \frac{\psi\eta}{1-\eta}\right],$$

$$\lambda_3 = \frac{1}{(1-\eta)^2}\left[\frac{1+10\eta+\eta^2}{6\alpha} - \frac{\psi\eta(1+\eta)}{1-\eta}\right], \text{and}$$

$$\lambda_4 = \frac{1}{(1-\eta)^3}\left[\frac{1+29\eta+29\eta^2+\eta^3}{12\alpha} - \frac{\psi\eta(1+3\eta+\eta^2)}{1-\eta}\right].$$

The distribution is restricted to a narrow range of L-CV ($\tau_2 = \lambda_2/\lambda_1$). If $\tau_2 = 1/3$, the process represented is a stationary Poisson for which the probability density function is simply the uniform distribution and $f(x) = 1/\psi$. If $\tau_2 = 1/2$, then the distribution is represented as the usual exponential distribution with a location parameter of zero and a scale parameter $1/\beta$. Both of these limiting conditions are supported.

If the distribution shows to be Uniform ($\tau_2 = 1/3$), then $\lambda_1 = \psi/2$, $\lambda_2 = \psi/6$, $\tau_3 = 0$, and $\tau_4 = 0$. If the distribution shows to be Exponential ($\tau_2 = 1/2$), then $\lambda_1 = \alpha$, $\lambda_2 = \alpha/2$, $\tau_3 = 1/3$ and $\tau_4 = 1/6$.

Usage

```
lmomtexp(para)
```

Arguments

para	The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: "lmomtexp".

Author(s)

W.H. Asquith

References

Vogel, R.M., Hosking, J.R.M., Elphick, C.S., Roberts, D.L., and Reed, J.M., 2008, Goodness of fit of probability distributions for sightings as species approach extinction: Bulletin of Mathematical Biology, DOI 10.1007/s11538-008-9377-3, 19 p.

See Also

`partexp`, `cdftexp`, `pdftexp`, `quatexp`

Examples

```
set.seed(1) # to get a suitable L-CV
X <- rexp(1000, rate=.001) + 100
Y <- X[X <= 2000]
lmr <- lmoms(Y)

print(lmr$lambdas)
print(lmomtexp(partexp(lmr))$lambdas)

print(lmr$ratios)
print(lmomtexp(partexp(lmr))$ratios)
```

lmomTLgld *Trimmed L-moments of the Generalized Lambda Distribution*

Description

This function estimates the symmetrical trimmed L-moments (TL-moments) for $t = 1$ of the Generalized Lambda distribution given the parameters (ξ, α, κ, and h) from parTLgld. The TL-moments in terms of the parameters are complicated; however, there are analytical solutions. There are no simple expressions of the parameters in terms of the L-moments. The first four TL-moments (trim = 1) of the distribution are

$$\lambda^{(1)}_1 = \xi + 6\alpha \left(\frac{1}{(\kappa+3)(\kappa+2)} - \frac{1}{(h+3)(h+2)}\right),$$

$$\lambda^{(1)}_2 = 6\alpha \left(\frac{\kappa}{(\kappa+4)(\kappa+3)(\kappa+2)} + \frac{h}{(h+4)(h+3)(h+2)}\right),$$

$$\lambda^{(1)}_3 = \frac{20\alpha}{3} \left(\frac{\kappa(\kappa-1)}{(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2)} - \frac{h(h-1)}{(h+5)(h+4)(h+3)(h+2)}\right),$$

$$\lambda^{(1)}_4 = \frac{15\alpha}{2} \left(\frac{\kappa(\kappa-2)(\kappa-1)}{(\kappa+6)(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2)} + \frac{h(h-2)(h-1)}{(h+6)(h+5)(h+4)(h+3)(h+2)}\right),$$

$$\lambda^{(1)}_5 = \frac{42\alpha}{5} \left(N1 - N2\right),$$

where

$$N1 = \frac{\kappa(\kappa-3)(\kappa-2)(\kappa-1)}{(\kappa+7)(\kappa+6)(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2)} \text{ and}$$

$$N2 = \frac{h(h-3)(h-2)(h-1)}{(h+7)(h+6)(h+5)(h+4)(h+3)(h+2)}.$$

The TL-moment ($t = 1$) for $\tau^{(1)}_3$ is

$$\tau^{(1)}_3 = \frac{10}{9} \left(\frac{\kappa(\kappa-1)(h+5)(h+4)(h+3)(h+2) - h(h-1)(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2)}{(\kappa+5)(h+5) \times [\kappa(h+4)(h+3)(h+2) + h(\kappa+4)(\kappa+3)(\kappa+2)]}\right).$$

The TL-moment ($t = 1$) for $\tau^{(1)}_4$ is

$$N1 = \kappa(\kappa-2)(\kappa-1)(h+6)(h+5)(h+4)(h+3)(h+2),$$

$$N2 = h(h-2)(h-1)(\kappa+6)(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2),$$

$$D1 = (\kappa+6)(h+6)(\kappa+5)(h+5),$$

$$D2 = [\kappa(h+4)(h+3)(h+2) + h(\kappa+4)(\kappa+3)(\kappa+2)], \text{ and}$$

$$\tau^{(1)}_4 = \frac{5}{4} \left(\frac{N1 + N2}{D1 \times D2}\right).$$

Finally the TL-moment ($t = 1$) for $\tau_5^{(1)}$ is

$$N1 = \kappa(\kappa-3)(\kappa-2)(\kappa-1)(h+7)(h+6)(h+5)(h+4)(h+3)(h+2),$$

$$N2 = h(h-3)(h-2)(h-1)(\kappa+7)(\kappa+6)(\kappa+5)(\kappa+4)(\kappa+3)(\kappa+2),$$

$$D1 = (\kappa+7)(h+7)(\kappa+6)(h+6)(\kappa+5)(h+5),$$

$$D2 = [\kappa(h+4)(h+3)(h+2) + h(\kappa+4)(\kappa+3)(\kappa+2)], \text{ and}$$

$$\tau_5^{(1)} = \frac{7}{5}\left(\frac{N1 - N2}{D1 \times D2}\right).$$

By inspection the τ_r equations are not applicable for negative integer values $k = \{-2, -3, -4, \dots\}$ and $h = \{-2, -3, -4, \dots\}$ as division by zero will result. There are additional, but difficult to formulate, restrictions on the parameters both to define a valid Generalized Lambda distribution as well as valid L-moments. Verification of the parameters is conducted through `are.pargld.valid`, and verification of the L-moment validity is conducted through `are.lmom.valid`.

Usage

```
lmomTLgld(para, nmom=6, trim=1, leftrim=NULL, rightrim=NULL, tau34=FALSE)
```

Arguments

`para`	The parameters of the distribution.
`nmom`	Number of L-moments to compute.
`trim`	Symmetrical trimming level set to unity as the default.
`leftrim`	Left trimming level, t_1.
`rightrim`	Right trimming level, t_2.
`tau34`	A logical controlling the level of L-moments returned by the function. If true, then this function returns only τ_3 and τ_4; this feature might be useful in certain research applications of the Generalized Lambda distribution associated with the multiple solutions possible for the distribution.

Details

The opening comments in the description pertain to single and symmetrical endpoint trimming, which has been extensively considered by Asquith (2007). Deriviations backed by numerical proofing of variable arrangement in March 2011 led the the inclusion of the following generalization of the L-moments and TL-moments of the Generalized Lambda shown in Asquith (2011) that was squeezed in late ahead of the deadlines for that monograph.

$$\lambda_r^{(t_1,t_2)} = \alpha(r^{-1})(r+t_1+t_2)\sum_{j=0}^{r-1}(-1)^r\binom{r-1}{j}\binom{r+t_1+t_2-1}{r+t_1-j-1} \times A,$$

where A is

$$A = \left(\frac{\Gamma(\kappa+r+t_1-j)\Gamma(t_2+j+1)}{\Gamma(\kappa+r+t_1+t_2+1)} - \frac{\Gamma(r+t_1-j)\Gamma(h+t_2+j+1)}{\Gamma(h+r+t_1+t_2+1)}\right),$$

where for the special condition of $r = 1$, the real mean is

$$\mathrm{mean} = \xi + \lambda_1^{(t_1,t_2)},$$

but for $r \geq 2$ the $\lambda^{(t_1,t_2)}$ provides correct values. So care is needed algorithmically also when $\tau_2^{(t1,t2)}$ is computed. Inspection of the $\Gamma(\cdot)$ arguments, which must be > 0, shows that

$$\kappa > -(1 + t_1)$$

and

$$h > -(1 + t_2).$$

Value

An R `list` is returned.

`lambdas`	Vector of the TL-moments. First element is $\lambda_1^{(t_1,t_2)}$, second element is $\lambda_2^{(t_1,t_2)}$, and so on.
`ratios`	Vector of the TL-moment ratios. Second element is $\tau^{(1)}$, third element is $\tau_3^{(1)}$ and so on.
`trim`	Trim level = left or right values if they are equal. The default for this function is `trim = 1` because the `lmomgld` provides for `trim = 0`.
`leftrim`	Left trimming level
`rightrim`	Right trimming level
`source`	An attribute identifying the computational source of the TL-moments: “lmomTL-gld”.

Author(s)

W.H. Asquith

Source

Derivations conducted by W.H. Asquith on February 18 and 19, 2006 and others in early March 2011.

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational statistics and data analysis, v. 43, pp. 299–314.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

lmomgld, parTLgld, pargld, cdfgld, quagld

Examples

```
## Not run:
lmomgld(vec2par(c(10,10,0.4,1.3), type='gld'))

PARgld <- vec2par(c(15,12,1,.5), type="gld")
theoTLmoms(PARgld, leftrim=0, rightrim=0, nmom=6)
lmomTLgld(PARgld, leftrim=0, rightrim=0)

theoTLmoms(PARgld, trim=2, nmom=6)
lmomTLgld(PARgld, trim=2)

theoTLmoms(PARgld, trim=3, nmom=6)
lmomTLgld(PARgld, leftrim=3, rightrim=3)

theoTLmoms(PARgld, leftrim=10, rightrim=2, nmom=6)
lmomTLgld(PARgld, leftrim=10, rightrim=2)

## End(Not run)
```

---

lmomTLgpa *Trimmed L-moments of the Generalized Pareto Distribution*

Description

This function estimates the symmetrical trimmed L-moments (TL-moments) for $t = 1$ of the Generalized Pareto distribution given the parameters (ξ, α, and κ) from parTLgpa. The TL-moments in terms of the parameters are

$$\lambda^{(1)}_1 = \xi + \frac{\alpha(\kappa + 5)}{(\kappa + 3)(\kappa + 2)},$$

$$\lambda^{(1)}_2 = \frac{6\alpha}{(\kappa + 4)(\kappa + 3)(\kappa + 2)},$$

$$\tau^{(1)}_3 = \frac{10(1 - \kappa)}{9(\kappa + 5)}, \text{ and}$$

$$\tau^{(1)}_4 = \frac{5(\kappa - 1)(\kappa - 2)}{4(\kappa + 6)(\kappa + 5)}.$$

Usage

```
lmomTLgpa(para)
```

Arguments

para	The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the trimmed L-moments. First element is $\lambda_1^{(1)}$, second element is $\lambda_2^{(1)}$, and so on.
`ratios`	Vector of the L-moment ratios. Second element is $\tau^{(1)}$, third element is $\tau_3^{(1)}$ and so on.
`trim`	Level of symmetrical trimming used in the computation, which is unity.
`leftrim`	Level of left-tail trimming used in the computation, which is unity.
`rightrim`	Level of right-tail trimming used in the computation, which is unity.
`source`	An attribute identifying the computational source of the TL-moments: "lmomTL-gpa".

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

`lmomgpa`, `parTLgpa`, `cdfgpa`, `pdfgpa`, `quagpa`

Examples

```
TL <- TLmoms(c(123,34,4,654,37,78,21,3400),trim=1)
TL
lmomTLgpa(parTLgpa(TL))
```

`lmomtri` *L-moments of the Asymmetric Triangular Distribution*

Description

This function estimates the L-moments of the Asymmetric Triangular distribution given the parameters (ν, ω, and ψ) from `partri`. The first three L-moments in terms of the parameters are

$$\lambda_1 = \frac{(\nu + \omega + \psi)}{3},$$

$$\lambda_2 = \frac{1}{15}\left[\frac{(\nu - \omega)^2}{(\psi - \nu)} - (\nu + \omega) + 2\psi\right], \text{ and}$$

$$\lambda_3 = G + H_1 + H_2 + J,$$

where G is dependent on the integral defining the L-moments in terms of the quantile function (Asquith, 2011, p. 92) with limits of integration of $[0, P]$, H_1 and H_2 are dependent on the integral defining the L-moment in terms of the quantile function with limits of integration of $[P, 1]$, and J is dependent on the λ_2 and λ_1. Finally, the variables G, H_1, H_2, and J are

$$G = \frac{2}{7}\frac{(\nu + 6\omega)(\omega - \nu)^3}{(\psi - \nu)^3},$$

$$H_1 = \frac{12}{7}\frac{(\omega - \psi)^4}{(\nu - \psi)^3} - 2\psi\frac{(\nu - \omega)^3}{(\nu - \psi)^3} + 2\psi,$$

$$H_2 = \frac{4}{5}\frac{(5\nu - 6\omega + \psi)(\omega - \psi)^2}{(\nu - \psi)^2}, \text{ and}$$

$$J = -\frac{1}{15}\left[\frac{3(\nu - \omega)^2}{(\psi - \nu)} + 7(\nu + \omega) + 16\psi\right].$$

The higher L-moments are even more ponderous and simpler expressions for the L-moment ratios appear elusive. Bounds for τ_3 and τ_4 are $|\tau_3| \leq 0.14285710$ and $0.04757138 < \tau_4 < 0.09013605$. An approximation for τ_4 is

$$\tau_4 = 0.09012180 - 1.777361\tau_3^2 - 17.89864\tau_3^4 + 920.4924\tau_3^6 - 37793.50\tau_3^8,$$

where the residual standard error is $<1.750 \times 10^{-5}$ and the absolute value of the maximum residual is $< 9.338 \times 10^{-5}$. The L-moments of the Symmetrical Triangular distribution for $\tau_3 = 0$ are considered by Nagaraja (2013) and therein for a symmetric triangular distribution having $\lambda_1 = 0.5$ then $\lambda_4 = 0.0105$ and $\tau_4 = 0.09$. These L-kurtosis values agree with results of this function that are based on the `theoLmoms.max.ostat` function. The 4th and 5th L-moments λ_4 and λ_5, respectively, are computed using expectations of order statistic maxima (`expect.max.ostat`) and are defined (Asquith, 2011, p. 95) as

$$\lambda_4 = 5\mathrm{E}[X_{4:4}] - 10\mathrm{E}[X_{3:3}] + 6\mathrm{E}[X_{2:2}] - \mathrm{E}[X_{1:1}]$$

and

$$\lambda_5 = 14\mathrm{E}[X_{5:5}] - 35\mathrm{E}[X_{4:4}] + 30\mathrm{E}[X_{3:3}] - 10\mathrm{E}[X_{2:2}] + \mathrm{E}[X_{1:1}].$$

These expressions are solved using the `expect.max.ostat` function to compute the $\mathrm{E}[X_{r:r}]$.

For the symmetrical case of $\omega = (\psi + \nu)/2$, then

$$\lambda_1 = \frac{(\nu + \psi)}{2} \text{ and}$$

$$\lambda_2 = \frac{7}{60}\left[\psi - \nu\right],$$

which might be useful for initial parameter estimation through

$$\psi = \lambda_1 + \frac{30}{7}\lambda_2 \text{ and}$$

$$\nu = \lambda_1 - \frac{30}{7}\lambda_2.$$

Usage

```
lmomtri(para, paracheck=TRUE, nmom=c("3", "5"))
```

Arguments

`para`	The parameters of the distribution.
`paracheck`	A logical controlling whether the parameters and checked for validity. Overriding of this check might help in numerical optimization of parameters for modes near either the minimum or maximum. The argument here makes code base within `partri` a little shorter.
`nmom`	The L-moments greater the $r > 3$ require numerical integration using the expectations of the maxima order statistics of the fitted distribution. If this argument is set to `"3"` then executation of `lmomtri` is stopped at $r = 3$ and the first three L-moments returned, otherwise the 4th and 5th L-moments are computed.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`E33err`	A percent error between the expectation of the $X_{3:3}$ order statistic by analytical expression versus a theoretical by numerical integration using the `expect.max.ostat` function. This will be `NA` if `nmom == "3"`.
`source`	An attribute identifying the computational source of the L-moments: "lmomtri".

Note

The expression for τ_4 in terms of τ_3 is

```
"tau4tri" <- function(t3) {
   t3[t3 < -0.14285710 | t3 >  0.14285710] <- NA
   b <- 0.09012180
   a <- c(0, -1.777361, 0, -17.89864, 0,  920.4924, 0, -37793.50)
   t4 <- b + a[2]*t3^2 + a[4]*t3^4 + a[6]*t3^6 + a[8]*t3^8
   return(t4)
}
```

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Nagaraja, H.N., 2013, Moments of order statistics and L-moments for the symmetric triangular distribution: Statistics and Probability Letters, v. 83, no. 10, pp. 2357–2363.

See Also

partri, cdftri, pdftri, quatri

Examples

```
lmr <- lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52))
lmr
lmomtri(partri(lmr), nmom="5")

par <- vec2par(c(-405, -390, -102), type="tri")
lmomtri(par, nmom="5")$lambdas
# -299           39.4495050    5.5670228    1.9317914    0.8007511
theoLmoms.max.ostat(para=par, qua=quatri, nmom=5)$lambdas
# -299.0000126   39.4494885    5.5670486    1.9318732    0.8002989
# The -299 is the correct by exact solution as are 39.4495050 and 5.5670228, the 4th and
# 5th L-moments diverge from theoLmoms.max.ostat() because the exact solutions and not
# numerical integration of the quantile function was used for E11, E22, and E33.
# So although E44 and E55 come from expect.max.ostat() within both lmomtri() and
# theoLmoms.max.ostat(), the Lambda4 and Lambda5 are not the same because the E11, E22,
# and E33 values are different.

## Not run:
# At extreme limit of Tau3 for the triangular distribution, L-moment ratio diagram
# shows convergence to the trajectory of the Generalized Pareto distribution.
"tau4tri" <- function(t3) { t3[t3 < -0.14285710 | t3 >  0.14285710] <- NA
   b <- 0.09012180; a <- c(0, -1.777361, 0, -17.89864, 0,  920.4924, 0, -37793.50)
   t4 <- b + a[2]*t3^2 + a[4]*t3^4 + a[6]*t3^6 + a[8]*t3^8; return(t4)
}
F <- seq(0,1, by=0.001)
lmr  <- vec2lmom(c(10,9,0.142857, tau4tri(0.142857)))
parA <- partri(lmr); parB <- pargpa(lmr)
xA <- qlmomco(F,  parA); xB <- qlmomco(F, parB); x <- sort(unique(c(xA,xB)))
plot(x,  pdftri(x,parA), type="l", col=8, lwd=4) # Compare Asym. Tri. to
lines(x, pdfgpa(x,parB),           col=2)        # Gen. Pareto

## End(Not run)
```

---

lmomwak *L-moments of the Wakeby Distribution*

Description

This function estimates the L-moments of the Wakeby distribution given the parameters (ξ, α, β, γ, and δ) from parwak. The L-moments in terms of the parameters are complicated and solved numerically.

Usage

```
lmomwak(wakpara)
```

Arguments

wakpara	The parameters of the distribution.

Value

An R list is returned.

lambdas	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
ratios	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
trim	Level of symmetrical trimming used in the computation, which is 0.
leftrim	Level of left-tail trimming used in the computation, which is NULL.
rightrim	Level of right-tail trimming used in the computation, which is NULL.
source	An attribute identifying the computational source of the L-moments: “lmomwak”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parwak, cdfwak, pdfwak, quawak

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomwak(parwak(lmr))
```

`lmomwei` *L-moments of the Weibull Distribution*

Description

This function estimates the L-moments of the Weibull distribution given the parameters (ζ, β, and δ) from parwei. The Weibull distribution is a reverse Generalized Extreme Value distribution. As result, the Generalized Extreme Value algorithms (lmomgev) are used for computation of the L-moments of the Weibull in this package (see parwei).

Usage

```
lmomwei(para)
```

Arguments

`para`	The parameters of the distribution.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which is `0`.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL`.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL`.
`source`	An attribute identifying the computational source of the L-moments: “lmomwei”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

parwei, cdfwei, pdfwei, quawei

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lmr
lmomwei(parwei(lmr))
```

lmorph *Morph an L-moment Object*

Description

Morph or change one L-moment object type into another. The first L-moment object created for **lmomco** used an R `list` with named L-moment values (`lmom.ub`) such as `L1` or `TAU3`. This object was bounded for L-moment orders less than or equal to five. However, subsequent **lmomco** development in early 2006 that was related to the trimmed L-moments suggested that an alternative L-moment object structure be used that utilized two vectors for the L-moments and the L-moment ratios (`lmorph`). This second object type is not bounded by L-moment order. In turn it became important to seemlessly morph from one object structure to the other and back again. The canonical structure of the first L-moment object type is documented under `lmom.ub`; whereas, the canonical structure for the second L-moment object type is documented under `lmoms` (actually through `TLmoms`). Because the first L-moment object is bounded by five, L-moment order larger than this will be ignored in the morphing process.

Usage

```
lmorph(lmom)
```

Arguments

`lmom`	An L-moment object of type like `lmom.ub` or `lmoms`.

Value

A two different R `lists` (L-moment objects), which are the opposite of the argument type—see the documentation for `lmom.ub` and `lmoms`.

Note

If any of the trimming characteristics of the second type of L-moment object (`trim`, `leftrim`, or `rightrim`) have a greater than zero value, then conversion to the L-moment object with named values will not be performed. A message will be provided that the conversion was not performed. In April 2014, it was decided that all `lmomCCC()` functions, such as `lmomgev` or `lmomnor`, would be standardized to the less limited and easier to maintain vector output style of `lmoms`.

Author(s)

W.H. Asquith

See Also

`lmom.ub`, `lmoms`, `TLmoms`

Examples

```
lmr <- lmom.ub(c(123,34,4,654,37,78))
lmorph(lmr)
lmorph(lmorph(lmr))
```

`lmrdia` *L-moment Ratio Diagram Components*

Description

This function returns a list of the L-skew and L-kurtosis (τ_3 and τ_4, respectively) ordinates for construction of L-moment Ratio (L-moment diagrams) that are useful in selecting a distribution to model the data.

Usage

```
lmrdia()
```

Value

An R `list` is returned.

`limits`	The theoretical limits of τ_3 and τ_4; below τ_4 of the theoretical limits are theoretically not possible.
`aep4`	τ_3 and τ_4 lower limits of the Asymmetric Exponential Power distribution.
`cau`	$\tau_3^{(1)} = 0$ and $\tau_4^{(1)} = 0.34280842$ of the Cauchy distribution (TL-moment [trim=1]) (see **Examples** `lmomcau` for source).
`exp`	τ_3 and τ_4 of the Exponential distribution.
`gev`	τ_3 and τ_4 of the Generalized Extreme Value distribution.
`glo`	τ_3 and τ_4 of the Generalized Logistic distribution.
`gpa`	τ_3 and τ_4 of the Generalized Pareto distribution.
`gum`	τ_3 and τ_4 of the Gumbel distribution.
`gno`	τ_3 and τ_4 of the Generalized Normal distribution.
`gov`	τ_3 and τ_4 of the Govindarajulu distribution.
`ray`	τ_3 and τ_4 of the Rayleigh distribution.
`lognormal`	τ_3 and τ_4 of the Generalized Normal (3-parameter Log-Normal) distribution.
`nor`	τ_3 and τ_4 of the Normal distribution.
`pe3`	τ_3 and τ_4 of the Pearson Type III distribution.
`pdq3`	τ_3 and τ_4 of the Polynomial Density-Quantile3 distribution.
`rgov`	τ_3 and τ_4 of the reversed Govindarajulu.
`rgpa`	τ_3 and τ_4 of the reversed Generalized Pareto.
`sla`	$\tau_3^{(1)} = 0$ and $\tau_4^{(1)} = 0.30420472$ of the Slash distribution (TL-moment [trim=1]) (see **Examples** `lmomsla` for source).
`uniform`	τ_3 and τ_4 of the uniform distribution.
`wei`	τ_3 and τ_4 of the Weibull distribution (reversed Generalized Extreme Value).

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2,870–2,891, doi:10.1016/j.jspi.2006.10.010.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

plotlmrdia

Examples

```
lratios <- lmrdia()
```

---

lmrdia46 — *L-moment Ratio Diagram Components of Tau4 and Tau6*

Description

This function returns a list of the L-kurtosis (τ_4 and sixth L-moment ratio τ_6, respectively) ordinates for construction of L-moment Ratio (L-moment diagrams) that are useful in selecting a distribution to model the data.

Usage

```
lmrdia46()
```

Details

The `lmrdia46` returns a list of the tables for drawing the trajectories of the distributions by its access of `.lmomcohash$t46list` created by the `inst/doc/SysDataBuilder02.R` script for `sysdata.rda` construction used by the **lmomco** package itself. The lookup table references below are pointing to the `inst/doc/t4t6` subdirectory of the package.

A lookup table for the Exponential Power distribution is provided as `PowerExponential.txt` (`.lmomcohash$tau46list$pwrexp`), and this distribution is a special case of the Asymmetric Exponential Power4 (`lmomaep4`) (`.lmomcohash$tau46list$aep4`).

A lookup table for the Symmetric Stable distribution is provided as `StableDistribution.txt` (`.lmomcohash$tau46list$symstable`).

A lookup table for the Student t distribution is provided as `StudentT.txt` (`.lmomcohash$tau46list$st2`), and this distribution is the same as the Student 3t (`lmomst3`) (`.lmomcohash$tau46list$st3`).

A lookup table for the Tukey Lamda distribution is provided as `SymTukeyLambda.txt` (`.lmomcohash$tau46list$tukeylam`), and this distribution is not quite the same as the Generalized Lambda distribution (`lmomgld`) (`.lmomcohash$tau46list$gld`).

The normal distribution plots as a point in a Tau4-Tau6 L-moment ratio diagram as `.lmomcohash$tau46list$nor` for which $\tau_4^{\mathrm{nor}} = 30/\pi \times \mathrm{atan}(\sqrt{2}) - 9 = 0.1226017$ and $\tau_6^{\mathrm{nor}} = 0.04365901$ (numerical integration).

Finally, the Cauchy and Slade distributions are symmetrical and can be plotted as well on Tau4-Tau6 L-moment ratio diagram if we permit their trim=1 TL-moments to be shown instead. These are inserted into the returned list as part of the operation of `lmrdia46()`.

Tukey Lambda Notes—The Tukey Lambda distribution is a simpler formulation than the Generalized Lambda.

$$Q(F) = \frac{1}{\lambda}\left[F^{\lambda} - (1-F)^{\lambda}\right],$$

for nonexceedance probability F and $\lambda \neq 0$ and

$$Q(F) = \log\left(\frac{F}{1-F}\right),$$

for $\lambda = 0$ using the natural logarithm.

Inspection of the distribution formulae inform us that the variation in the distribution, the scaling factor $1/\lambda$ to far left in the first definition, for instance, implies that the L-scale (λ_2) is not constant and varies with λ. The second L-moment of the Tukey Lambda (all odd order L-moments are zero) is

$$\lambda_2 = \frac{2}{\lambda}\left[-\frac{1}{1+\lambda} + \frac{2}{2+\lambda}\right], \text{ and}$$

the fourth and sixth L-moments are

$$\lambda_4 = \frac{2}{\lambda}\left[-\frac{1}{1+\lambda} + \frac{12}{2+\lambda} - \frac{30}{3+\lambda} + \frac{20}{4+\lambda}\right],$$

$$\lambda_6 = \frac{2}{\lambda}\left[-\frac{1}{1+\lambda} + \frac{30}{2+\lambda} - \frac{210}{3+\lambda} + \frac{560}{4+\lambda} - \frac{630}{5+\lambda} + \frac{252}{6+\lambda}\right] \text{ and}$$

$\tau_4 = \lambda_4/\lambda_2$ and $\tau_6 = \lambda_6/\lambda_2$. The Tukey Lambda is not separately implemented in the **lmomco** package. It is provided herein for theoretical completeness, but it is possible to implement the Tukey Lambda by the following example:

```
tukeylam <- .lmomcohash$tau46list$gld_byt6tukeylam
lmr1 <- tukeylam[tukeylam$lambda2 == 1, ] # L-scale equal to one (for instance)
lmr1 <- vec2lmom(c(0, lmr1$lambda2, 0, lmr1$tau4, 0, lmr1$tau6))
tuk1 <- pargld(lmr1, aux="tau6")
print(tuk1$para, 12)
#                 xi                alpha                kappa                  h
#  2.50038766315e-04 -5.82180675380e+03 -1.71745206920e-04 -1.71702273015e-04
lambda <- mean(tuk1$para[3:4]) # remember optimization is used for parameters in
# GLD parlance and so the two shape parameters are not constrained in pargld()
# to be numerically identical. So, here, let us compute a mean of the two and then
# use that as the Lambda in the distribution.
eps <- 1/tuk1$para[2] - lambda
message("EPS should be very close to zero, eps = ", eps, " !!!!!")
tuk2 <- vec2par(c(0, 1/lambda, lambda, lambda), type="gld") # now Tukey Lambda
lmr2 <- lmomgld(tuk2)

"ofunc" <- function(lambda, lambda2=NA) {
  tukeyL2 <- ( 2 / lambda ) * ( -1 / (1+lambda) + 2 / (2+lambda) )
  return(lambda2 - tukeyL2)
}
lam  <- uniroot(ofunc, interval=c(-1, 1), lambda2=1)$root
tuk3 <- vec2par(c(0, 20/lam, lam, lam), type="gld")
lmr3 <- lmomgld(tuk3)

gld5 <- pargld(lmr3, aux="tau5"); gldlmr5 <- theoLmoms(gld5, nmom=6)
gld6 <- pargld(lmr3, aux="tau6"); gldlmr6 <- theoLmoms(gld6, nmom=6)
plotlmrdia46(lmrdia46(), nogld_byt5opt=FALSE)
points(gldlmr5$ratios[4], gldlmr5$ratios[6], pch=16, col="purple")
points(gldlmr6$ratios[4], gldlmr6$ratios[6], pch=21, col="purple", bg="white")
# See how GLD by tau5 optimization, which leaves Tau6 to float plots on the
# "gld_byt5opt" trajectory, but GLD by tau6 optimization, plots on the Tukey
# Lambda line, and gld6$para[2] / (1/gld6$para[3]) is equal to the 20 in the
# parameter setting for tuk3.
```

The finally differences in the L-moments between the two `lmr` objects are all are reasonably close to zero with the recognition that `optim()` has been involved getting us close to the Tukey Lambda that we desire as a GLD with fixed shape parameters and scale factor equal to the inverse of the shape parameter. The demonstration to how to thus acquire a Tukey Lambda from GLD implementation in the **lmomco** package is thus shown.

Value

An R `list` is returned.

`aep4` : τ_4 and τ_6 of the 4-parameter Asymmetric Exponential Power (AEP4) distribution given L-skew set as $\tau_3 = 0$. This becomes then the (Symmetrical) Exponential Power. The complementary entry `pwrexp` are the effectively the same curve for the power exponential distribution based on lookup table archived in the **lmomco** package. The table stems from `inst/doc/SysDataBuilder02.R`. The `aep4` and not `pwrexp` is used in the line drawing by `plotlmrdia46`.

`gld_byt5opt` : τ_4 and τ_6 of the Generalized Lambda (GLD) distribution given L-skew set as $\tau_3 = 0$ and optimized by `pargld` with `pargld(..., aux="tau5")` with $\tau_5 = 0$. The table stems from `inst/doc/SysDataBuilder02.R`. The table `gld_byt5opt` is used in the line drawing by `plotlmrdia46` in relation to the argument therein of `nogld_byt5opt`. This is the trajectory of the symmetrical GLD having constant L-scale (λ_2); this is different than the structurally similar by not identical Tukey Lambda distribution.

`gld_byt6tukeylam` : τ_4 and τ_6 of the Generalized Lambda distribution given L-skew set as $\tau_3 = 0$ and optimized by `pargld` with `pargld(..., aux="tau6")` with $\tau_6(\tau_4)$ (τ_6 as a function of τ_4, see `gld_byt6tukeylam` table). The table stems from `inst/doc/SysDataBuilder02.R`. The `gld_byt6tukeylam` is used in the line drawing by `plotlmrdia46` in relation to the argument therein of `notukey`. This relation between $\{\tau_4, \tau_6\}$ is that of the Tukey Lambda distribution; this is the trajectory of the symmetrical GLD having nonconstant L-scale (λ_2).

`nor` : τ_4 and τ_6 of the Normal distribution. The table stems from `inst/doc/SysDataBuilder02.R`. The `nor` is used in the point drawing by `plotlmrdia46`.

`pdq4` : τ_4 and τ_6 of the Polynomial Density-Quantile4 distribution, which implicitly is symmetrical, and therefore L-skew set as $\tau_3 = 0$. The table stems from `inst/doc/SysDataBuilder02.R`. The `pdq4` is used in the line drawing by `plotlmrdia46`.

`pwrexp` : τ_4 and τ_6 of the Power Exponential distribution of which the Asymmetric Exponential Power distribution (see also `lmomaep4`). The lookup table archive in the **lmomco** package for the Power Exponential (`PowerExponential.txt`) is confirmed to match the computation in `aep4` based on the AEP4 instead. The table stems from `inst/doc/SysDataBuilder02.R`.

`st2` : τ_4 and τ_6 of the well-known Student t distribution. The lookup table archive in the **lmomco** package for the Student t (`StudentT.txt`) is confirmed to match the computation in `st3` based on the ST3 instead. The table stems from `inst/doc/SysDataBuilder02.R`. The `st3` and not `st2` is used in the line drawing by `plotlmrdia46`.

`st3` : τ_4 and τ_6 of the Student 3t distribution (`lmomst3`). The table stems from `inst/doc/SysDataBuilder02.R`. The `st3` and not `st2` is used in the line drawing by `plotlmrdia46`.

symstable
: τ_4 and τ_6 of the Stable distribution, which is not otherwise supported in **lmomco**. The lookup table archive in the **lmomco** package for the Symmetrical Stable distribution is `StableDistribution.txt`. The table stems from `inst/doc/SysDataBuilder02.R`. The `symstable` is used in the line drawing by `plotlmrdia46`.

tukeylam
: (reference copy of `gld_byt6tukeylam`) τ_4 and τ_6 of the Tukey Lambda distribution (https://en.wikipedia.org/wiki/Tukey_lambda_distribution) that is not supported per se in **lmomco** because the Generalized Lambda distribution is instead. The `SymTukeyLambda.txt` is the lookup table archive in the **lmomco** package for the Tukey Lambda distribution confirmed to match the mathematics shown herein. The measure $L - scale$ or the second L-moment is not constant for the Symmetric Tukey Lambda as formulated. So, the trajectory of this distribution is not for a constant L-scale, which is unlike that for the Generalized Lambda. The table stems from `inst/doc/SysDataBuilder02.R`. The `tukeylam` is used in the line drawing by `plotlmrdia46`.

cau
: $\tau_4^{(1)} = 0.34280842$ and $\tau_6^{(1)} = 0.20274358$ (trim=1 TL-moments) of the Cauchy distribution (TL-moment [trim=1]) (see **Examples** `lmomcau` for source).

sla
: $\tau_4^{(1)} = 0.30420472$ and $\tau_6^{(1)} = 0.18900723$ (trim=1 TL-moments) of the Slash distribution (TL-moment [trim=1]) (see **Examples** `lmomsla` for source).

Author(s)

W.H. Asquith

See Also

`plotlmrdia46`, `lmrdia`

Examples

```
lratios <- lmrdia46()
```

`lmrdiscord` *Compute Discordance on L-CV, L-skew, and L-kurtosis*

Description

This function computes the Hosking and Wallis discordancy of the first three L-moment ratios (L-CV, L-skew, and L-kurtosis) according to their implementation in Hosking and Wallis (1997) and earlier. Discordancy triplets of these L-moment ratios is heuristically measured by effectively locating the triplet from the mean center of the 3-dimensional cloud of values. The **lmomRFA** provides for discordancy embedded in the "L-moment method" of regional frequency analysis. The author of **lmomco** chooses to have a separate "high level" implementation for emergent ideas of his in evaluating unusual sample distributions outside of the `regdata` object class envisioned by Hosking in the **lmomRFA** package.

Let $\boldsymbol{\mu_i}$ be a row vector of the values of $\tau_2^{[i]}, \tau_3^{[i]}, \tau_4^{[i]}$ and these are the L-moment ratios for the ith group or site out of n sites. Let $\overline{\boldsymbol{\mu}}$ be a row vector of mean values of all the n sites. Defining a sum of squares and cross products 3×3 matrix as

$$\boldsymbol{S} = \sum_{i}^{n} (\boldsymbol{\mu} - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu} - \overline{\boldsymbol{\mu}})^T$$

compute the discorancy of the ith site as

$$D_i = \frac{n}{3}(\boldsymbol{\mu} - \overline{\boldsymbol{\mu}})^T \boldsymbol{S}^{-1}(\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}.)$$

The L-moments of a sample for a location are judged to be discordance if D_i exceeds a critical value. The critical value is a function of sample size. Hosking and Wallis (1997, p. 47) provide a table for general application. By about $n = 14$, the critical value is taken as $D_c = 3$, although the D_{max} increases with sample size. Specifically, the D_i has an upper limit of

$$D_i \leq (n-1)/3.$$

However, Hosking and Wallis (1997, p. 47) recommend "that any site with $D_i > 3$ be regarded as discordant." A statistical test of D_i can be constructed. Hosking and Wallis (1997, p. 47) report that the $D_{critical}$ is

$$D_{critical,n,\alpha} = \frac{(n-1)Z}{n-4+3Z},$$

where

$$Z = F(\alpha/n, 3, n-4),$$

upper-tail quantile of the F distribution with degrees of freedom 3 and $n-4$. A table of critical values is preloaded into the `lmrdiscord` function as this mimics the table of Hosking and Wallis (1997, table 3.1) as a means for cross verification. This table corresponds to an $\alpha = 0.1$ significance.

Usage

```
lmrdiscord(site=NULL, t2=NULL, t3=NULL, t4=NULL,
           Dcrit=NULL, digits=4, lmrdigits=4, sort=TRUE,
           alpha1=0.10, alpha2=0.01, ...)
```

Arguments

`site`	An optional group or site identification; it will be sequenced from 1 to n if `NULL`.
`t2`	L-CV values; emphasis that L-scale is not used.
`t3`	L-skew values.
`t4`	L-kurtosis values.
`Dcrit`	An optional (user specified) critical value for discordance. This value will override the Hosking and Wallis (1997, table 3.1) critical values.
`digits`	The number of digits in rounding operations.
`lmrdigits`	The numer of digits in rounding operation for the echo of the L-moment ratios.
`sort`	A logical on the sort status of the returned data frame.

alpha1	A significance level that is greater (less significant, although in statistics we need to avoid assigning less or more in this context) than alpha2.
alpha2	A significance level that is less (more significant, although in statistics we need to avoid assigning less or more in this context) than alpha1.
...	Other arguments that might be used. The author added these because it was found that the function was often called by higher level functions that aggregated much of the discordance computations.

Value

An R data.frame is returned.

site	The group or site identification as used by the function.
t2	L-CV values.
t3	L-skew values.
t4	L-kurtosis.
Dmax	The maximum discordancy $D_{max} = (n-1)/3$.
Dalpha1	The critical value of D for $\alpha_1 = 0.10$ (default) significance as set by alpha1 argument.
Dalpha2	The critical value of D for $\alpha_2 = 0.01$ (default) significance as set by alpha1 argument.
Dcrit	The critical value of discordancy (user or tabled).
D	The discordancy of the L-moment ratios used to trigger the logical in isD.
isD	Are the L-moment ratios discordant (if starred).
signif	A hyphen, star, or double star based on the Dalpha1 and Dalpha2 values.

Author(s)

W.H. Asquith

Source

Consultation of the lmomRFA.f and regtst() function of the **lmomRFA** R package by J.R.M. Hosking. Thanks Jon and Jim Wallis for such a long advocation of the discordancy issue that began at least as early as the 1993 Water Resources Research Paper (-wha).

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmoms

Examples

```
## Not run:
# This is the canonical test of lmrdiscord().
library(lmomRFA) # Import lmomRFA, needs lmom package too
data(Cascades)   # Extract Hosking's data use in his examples
data <- as.regdata(Cascades) # A "regional" data structure
Dhosking <- sort(regtst(data)$D, decreasing=TRUE) # Discordancy

Dlmomco <- lmrdiscord(site=data$name, t2=data$t, t3=data$t_3, t4=data$t_4)

Dasquith <- Dlmomco$D
# Now show the site id, and the two discordancy computations
print(data.frame(NAME=data$name, Dhosking=Dhosking,
                                 Dasquith=Dasquith))
# The Dhosking and Dasquith columns had better match!

set.seed(3) # This seed produces a "*" and "**", but users
# are strongly encouraged to repeat the folowing code block
# over and over with an unspecified seed and look at the table.
n <- 30 # simulation sample size
par1 <- lmom2par(vec2lmom(c(1, .23, .2, .1)), type="kap")
par2 <- lmom2par(vec2lmom(c(1, .5, -.1)),      type="gev")
name <- t2 <- t3 <- t4 <- vector(mode="numeric")
for(i in 1:20) {
  X <- rlmomco(n, par1); lmr <- lmoms(X)
  t2[i] <- lmr$ratios[2]
  t3[i] <- lmr$ratios[3]
  t4[i] <- lmr$ratios[4]
  name[i] <- "kappa"
}
j <- length(t2)
for(i in 1:3) {
  X <- rlmomco(n, par2); lmr <- lmoms(X)
  t2[j + i] <- lmr$ratios[2]
  t3[j + i] <- lmr$ratios[3]
  t4[j + i] <- lmr$ratios[4]
  name[j + i] <- "gev"
}
D <- lmrdiscord(site=name, t2=t2, t3=t3, t4=t4)
print(D)

plotlmrdia(lmrdia(), xlim=c(-.2,.6), ylim=c(-.1, .4),
           autolegend=TRUE, xleg=0.1, yleg=.4)
points(D$t3,D$t4)
text(D$t3,D$t4,D$site, cex=0.75, pos=3)
text(D$t3,D$t4,D$D, cex=0.75, pos=1) #
## End(Not run)
```

---

lmrloc — *Line-of-Organic Correlation*

Description

Compute the line-of-organic correlation (LOC) (Helsel and others, 2020, sec. 10.2.2, p. 280). The LOC is estimated by both L-moments and product moments. The LOC has other names in the literature including reduced major axis and line of diagonal correlation. When describing a functional relations between two variables without trying to predict one from the other, LOC is more appropriate than ordinary least squares (OLS).

The LOC is a regression line whose slope is computed by the ratio between respective variations of the predictor variable and the response variable. The intercept of the line is computed such that the line passes through the familiar arithmetic mean (first L-moment) (λ_1) each for the two variables. Relative variation is readily computed by the ratio of standard deviations or for more robust and less biased estimation by the ratio of the L-variations (second L-moment) (λ_2) of the two variables.

The λ_2 is generically based on the so-called Gini mean difference statistic (GMD) ($\mathcal{G}$) by $\lambda_2 = \mathcal{G}/2$ (`gini.mean.diff`). Incidentally for the normal distribution, the well-known standard deviation is the product $\lambda_2\sqrt{\pi}$ (see also `lmomnor`). Mathematically, GMD is defined as the linear combination

$$\mathcal{G} = \frac{2}{n(n-1)}\sum_{i=1}^{n}(2i-n-1)x_{i:n},$$

where $x_{i:n}$ are the sample ascending order statistics.

Returning to the need to estimate the LOC slope, algebra shows the slope is the ratio of the $\mathcal{G}$ values as

$$m = \mathrm{sign}[\rho]\cdot\frac{\sum_{i=1}^{n}(2i-n-1)X_{i:n}}{\sum_{i=1}^{n}(2i-n-1)Y_{i:n}},$$

where $X_{i:n}$ is an ordered (ascending) vector of random variable X, $Y_{i:n}$ is an ordered (ascending) vector of random variable Y, and the slope sign can be computed by a correlation coefficient sign (Pearson R, Kendall Tau [computationally slowest], Spearman Rho would all work [implemented for the function, ρ]). For applications, it is critical that the correlation coefficient is computed using the original correlated-ordering of X and Y and not after individual vector sorting that is needed for the GMD (L-moments). A developer, therefore, must be cognizant of the placement in code when the two variables are sorted to the order statistics for $\mathcal{G}$ computations.

The LOC intercept is given by algebra by

$$b = \frac{1}{n}\left(\sum_{i=1}^{n}X_{i:n} - m\cdot\sum_{i=1}^{n}Y_{i:n}\right).$$

Helsel and others (2020, p. 281) enumerate some advantages to the use of the LOC: (1) it minimizes errors in both `x` and `y` directions, (2) it provides a single line regardless of which variable (x or y) is used as the response variable, and (3) its cumulative distribution function of the predictions, including the variance and probabilities, is correct (meaning not compressed as in OLS). The LOC is particularly useful for modeling the intrinsic functional relation between two variables, both of which are measured with error and (or) when neither variable is considered an independent variable appropriate to predict the other.

Usage

```
lmrloc(x, y=NULL, terse=TRUE)
```

Arguments

x	A numeric vector, matrix or data frame.
y	NULL (default) or a vector of same length of x.
terse	A logical triggering only return of the coefficients of the two lines; otherwise, the intermediate computations are also returned.

Value

An R list is returned with terse=TRUE with two vectors of the intercept and slope coefficients for the L-moment and the product moment versions. The names on the vectors, respectively, are "LMR_Intercept", "LMR_Slope" and "PMR_Intercept", "PMR_Slope" for LMR (L-moment ratio) and PMR (product moment ratio) are monikers for the two approaches. An expanded R list is returned with terse=FALSE with the intermediate computations also provided.

loc_lmr	The LOC by L-moments (L-variations or equivalently Gini Mean Differences).
loc_pmr	The LOC by product moments (standard deviations).
srho	The sign on Spearman Rho.
mu_x	The arithmetic mean of the x variable.
mu_y	The arithmetic mean of the y variable.
gini_x	The GMD of the x variable.
gini_y	The GMD of the y variable.
sd_x	The standard deviation of the x variable.
sd_y	The standard deviation of the y variable.

Author(s)

W.H. Asquith

References

Helsel, D.R., Hirsch, R.M., Ryberg, K.R., Archfield, S.A., and Gilroy, E.J., 2020, Statistical methods in water resources: U.S. Geological Survey Techniques and Methods, book 4, chap. A3, 458 p., doi:10.3133/tm4a3.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Jurečková, J., and Picek, J., 2006, Robust statistical methods with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 1–58488–454–1.

See Also

gini.mean.diff

Examples

```
n <- 100; x <- rnorm(n); y <- -0.4 * x + rnorm(n, sd=0.2)
y[x == min(x)] <- 2 * min(y) # throw in an outlier to help separate two lines
loc <- lmrloc(x, y, terse=FALSE)
plot(x, y)
abline(loc$loc_lmr, lty=1)
abline(loc$loc_pmr, lty=2)
legend("topright", c("LOC by L-moments", "LOC by product moments"), lty=c(1,2))

olsxy <- 1 / coefficients(stats::lm(x~y))[2] # yes inversion needed to show
olsyx <-     coefficients(stats::lm(y~x))[2] # geometric mean in proper way
mstar <- loc$srho * sqrt(abs(olsxy) * abs(olsyx)); names(mstar) <- NULL
m_pmr <- loc$loc_pmr[2]; names(m_pmr) <- NULL
m_lmr <- loc$loc_lmr[2]; names(m_lmr) <- NULL
message("Geometric mean OLS slopes = ", mstar) # see that these two are
message("           PMR LOC slope  = ", m_pmr) # equivalent by theory
message("           LMR LOC slope  = ", m_lmr) # this one is not
```

`lrv2prob` *Convert a Vector of Logistic Reduced Variates to Annual Nonexceedance Probabilities*

Description

This function converts a vector of logistic reduced variates (lrv) to annual nonexceedance probabilities F

$$F = -\log((1 - lrv)/lrv),$$

where $0 \leq F \leq 1$.

Usage

```
lrv2prob(lrv)
```

Arguments

`lrv` A vector of logistic reduced variates.

Value

A vector of annual nonexceedance probabilities.

Author(s)

W.H. Asquith

References

Bradford, R.B., 2002, Volume-duration growth curves for flood estimation in permeable catchments: Hydrology and Earth System Sciences, v. 6, no. 5, pp. 939–947.

See Also

prob2lrv, prob2T

Examples

```
T <- c(1, 2, 5, 10, 25, 50, 100, 250, 500); lrv <- prob2grv(T2prob(T))
F <- lrv2prob(lrv)
```

---

lrzlmomco *Lorenz Curve of the Distributions*

Description

This function computes the Lorenz Curve for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair et al. (2013, p. 174) as

$$L(u) = \frac{1}{\mu}\int_0^u x(p)\;\mathrm{d}p,$$

where $L(u)$ is the Lorenz curve for nonexceedance probability u. The Lorenz curve is related to the Bonferroni curve ($B(u)$, bfrlmomco) by

$$L(u) = \mu B(u).$$

Usage

```
lrzlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.

Value

Lorzen curve value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, bfrlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
f <- c(0.25, 0.75) # Both computations report: 0.02402977 and 0.51653731
Lu1 <-   lrzlmomco(f, A)
Lu2 <- f*bfrlmomco(f, A)

# The Lorenz curve is related to the Gini index (G), which is L-CV:
"afunc" <- function(u) { return(lrzlmomco(f=u, A)) }
L <- integrate(afunc, lower=0, upper=1)$value
G <- 1 - 2*L                                                  # 0.4129159
G <- 1 - expect.min.ostat(2,para=A,qua=quagov)*cmlmomco(f=0,A)  # 0.4129159
LCV <- lmomgov(A)$ratios[2]                                   # 0.41291585
```

---

mle2par *Use Maximum Likelihood to Estimate Parameters of a Distribution*

Description

This function uses the method of maximum likelihood (MLE) to estimate the parameters of a distribution. MLE is a straightforward optimization problem that is formed by maximizing the sum of the logarithms of probability densities. Let Θ represent a vector of parameters for a candidate fit to the specified probability density function $g(x|\Theta)$ and x_i represent the observed data for a sample of size n. The objective function is

$$\mathcal{L}(\Theta) = -\sum_{i=1}^{n} \log\, g(x_i|\Theta),$$

where the Θ for a maximized $-\mathcal{L}$ (note the 2nd negation for the adjective "maximized", `optim()` defaults as a minimum optimizer) represents the parameters fit by MLE. The initial parameter estimate by default will be seeded by the method of L-moments.

Usage

```
mle2par(x, type, init.para=NULL, silent=TRUE, null.on.not.converge=TRUE,
                 ptransf=  function(t) return(t),
                 pretransf=function(t) return(t), ...)
```

Arguments

`x`	A vector of data values.
`type`	Three character (minimum) distribution type (for example, `type="gev"`), see `dist.list`.
`init.para`	Initial parameters as a vector Θ or as an **lmomco** parameter "object" from say `vec2par`. If a vector is given, then internally `vec2par` is called with distribution equal to `type`.
`silent`	A logical to silence the `try()` function wrapping the `optim()` function.

`null.on.not.converge`
: A logical to trigging simple return of `NULL` if the `optim()` function returns a nonzero convergence status.

`ptransf`
: An optional parameter transformation function (see **Examples**) that is useful to guide the optimization run. For example, suppose the first parameter of a three parameter distribution resides in the positive domain, then `ptransf(t) = function(t) c(log(t[1]), t[2], t[3])`.

`pretransf`
: An optional parameter retransformation function (see **Examples**) that is useful to guide the optimization run. For example, suppose the first parameter of a three parameter distribution resides in the positive domain, then `pretransf(t) = function(t) c(exp(t[1]), t[2], t[3])`.

`...`
: Additional arguments for the `optim()` function and other uses.

Value

An R `list` is returned. This list should contain at least the following items, but some distributions such as the `revgum` have extra.

`type` The type of distribution in three character (minimum) format.

`para` The parameters of the distribution.

`source` Attribute specifying source of the parameters.

`AIC` The Akaike information criterion (AIC).

`optim` The returned `list` of the `optim()` function.

Note

During the optimization process, the function requires evaluation at the initial parameters. The following error rarely will be seen:

```
Error in optim(init.para$para, afunc) :
  function cannot be evaluated at initial parameters
```

if `Inf` is returned on first call to the objective function. The `silent` by default though will silence this error. Alternative starting parameters might help. This function is not built around subordinate control functions to say keep the parameters within distribution-specific bounds. However, in practice, the L-moment estimates should already be fairly close and the optimizer can take it from there. More sophisticated MLE for many distributions is widely available in other R packages. The **lmomco** package uses its own probability density functions.

Author(s)

W.H. Asquith

See Also

`lmom2par`, `mps2par`, `tlmr2par`

Examples

```
## Not run:
# This example might fail on mle2par() or mps2par() depending on the values
# that stem from the simulation. Trapping for a NULL return is not made here.
father <- vec2par(c(37,25,114), type="st3"); FF <- nonexceeds(); qFF <- qnorm(FF)
X <- rlmomco(78, father) # rerun if MLE and MPS fail to get a solution
plot(qFF,  qlmomco(FF, father), type="l", xlim=c(-3,3),
     xlab="STANDARD NORMAL VARIATE", ylab="QUANTILE") # parent (black)
lines(qFF, qlmomco(FF, lmr2par(X, type="gev")), col="red"  ) # L-moments (red)
lines(qFF, qlmomco(FF, mps2par(X, type="gev")), col="green") #     MPS (green)
lines(qFF, qlmomco(FF, mle2par(X, type="gev")), col="blue" ) #     MLE  (blue)
points(qnorm(pp(X)), sort(X)) # the simulated data
## End(Not run)

## Not run:
# REFLECTION SYMMETRY
set.seed(451)
X <- rlmomco(78, vec2par(c(2.12, 0.5, 0.6), type="pe3"))
# MLE and MPS are almost reflection symmetric, but L-moments always are.
mle2par( X, type="pe3")$para #  2.1796827 0.4858027  0.7062808
mle2par(-X, type="pe3")$para # -2.1796656 0.4857890 -0.7063917
mps2par( X, type="pe3")$para #  2.1867551 0.5135882  0.6975195
mps2par(-X, type="pe3")$para # -2.1868252 0.5137325 -0.6978034
parpe3(lmoms( X))$para       #  2.1796630 0.4845216  0.7928016
parpe3(lmoms(-X))$para       # -2.1796630 0.4845216 -0.7928016
## End(Not run)

## Not run:
Ks <- seq(-1,+1,by=0.02); n <- 100; MLE <- MPS <- rep(NA, length(Ks))
for(i in 1:length(Ks)) {
  sdat   <- rlmomco(n, vec2par(c(1,0.2,Ks[i]), type="pe3"))
  mle    <- mle2par(sdat, type="pe3")$para[3]
  mps    <- mps2par(sdat, type="pe3")$para[3]
  MLE[i] <- ifelse(is.null(mle), NA, mle) # A couple of failures expected as NA's.
  MPS[i] <- ifelse(is.null(mps), NA, mps) # Some amount fewer failures than MLE.
}
plot( MLE, MPS, xlab="SKEWNESS BY MLE", ylab="SKEWNESS BY MPS")#
## End(Not run)

## Not run:
# Demonstration of parameter transformation and retransformation
set.seed(9209) # same seed used under mps2par() in parallel example
x <- rlmomco(500, vec2par(c(1,1,3), type="gam")) # 3-p Generalized Gamma
guess <- lmr2par(x, type="gam", p=3) # By providing a 3-p guess the 3-p
# Generalized Gamma will be triggered internally. There are problems passing
# "p" argument to optim() if that function is to pick up the ... argument.
mle2par(x, type="gam", init.para=guess, silent=FALSE,
           ptransf=  function(t) { c(log(t[1]), log(t[2]), t[3])},
           pretransf=function(t) { c(exp(t[1]), exp(t[2]), t[3])})$para
# Reports:       mu     sigma        nu   for some simulated data.
#         1.0341269 0.9731455 3.2727218
## End(Not run)
```

```
## Not run:
# Demonstration of parameter estimation with tails of density zero, which
# are intercepted internally to maintain finiteness. We explore the height
# distribution for male cats of the cats dataset from the MASS package and
# fit the generalized lambda. The log-likelihood is shown by silent=FALSE
# to see that the algorithm converges slowly. It is shown how to control
# the relative tolerance of the optim() function as shown below and
# investigate the convergence by reviewing the five fits to the data.
FF <- nonexceeds(sig6=TRUE); qFF <- qnorm(FF)
library(MASS); data(cats); x <- cats$Hwt[cats$Sex == "M"]
p2 <- mle2par(x, type="gld", silent=FALSE, control=list(reltol=1E-2))
p3 <- mle2par(x, type="gld", silent=FALSE, control=list(reltol=1E-3))
p4 <- mle2par(x, type="gld", silent=FALSE, control=list(reltol=1E-4))
p5 <- mle2par(x, type="gld", silent=FALSE, control=list(reltol=1E-5))
p6 <- mle2par(x, type="gld", silent=FALSE, control=list(reltol=1E-6))
plot( qFF,  quagld(FF, p2), type="l", col="black",  # see poorest fit
      xlab="Standard normal variable", ylab="Quantile")
points(qnorm(pp(x)), sort(x), lwd=0.6, col=grey(0.6))
lines(qFF,  quagld(FF, p3), col="red"    )
lines(qFF, par2qua(FF, p4), col="green"  )
lines(qFF,  quagld(FF, p5), col="blue"   )
lines(qFF, par2qua(FF, p6), col="magenta") #
## End(Not run)
```

---

mps2par *Use Maximum Product of Spacings to Estimate the Parameters of a Distribution*

Description

This function uses the method of maximum product of spacings (MPS) (maximum spacing estimation or maximum product of spacings estimation) to estimate the parameters of a distribution. MPS is based on maximization of the *geometric mean* of probability spacings in the data where the spacings are defined as the differences between the values of the cumulative distribution function, $F(x)$, at sequential data indices.

MPS (Dey *et al.*, 2016, pp. 13–14) is an optimization problem formed by maximizing the geometric mean of the spacing between consecutively ordered observations standardized to a U-statistic. Let Θ represent a vector of parameters for a candidate fit of $F(x|\Theta)$, and let $U_i(\Theta) = F(X_{i:n}|\Theta)$ be the nonexceedance probabilities of the observed values of the order statistics $x_{i:n}$ for a sample of size n. Define the differences

$$D_i(\Theta) = U_i(\Theta) - U_{i-1}(\Theta) \text{ for } i = 1, \ldots, n+1,$$

with the additions to the vector U of $U_0(\Theta) = 0$ and $U_{n+1}(\Theta) = 1$. The objective function is

$$M_n(\Theta) = -\sum_{i=1}^{n+1} \log\, D_i(\Theta),$$

where the Θ for a maximized $-M_n$ represents the parameters fit by MPS. Some authors to keep with the idea of geometric mean include factor of $1/(n+1)$ for the definition of M_n. Whereas other authors (Shao and Hahn, 1999, eq. 2.0), show

$$S_n(\Theta) = (n+1)^{-1} \sum_{i=1}^{n+1} \log[(n+1)D_i(\Theta)].$$

So it seems that some care is needed when considering the implementation when the value of "the summation of the logarithms" is to be directly interpreted. Wong and Li (2006) provide a salient review of MPS in regards to an investigation of maximum likelihood (MLE), MPS, and probability-weighted moments (`pwm`) for the GEV (`quagev`) and GPA (`quagpa`) distributions. Finally, Soukissian and Tsalis (2015) also study MPS, MLE, L-moments, and several other methods for GEV fitting.

If the initial parameters have a support inside the range of the data, infinity is returned immediately by the optimizer and further action stops and the parameters returned are `NULL`. For the implementation here, if `check.support` is true, and the initial parameter estimate (if not provided and acceptable by `init.para`) by default will be seeded through the method of L-moments (unbiased, `lmoms`), which should be close and convergence will be fairly fast if a solution is possible. If these parameters can not be used for spinup, the implementation will then attempt various probability-weighted moment by plotting position (`pwm.pp`) converted to L-moments (`pwm2lmom`) as part of an extended attempt to find a support of the starting distribution encompass the data. Finally, if that approach fails, a last ditch effort using starting parameters from maximum likelihood computed by a default call to `mle2par` is made. Sometimes data are pathological and user supervision is needed but not always successful—MPS can show failure for certain samples and(or) choice of distribution.

It is important to remark that the support of a fitted distribution is not checked within the loop for optimization once spun up. The reasons are twofold: (1) The speed hit by repeated calls to `supdist`, but in reality (2) PDFs in **lmomco** are supposed to report zero density for outside the support of a distribution (see NEWS) and for the $-\log(D_i(\Theta) \rightarrow 0) \rightarrow \infty$ and hence infinity is returned for that state of the optimization loop and alternative solution will be tried.

As a note, if all U are equally spaced, then $|M(\Theta)| = I_o = (n+1)\log(n+1)$. This begins the concept towards goodness-of-fit. The $M_n(\Theta)$ is a form of the Moran-Darling statistic for goodness-of-fit. The $M_n(\Theta)$ is a Normal distribution with

$$\mu_M \approx (n+1)[\log(n+1) + \gamma] - \frac{1}{2} - \frac{1}{12(n+1)},$$

$$\sigma_M \approx (n+1)\left(\frac{\pi^2}{6} - 1\right) - \frac{1}{2} - \frac{1}{6(n+1)},$$

where $\gamma \approx 0.577221$ (Euler–Mascheroni constant, `-digamma(1)`) or as the definite integral

$$\gamma^{\mathrm{Euler}}_{\mathrm{Mascheroni}} = -\int_0^\infty \exp(-t)\log(t)\;\mathrm{d}t,$$

An extension into small samples using the Chi-Square distribution is

$$A = C_1 + C_2 \times \chi^2_n,$$

where

$$C_1 = \mu_M - \sqrt{\frac{\sigma^2_M\, n}{2}} \text{ and } C_2 = \sqrt{\frac{\sigma^2_M}{2n}},$$

and where χ^2_n is the Chi-Square distribution with n degrees of freedom. A test statistic is

$$T(\Theta) = \frac{M_n(\Theta) - C_1 + \frac{p}{2}}{C_2},$$

where the term $p/2$ is a bias correction based on the number of fitted distribution parameters p. The null hypothesis that the fitted distribution is correct is to be rejected if $T(\Theta)$ exceeds a critical value from the Chi-Square distribution. The MPS method has a relation to maximum likelihood (mle2par) and the two are asymptotically equivalent.

Important Remark Concerning Ties—Ties in the data cause *instant degeneration* with MPS and must be mitigated for and thus attention to this documentation and even the source code itself is required.

Usage

```
mps2par(x, type, init.para=NULL, ties=c("bernstein", "rounding", "density"),
            delta=0, log10offset=3, get.untied=FALSE, check.support=TRUE,
            moran=TRUE, silent=TRUE, null.on.not.converge=TRUE,
            ptransf=  function(t) return(t),
            pretransf=function(t) return(t),
            mle2par=TRUE, ...)
```

Arguments

`x`	A vector of data values.
`type`	Three character (minimum) distribution type (for example, `type="gev"`, see `dist.list`).
`init.para`	Initial parameters as a vector Θ or as an **lmomco** parameter "object" from say `vec2par`. If a vector is given, then internally `vec2par` is called with distribution equal to `type`.
`ties`	Ties cause degeneration in the computation of $M(\Theta)$: Option `bernstein` triggers a smoothing of only the ties using the `dat2bernqua` function—Bernstein-type smoothing for ties is likely near harmless when ties are near the center of the distribution, but of course caution is advised if ties exist near the extremal values; the settings for `log10offset` and `delta` are ignored if `bernstein` is selected Also for a tie-run having an odd number of elements, the middle tied value is left as original data. Option `rounding` triggers two types of adjustment: if `delta > 0` then a round-off error approach inspired by Cheng and Stephens (1989, eq. 4.1) is used (see **Note**) and `log10offset` is ignored, but if `delta=0`, then `log10offset` is picked up as an order of magnitude offset (see **Note**). Use of options `log10offset` and `delta` are likely to not keep a middle unmodified in an odd-length, tie-run in contrast to use of `bernstein`. Option `density` triggers the substitution of the probability density $g(x_{i:n}\|\Theta)$ at the ith tie from the current fit of the distribution. **Warning**—It appears that inference is lost almost immediately because the magnitude of M_n losses meaning because probability densities are not in the same scale as changes in probabilities exemplified by the D_i. This author has not yet found literature discussing this, but density substitution is a recognized strategy.

`delta` The optional δ value if $\delta > 0$ and if `ties="rounding"`.

`log10offset` The optional base-10 logarithmic offset approach to roundoff errors if `delta=0` and if `ties="rounding"`.

`get.untied` A logical to populate a `ties` element in the returned `list` with the untied-pseudo data as it was made available to the optimizer and the number of iternations required to exhaust all ties. An emergency break it implemented if the number of iterations appears to be blowing up.

`check.support` A logical to trigger a call to `supdist` to compute the support of the distribution at the initial parameters. As mentioned, MPS degenerates if `min(x)` < the lower support or if `max(x)` > the upper support. Regardless of the setting of `check.support` and `NULL` will be returned because this is what the optimizer will do anyway.

`moran` A logical to trigger the goodness-of-fit test described previously.

`silent` A logical to silence the `try()` function wrapping the `optim()` function and to provide a returned list of the optimization output.

`null.on.not.converge` A logical to trigging simple return of `NULL` if the `optim()` function returns a nonzero convergence status.

`ptransf` An optional parameter transformation function (see **Examples**) that is useful to guide the optimization run. For example, suppose the first parameter of a three parameter distribution resides in the positive domain, then `ptransf(t) = function(t) c(log(t[1]), t[2], t[3])`.

`pretransf` An optional parameter retransformation function (see **Examples**) that is useful to guide the optimization run. For example, suppose the first parameter of a three parameter distribution resides in the positive domain, then `pretransf(t) = function(t) c(exp(t[1]), t[2], t[3])`.

`mle2par` A logical to turn off the potential last attempt at maximum likelihood estimates of a valid seed as part of `check.support=TRUE`.

`...` Additional arguments for the `optim()` function and other uses.

Value

An R `list` is returned. This list should contain at least the following items, but some distributions such as the `revgum` have extra.

`type` The type of distribution in three character (minimum) format.

`para` The parameters of the distribution.

`source` Attribute specifying source of the parameters.

`init.para` The initial parameters. Warning to users, when inspecting returned values make sure that one is referencing the MPS parameters in `para` and not those shown in `init.para`!

`optim` An optional `list` of returned content from the optimizer if not `silent`.

`ties` An optional `list` of untied-pseudo data and number of iterations required to achieve no ties (usually unity!) if and only if there were ties in the original data, `get.untied` is true, and `ties != "density"`.

MoranTest An optional `list` of returned values that will include both diagnostics and statistics. The diagnostics are the computed $\mu_M(n)$, $\sigma^2_M(n)$, C_1, C_2, and n. The statistics are the minimum value I_o theoretically attainable $|M_n(\Theta)|$ for equally spaced differences, the minimized value $M_n(\Theta)$, the $T(\Theta)$, and the corresponding `p.value` from the upper tail of the χ^2_n distribution.

Note

During optimization, the objective function requires evaluation at the initial parameters and must be finite. If `Inf` is returned on first call to the objective function, then a warning like this

```
optim() attempt is NULL
```

should be seen. The `silent` by default though will silence this error. Error trapping for the estimated support of the distribution from the initial parameter values is made by `check.support=TRUE` and verbose warnings given to help remind the user. Considerable attempt is made internally to circumvent the appearance of the above error.

More specifically, an MPS solution degenerates when the fitted distribution has a narrower support than the underlying data and artificially "ties" show up within the objective function even if the original data lacked ties or were already mitigated for. The user's only real recourse is to try fitting another distribution either by starting parameters or even distribution type. Situations could arise for which carefully chosen starting parameters could permit the optimizer to keep its simplex within the viable domain. The MPS method is sensitive to tails of a distribution having asymoptic limits as $F \rightarrow 0^+$ or $F \rightarrow 1^-$.

The Moran test can be quickly checked with highly skewed and somewhat problematic data by

```
# CPU intensive experiment
gev <- vec2par(c(4,0.3,-0.2), type="gev"); nsim <- 5000
G <- replicate(nsim, mps2par(rlmomco(100, gev), # extract the p-values
                             type="gev")$MoranTest$statistics[4])
G <- unlist(G) # unlisting required if NULLs came back from mps2par()
length(G[G <= 0.05])/length(G) # 0.0408 (!=0.05 but some fits not possible)
V <- replicate(nsim, mps2par(rlmomco(100, gev),
                             type="nor")$MoranTest$statistics[4])
V <- unlist(V) # A test run give 4,518 solutions
length(V[V <= 0.05])/length(V) # 0.820 higher because not gev used
W <- replicate(nsim, mps2par(rlmomco(100, gev),
                             type="glo")$MoranTest$statistics[4])
W <- unlist(W)
length(W[W <= 0.05])/length(W) # 0.0456 higher because not gev used but
# very close because of the proximity of the glo to the gev for the given
# L-skew of the parent: lmomgev(gev)$ratios[3] = 0.3051
```

Concerning round-off errors, the Cheng and Stephens (1989, eq. 4.1) approach is to assume that the round-off errors are $x \pm \delta$, compute the upper and lower probabilities f for $f_L \mapsto x - \delta$ and $f_U \mapsto x + \delta$, and then prorate the D_i in even spacings of $1/(r-1)$ where r is the number of tied values in a given tie-run. The approach for `mps2par` is similar but simplies the algorithm to evenly prorate the x values in a tie-run. In other words, the current implementation is to actually massage the data before passage into the optimizer. If the $\delta = 0$, a base-10 logarithmic approach will be used in which, the order of magnitude of the value in a tie-run is computed and the `log10offset` subtracted to approximate the roundoff but recognize that for skewed data the roundoff might be scale dependent. The default treats a tie of three $x_i = 15{,}000$ as $x_{i|r} = 14{,}965.50; 15{,}000.00; 15{,}034.58$. In either approach, an iterative loop is present to continue looping until no further ties are found—this is made to protect against the potential for the algorithm to create new ties. A sorted vector of the final data for the optimize is available in the `ties` element of the returned list if and only if ties were originally present, `get.untied=TRUE`, and `ties != "density"`. Ties and compensation likely these prorations can only make $M(\Theta)$ smaller, and hence the test becomes conservative.

A note of other MPS implementations in R is needed. The **fBasics** and **gld** packages both provide for MPS estimation for the generalized lambda distribution. The salient source files and code chunks are shown. First, consider package **fBasics**:

```
fBasics --> dist-gldFit.R --> .gldFit.mps -->
          f = try(-typeFun(log(DH[DH > 0])), silent = TRUE)
```

where it is seen that $D_i = 0$ are ignored! Such a practice does not appear efficacious during development and testing of the implementation in **lmomco**, parameter solutions very substantially different than reason can occur or even failure of convergence by the **fBasics** implementation. Further investigation is warranted. Second, consider package **gld**:

```
gld --> fit_fkml.R --> fit_fkml.c --> method.id == 2:
# If F[i]-F[i-1] = 0, replace by f[i-1]
#                      (ie the density at smaller observation)
```

which obviously make the density substitution for ties as well `ties="density"` for the implementation here. Testing indicates that viable parameter solutions will result with direct insertion of the density in the case of ties. Interference, however, of the M_n is almost assuredly to be greatly weakened or destroyed depending on the shape of the probability density function or a large number of ties. The problem is that the sum of the D_i are no longered ensured to sum to unity. The literature appears silent on this particular aspect of MPS, and further investigation is warranted.

The **eva** package provides MPS for GEV and GPD. The approach there does not appear to replace changes of zero by density but to insert a “smallness” in conjunction with other conditioning checking (only the `cond3` is shown below) and a curious penalty of `1e6`. The point is that different approaches have been made by others.

```
eva --> gevrFit --> method="mps"
cdf[(is.nan(cdf) | is.infinite(cdf))] <- 0
cdf <- c(0, cdf, 1); D <- diff(cdf); cond3 <- any(D < 0)
## Check if any differences are zero due to rounding and adjust
D <- ifelse(D <= 0, .Machine$double.eps, D)
if(cond1 | cond2 | cond3) { abs(sum(log(D))) + 1e6 } else { -sum(log(D)) }
```

Let us conclude with an example for the GEV between **eva** and **lmomco** and note sign difference in definition of the GEV shape but otherwise a general similarity in results:

```
X <- rlmomco(97, vec2par(c(100,12,-.5), type="gev"))
pargev(lmoms(X))$para
```

```
     #                  xi               alpha               kappa
     #         100.4015424          12.6401335          -0.5926457
eva::gevrFit(X, method="mps")$par.ests
     #Location (Intercept)    Scale (Intercept)    Shape (Intercept)
     #         100.5407709          13.5385491           0.6106928
```

Author(s)

W.H. Asquith

References

Cheng, R.C.H., Stephens, M.A., 1989, A goodness-of-fit test using Moran's statistic with estimated parameters: Biometrika, v. 76, no. 2, pp. 385–392.

Dey, D.K., Roy, Dooti, Yan, Jun, 2016, Univariate extreme value analysis, chapter 1, *in* Dey, D.K., and Yan, Jun, eds., Extreme value modeling and risk analysis—Methods and applications: Boca Raton, FL, CRC Press, pp. 1–22.

Shao, Y., and Hahn, M.G., 1999, Strong consistency of the maximum product of spacings estimates with applications in nonparametrics and in estimation of unimodal densities: Annals of the Institute of Statistical Mathematics, v. 51, no. 1, pp. 31–49.

Soukissian, T.H., and Tsalis, C., 2015, The effect of the generalized extreme value distribution parameter estimation methods in extreme wind speed prediction: Natural Hazards, v. 78, pp. 1777–1809.

Wong, T.S.T., and Li, W.K., 2006, A note on the estimation of extreme value distributions using maximum product of spacings: IMS Lecture Notes, v. 52, pp. 272–283.

See Also

lmom2par, mle2par, tlmr2par

Examples

```
## Not run:
pe3 <- vec2par(c(4.2, 0.2, 0.6), type="pe3") # Simulated values should have at least
X <- rlmomco(202, pe3); Xr  <- round(sort(X), digits=3) # one tie-run after rounding,
mps2par(X,  type="pe3")$para      # and the user can observe the (minor in this case)
mps2par(Xr, type="pe3")$para      # effect on parameters.
# Another note on MPS is needed. It is not reflection symmetric.
mps2par( X, type="pe3")$para
mps2par(-X, type="pe3")$para
## End(Not run)

## Not run:
# Use 1,000 replications for sample size of 75 and estimate the bias and variance of
# the method of L-moments and maximum product spacing (MPS) for the 100-year event
# using the Pearson Type III distribution.
set.seed(1596)
nsim <- 1000; n <- 75; Tyear <- 100; type <- "pe3"
parent.lmr <- vec2lmom(c(5.5, 0.15, 0.03))   # L-moments of the "parent"
parent  <- lmom2par(parent.lmr, type="pe3")  # "the parent"
```

```
Q100tru <- qlmomco(T2prob(Tyear), parent)    # "true value"
Q100lmr <- Q100mps <- rep(NA, nsim)          # empty vectors
T3lmr <- T4lmr <- T3mps <- T4mps <- rep(NA, nsim)
for(i in 1:nsim) { # simulate from the parent, compute L-moments
   tmpX <- rlmomco(n, parent); lmrX <- lmoms(tmpX)
   if(! are.lmom.valid(lmrX)) { # quiet check on viability
     lmrX <- pwm2lmom(pwms.pp(tmpX)) # try a pwm by plotting positions instead
     if(! are.lmom.valid(lmrX)) next
   }
   lmrpar <- lmom2par(lmrX, type=type)                  # Method of L-moments
   mpspar <-  mps2par(tmpX, type=type, init.para=lmrpar) # Method of MPS
   if(! is.null(lmrpar)) {
      Q100lmr[i] <- qlmomco(T2prob(Tyear), lmrpar); lmrlmr <- par2lmom(lmrpar)
      T3lmr[i] <- lmrlmr$ratios[3]; T4lmr[i] <- lmrlmr$ratios[4]
   }
   if(! is.null(mpspar)) {
      Q100mps[i] <- qlmomco(T2prob(Tyear), mpspar); mpslmr <- par2lmom(mpspar)
      T3mps[i] <- mpslmr$ratios[3]; T4mps[i] <- mpslmr$ratios[4]
   }
}
print(summary(Q100tru - Q100lmr)) # Method of L-moment   (mean = -0.00176)
print(summary(Q100tru - Q100mps)) # Method of MPS        (mean = -0.02746)
print(var(Q100tru - Q100lmr, na.rm=TRUE)) # Method of L-moments (0.009053)
print(var(Q100tru - Q100mps, na.rm=TRUE)) # Method of MPS       (0.009880)
# CONCLUSION: MPS is very competitive to the mighty L-moments.

LMR <- data.frame(METHOD=rep("Method L-moments",        nsim), T3=T3lmr, T4=T4lmr)
MPS <- data.frame(METHOD=rep("Maximum Product Spacing", nsim), T3=T3mps, T4=T4mps)
ZZ <- merge(LMR, MPS, all=TRUE)
boxplot(ZZ$T3~ZZ$METHOD, data=ZZ); mtext("L-skew Distributions")
boxplot(ZZ$T4~ZZ$METHOD, data=ZZ); mtext("L-kurtosis Distributions") #
## End(Not run)

## Not run:
# Data shown in Cheng and Stephens (1989). They have typesetting error on their
# "sigma." Results mu=34.072 and sigma=sqrt(6.874)=2.6218
H590 <- c(27.55, 31.82, 33.74, 34.15, 35.32, 36.78,
          29.89, 32.23, 33.74, 34.44, 35.44, 37.07,
          30.07, 32.28, 33.86, 34.62, 35.61, 37.36,
          30.65, 32.69, 33.86, 34.74, 35.61, 37.36,
          31.23, 32.98, 33.86, 34.74, 35.73, 37.36,
          31.53, 33.28, 34.15, 35.03, 35.90, 40.28,
          31.53, 33.28, 34.15, 35.03, 36.20) # breaking stress MPAx1E6 of carbon block.
mps2par(H590, type="nor", ties="rounding", delta=0.005)$para
mps2par(H590, type="nor", ties="rounding" )$para
mps2par(H590, type="nor", ties="bernstein")$para
#        mu     sigma
# 34.071424  2.622484 # using a slight variant on their eq. 4.1.
# 34.071424  2.622614 # using log10offset=3
# 34.088769  2.690781 # using Bernstein smooth and unaffecting middle of odd tie runs
# The MoranTest show rejection of the Normal distribution at alpha=0.05, with the
# "rounding" and "delta=0.005"" and T=63.8 compared to their result of T=63.1,
# which to be considered that the strategy here is not precisely the same as theirs.
```

```
## End(Not run)

## Not run:
# Demonstration of parameter transformation and retransformation
set.seed(9209) # same seed used under mle2par() in parallel example
x <- rlmomco(500, vec2par(c(1,1,3), type="gam")) # 3-p Generalized Gamma
guess <- lmr2par(x, type="gam", p=3) # By providing a 3-p guess the 3-p
# Generalized Gamma will be triggered internally. There are problems passing
# "p" argument to optim() if that function is to pick up the ... argument.
mps2par(x, type="gam", init.para=guess, silent=FALSE,
           ptransf=  function(t) { c(log(t[1]), log(t[2]), t[3])},
           pretransf=function(t) { c(exp(t[1]), exp(t[2]), t[3])})$para
# Reports:       mu     sigma        nu   for some simulated data.
#         0.9997019 1.0135674 3.0259012
## End(Not run)
```

---

nonexceeds *Some Common or Useful Nonexceedance Probabilities*

Description

This function returns a vector nonexceedance probabilities.

Usage

```
nonexceeds(f01=FALSE, less=FALSE, sig6=FALSE)
```

Arguments

f01	A logical and if TRUE then 0 and 1 are included in the returned vector.
less	A logical and if TRUE the default values are trimmed back.
sig6	A logical that will instead sweep ± 6 standard deviations and transform standard normal variates to nonexceedance probabilities.

Value

A vector of selected nonexceedance probabilities F useful in assessing the "frequency curve" in applications (noninclusive). This vector is intended to be helpful and self-documenting when common F values are desired to explore deep into both distribution tails.

Author(s)

W.H. Asquith

See Also

check.fs, prob2T, T2prob

Examples

```
lmr <- lmoms(rnorm(20))
para <- parnor(lmr)
quanor(nonexceeds(), para)
```

---

par2cdf — *Cumulative Distribution Function of the Distributions*

Description

This function acts as a front end or dispatcher to the distribution-specific cumulative distribution functions.

Usage

```
par2cdf(x, para, ...)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `lmom2par` or `vec2par`.
`...`	The additional arguments are passed to the cumulative distribution function such as `paracheck=FALSE` for the Generalized Lambda distribution (`cdfgld`).

Value

Nonexceedance probability ($0 \le F \le 1$) for `x`.

Author(s)

W.H. Asquith

See Also

`par2pdf`, `par2qua`

Examples

```
lmr       <- lmoms(rnorm(20))
para      <- parnor(lmr)
nonexceed <- par2cdf(0,para)
```

par2cdf2 *Equivalent Cumulative Distribution Function of Two Distributions*

Description

This function computes the nonexceedance probability of a given quantile from a linear weighted combination of two quantile functions but accomplishes this from the perspective of cumulative distribitution functions (see par2qua2). For the current implementation simply uniroot'ing of a internally declared function and par2qua2 is made. Mathematical details are provided under par2qua2.

Usage

```
par2cdf2(x, para1, para2, weight=NULL, ...)
```

Arguments

x	A real value vector.
para1	The first distribution parameters from lmom2par or vec2par.
para2	The second distribution parameters from lmom2par or vec2par.
weight	An optional weighting argument to use in lieu of the F. Consult the documentation for par2qua2 for the implementation details when weight is NULL.
...	The additional arguments are passed to the quantile function.

Value

Nonexceedance probabilities ($0 \leq F \leq 1$) for x from the two distributions.

Author(s)

W.H. Asquith

See Also

par2cdf, lmom2par, par2qua2

Examples

```
lmr <- lmoms(rnorm(20)); left <- parnor(lmr); right <- pargev(lmr)
mixed.median    <- par2qua2(0.5,          left, right)
mixed.nonexceed <- par2cdf2(mixed.median, left, right)
```

par2lmom *Convert the Parameters of a Distribution to the L-moments*

Description

This function acts as a frontend or dispatcher to the distribution-specific L-moments of the parameter values. This function dispatches to `lmomCCC` where CCC represents the three character (minimum) distribution identifier: `aep4`, `cau`, `emu`, `exp`, `gam`, `gev`, `gld`, `glo`, `gno`, `gov`, `gpa`, `gum`, `kap`, `kmu`, `kur`, `lap`, `lmrq`, `ln3`, `nor`, `pe3`, `ray`, `revgum`, `rice`, `sla`, `st3`, `texp`, `wak`, and `wei`.

The conversion of parameters to TL-moments (`TLmoms`) is not supported. Specific use of functions such as `lmomTLgld` and `lmomTLgpa` for the TL-moments of the Generalized Lambda and Generalized Pareto distributions is required.

Usage

```
par2lmom(para, ...)
```

Arguments

`para`	A parameter object of a distribution.
`...`	Other arguments to pass.

Value

An L-moment object (an R `list`) is returned.

Author(s)

W.H. Asquith

See Also

`lmom2par`

Examples

```
lmr      <- lmoms(rnorm(20))
para     <- parnor(lmr)
frompara <- par2lmom(para)
```

par2pdf *Probability Density Function of the Distributions*

Description

This function acts as a frontend or dispatcher to the distribution-specific probability density functions.

Usage

```
par2pdf(x, para, ...)
```

Arguments

x	A real value vector.
para	The parameters from lmom2par or similar.
...	The additional arguments are passed to the quantile function such as paracheck = FALSE for the Generalized Lambda distribution (quagld).

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

See Also

par2cdf, par2qua

Examples

```
para    <- parnor(lmoms(rnorm(20)))
density <- par2pdf(par2qua(0.5, para), para)
```

par2qua *Quantile Function of the Distributions*

Description

This function acts as a frontend or dispatcher to the distribution-specific quantile functions.

Usage

```
par2qua(f,para,...)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from lmom2par or vec2par.
...	The additional arguments are passed to the quantile function such as paracheck = FALSE for the Generalized Lambda distribution (quagld).

Value

Quantile value for F.

Author(s)

W.H. Asquith

See Also

par2cdf, par2pdf

Examples

```
lmr     <- lmoms(rnorm(20))
para    <- parnor(lmr)
median  <- par2qua(0.5,para)
```

par2qua2 *Equivalent Quantile Function of Two Distributions*

Description

This function computes the nonexceedance probability of a given quantile from a linear weighted combination of two quantile functions—a mixed distribution:

$$Q_{\mathrm{mixed}}(F; \Theta_1, \Theta_2, \omega) = (1-\omega)Q_1(F, \Theta_1) + \omega Q_2(F, \Theta 2),$$

where Q is a quantile function for nonexceedance probability F, the distributions have parameters Θ_1 and Θ_2, and ω is a weight factor.

The distributions are specified by the two parameter object arguments in usual **lmomco** style. When proration by the nonexceedance probability is desired (`weight=NULL`, default), the left-tail parameter object (`para1`) is the distribution obviously governing the left tail; the right-tail parameter object (`para2`) is of course governs the right tail. The quantile function algebra is

$$Q(F) = (1-F^{\star}) \times \triangleleft Q(F) + F^{\star} \times Q(F)\triangleright,$$

where $Q(F)$ is the mixed quantile for nonexceedance probability F. $\triangleleft Q(F)$ is the first or left-tail quantile function; $Q(F)\triangleright$ is the second or right-tail quantile function. In otherwords, if `weight = NULL`, then $F^{\star} = F =$ `f` and the weight between the two quantile functions thus continuously varies from left to right. This is a probability proration from one to the other. A word of caution in this regard. The resulting weighted- or mixed-quantile function is not rigorously checked for monotonic increase with F, which is a required property of quantile functions. However, a first-order difference on the mixed quantiles with the probabilities is computed and a warning issued if not monotonic increasing.

If the optional `weight` argument is provided with length 1, then ω equals that weight. If `weight = 0`, then only the quantiles for $Q_1(F)$ are returned, and if `weight = 1`, then only the quantiles for the left tail $Q_2(F)$ are returned.

If the optional `weight` argument is provided with length 2, then $(1-\omega)$ is replaced by the first weight and ω is replaced by the second weight. These are internally rescaled to sum to unity before use and a warning is issued that this was done. Finally, the `par2cdf2` function inverses the above equation for F.

Usage

```
par2qua2(f, para1, para2, wfunc=NULL, weight=NULL, as.list=FALSE, ...)
```

Arguments

- `f` Nonexceedance probability ($0 \le F \le 1$).
- `para1` The first or left-tail parameters from `lmom2par` or `vec2par`.
- `para2` The second or right-tail parameters from `lmom2par` or similar.
- `wfunc` A function taking the argument `f` and computing a weight for the `para2` curve for which the complement of the computed weight is used for the weight on `para1`.

weight	An optional weighting argument to use in lieu of F. If NULL then prorated by the f, if weight has length 1, then weight on left distribution is the complement of the weight and weight on right distribution is weight[1], and if weight had length 2, then weight[1] is the weight on the left distribution and weight[2] is the weight on the right distribution.
as.list	A logical to control whether an R data.frame is returned having a column for f and for the mixed quantiles. This feature is provided for some design consistency with par2qua2lo, which mandates a data.frame return.
...	The additional arguments are passed to the quantile function.

Value

The weighted quantile value for F from the two distributions.

Author(s)

W.H. Asquith

See Also

par2qua, par2cdf2, par2qua2lo

Examples

```
lmr <- lmoms(rnorm(20)); left <- parnor(lmr); right <- pargev(lmr)
mixed.median <- par2qua2(0.5, left, right)

# Bigger example--using Kappa fit to whole sample for the right tail and
# Normal fit to whole sample for the left tail
D   <- c(123, 523, 345, 356, 2134, 345, 2365, 235, 12, 235, 61, 432, 843)
lmr <- lmoms(D); KAP <- parkap(lmr); NOR <- parnor(lmr); PP <- pp(D)
plot( PP, sort(D), ylim=c(-500, 2300))
lines(PP, par2qua( PP, KAP),      col=2)
lines(PP, par2qua( PP, NOR),      col=3)
lines(PP, par2qua2(PP, NOR, KAP), col=4)
```

---

par2qua2lo — *Equivalent Quantile Function of Two Distributions Stemming from Left-Hand Threshold to Setup Conditional Probability Computations*

Description

EXPERIMENTAL! This function computes the nonexceedance probability of a given quantile from a linear weighted combination of two quantile functions—a mixed distribution—when the data have been processed through the x2xlo function setting up left-hand thresholding and conditional probability compuation. The par2qua2lo function is a partial generalization of the par2qua2 function (see there for the basic mathematics). The **Examples** section has an exhaustive demonstration. The resulting weighted- or mixed-quantile function is not rigorously checked for monotonic increase with F, which is a required property of quantile functions. However, a first-order difference on the mixed quantiles with the probabilities is computed and a warning issued if not monotonic increasing.

Usage

```
par2qua2lo(f, para1, para2, xlo1, xlo2,
              wfunc=NULL, weight=NULL, addouts=FALSE,
              inf.as.na=TRUE, ...)
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$).
`para1`	The first distribution parameters from `lmom2par` or `vec2par`.
`para2`	The second distribution parameters from `x2xlo`.
`xlo1`	The first distribution parameters from `x2xlo`.
`xlo2`	The second distribution parameters from `lmom2par` or similar.
`wfunc`	A function taking the argument `f` and computing a weight for the `para2` curve for which the complement of the computed weight is used for the weight on `para1`.
`weight`	An optional weighting argument to use in lieu of `F`. If `NULL` then weights are a function of `length(xlo1$xin)` and `length(xlo2$xin)` for the first and second distribution respectively, if `weight` has length 1, then weight on first distribution is the complement of the weight, and the weight on second distribution is `weight[1]`, and if `weight` had length 2, then `weight[1]` is the weight on the first distribution, and `weight[2]` is the weight on the second distribution.
`addouts`	In the computation of weight factors when the `xlo1$xin` and `xlo2$xin` are used by other argument settings, the `addouts` arguments triggers the inclusion of the lengths of the `xlo1$xout` and `xlo2$xout` (see source code).
`inf.as.na`	A logical controlling whether quantiles for each distribution that are non-finite are to be converted to `NA`s. If they are converter to `NA`s, then when the application of the weight or weights are made then that those indices of `NA` quantiles become a zero and the weight for the other quantile will become unity. It is suggested to review the source code.
`...`	Additional arguments to pass if needed.

Value

The mixed quantile values for likely a subset of the provided `f` from the two distributions depending on the internals of `xlo1` and `xlo2` require the quantiles to actually start. This requires this function to return an R `data.frame` that was only optional for `par2qua2`:

`f`	Nonexceedance probabilities.
`quamix`	The mixed quantiles.
`delta_curve1`	The computation `quamix` minus curve for `para1`.
`delta_curve2`	The computation `quamix` minus curve for `para2`.

Alternatively, the returned value could be a weighting function for subsequent calls as `wfunc` to `par2qua2lo` (see **Examples**). This alternative operation is triggered by setting `wfunc` to an arbitrary character string, and internally the contents of `xlo1` and `xlo2`, which themselves have to be called as named arguments, are recombined. This means that the `xin` and `xout` are recombined, into their respective samples. Each data point is then categorized with probability zero for the `xlo1` values and probability unity for the `xlo2` values. A logistic regression is fit using logit-link function for a binomial family using a generalized linear model. The binomial (0 or 1) is regressed as a function of the plotting positions of a sample composed of `xlo1` and `xlo2`. The coefficients of the regression are extracted, and a function created to predict the probability of event "`xlo2`". The `attributes` of the computed value inside the function store the coefficients, the regression model, and potentially useful for graphical review, a `data.frame` of the data used for the regression. This sounds more complicated than it really is (see source code and **Examples**).

Author(s)

W.H. Asquith

See Also

`par2qua`, `par2cdf2`, `par2qua2`, `x2xlo`

Examples

```
## Not run:
XloSNOW <- list( # data from "snow events" from prior call to x2xlo()
   xin=c(4670, 3210, 4400, 4380, 4350, 3380, 2950, 2880, 4100),
   ppin=c(0.9444444, 0.6111111, 0.8888889, 0.8333333, 0.7777778, 0.6666667,
          0.5555556, 0.5000000, 0.7222222),
   xout=c(1750, 1610, 1750, 1460, 1950, 1000, 1110, 2600),
   ppout=c(0.27777778, 0.22222222, 0.33333333, 0.16666667, 0.38888889,
           0.05555556, 0.11111111, 0.44444444),
   pp=0.4444444, thres=2600, nin=9, nout=8, n=17, source="x2xlo")
# RAIN data from prior call to x2xlo() are
XloRAIN <- list( # data from "rain events" from prior call to x2xlo()
   xin=c(5240, 6800, 5990, 4600, 5200, 6000, 4500, 4450, 4480, 4600,
         3290, 6700, 10600, 7230, 9200, 6540, 13500, 4250, 5070,
         6640, 6510, 3610, 6370, 5530, 4600, 6570, 6030, 7890, 8410),
   ppin=c(0.41935484, 0.77419355, 0.48387097, 0.25806452, 0.38709677, 0.51612903,
          0.22580645, 0.16129032, 0.19354839, 0.29032258, 0.06451613, 0.74193548,
          0.93548387, 0.80645161, 0.90322581, 0.64516129, 0.96774194, 0.12903226,
          0.35483871, 0.70967742, 0.61290323, 0.09677419, 0.58064516, 0.45161290,
          0.32258065, 0.67741935, 0.54838710, 0.83870968, 0.87096774),
   xout=c(1600), ppout=c(0.03225806),
   pp=0.03225806, thres=2599, nin=29, nout=1, n=30, source="x2xlo")

QSNOW <- c(XloSNOW$xin,  XloSNOW$xout ) # collect all of the snow
QRAIN <- c(XloRAIN$xin,  XloRAIN$xout ) # collect all of the rain
PSNOW <- c(XloSNOW$ppin, XloSNOW$ppout) # probabilities collected
PRAIN <- c(XloRAIN$ppin, XloRAIN$ppout) # probabilities collected

# Logistic regression to blend the proportion of snow versus rain events as
# ***also*** a function of nonexceedance probability
```

```
wfunc <- par2qua2lo(xlo1=XloSNOW, xlo2=XloRAIN, wfunc="wfunc") # weight function

# Plotting the data and the logistic regression. This shows how to gain access
# to the attributes, in order to get the data, so that we can visualize the
# probability mixing between the two samples. If the two samples are not a
# function of probability, then each systematically would have a regression-
# predicted weight of 50/50. For the RAIN and SNOW, the SNOW is likely to
# produce the smaller events and RAIN the larger.
 opts <- par(las=1) # Note the 0.5 in the next line is arbitrary, we simply
 bin <- attr(wfunc(0.5), "data") # have to use wfunc() to get its attributes.
 FF <- seq(0,1,by=0.01); HH <- wfunc(FF); n <- length(FF)
 plot(bin$f, bin$prob, tcl=0.5, col=2*bin$prob+2,
      xlab="NONEXCEEDANCE PROBABILITY", ylab="RAIN-CAUSED EVENT RELATIVE TO SNOW")
 lines(c(-0.04,1.04), rep(0.5,2), col=8, lwd=0.8) # origin line at 50/50 chance
 text(0, 0.5, "50/50 chance line", pos=4, cex=0.8)
 segments(FF[1:(n-1)], HH[1:(n-1)], x1=FF[2:n], y1=HH[2:n], lwd=1+4*abs(FF-0.5),
          col=rgb(1-FF,0,FF)) # line grades from one color to other
 text(1, 0.1, "Events caused by snow", col=2, cex=0.8, pos=2)
 text(0, 0.9, "Events caused by rain", col=4, cex=0.8, pos=4)
 par(opts)

# Suppose that the Pearson type III is thought applicable to the SNOW
# and the AEP4 for the RAIN, now estimate respective parameters.
parSNOW <- lmr2par(log10(XloSNOW$xin), type="nor" )
parRAIN <- lmr2par(log10(XloRAIN$xin), type="wak")
# Two distributions are chosen to show the user than we are not constrained to one.

Qall   <- c(QSNOW, QRAIN)                # combine into a "whole" sample
XloALL <- x2xlo(Qall, leftout=2600, a=0) # apply the low-outlier threshold
parALL <- lmr2par(log10(XloALL$xin), type="nor") # estimate Wakeby
# Wakey has five parameters and is very flexible.

FF <- nonexceeds() # useful nonexceedance probabilities
col <- c(rep(0,length(QSNOW)), rep(2,length(QRAIN))) # for coloring
plot(0, 0, col=2+col, ylim=c(1000,20000), xlim=qnorm(range(FF)), log="y",
           xlab="STANDARD NORMAL VARIATE", ylab="QUANTILE", type="n")
lines(par()$usr[1:2], rep(2600, 2), col=6, lty=2, lwd=0.5) # draw threshold
points(qnorm(pp(Qall, sort=FALSE)), Qall, col=2+col, lwd=0.98) # all record
points(qnorm(PSNOW), QSNOW, pch=16, col=2) # snow events
points(qnorm(PRAIN), QRAIN, pch=16, col=4) # rain events
lines(     qnorm(f2f(  FF, xlo=XloSNOW)), # show fitted curve for snow events
      10^par2qua(f2flo(FF, xlo=XloSNOW ), parSNOW), col=2)
lines(     qnorm(f2f(  FF, xlo=XloRAIN)), # show fitted curve for rain events
      10^par2qua(f2flo(FF, xlo=XloRAIN ), parRAIN), col=4)
lines(     qnorm(f2f(  FF, xlo=XloALL )), # show fitted curve for all events combined
      10^par2qua(f2flo(FF, xlo=XloALL  ), parALL ), col=1, lty=3)
PQ <- par2qua2lo(      FF, parSNOW, parRAIN, XloSNOW, XloRAIN, wfunc=wfunc)
lines(qnorm(PQ$f), 10^PQ$quamix, lwd=2)                # draw the mixture
legend(-3,20000, c("Rain curve", "Snow curve", "All combined (all open circles)",
                   "MIXED CURVE by par2qua2lo()"),
                 bty="n", lwd=c(1,1,1,2), lty=c(1,1,3,1), col=c(4,2,1,1))
text(-3, 15000, "A low-outlier threshold of 2,600 is used throughout.", col=6, pos=4)
text(-3,  2600, "2,600", cex=0.8, col=6, pos=4)
```

```
mtext("Mixed population frequency computation of snow and rainfall streamflow")#
## End(Not run)

## Not run:
nsim <- 50000; FF <- runif(nsim); WF <- wfunc(FF)
rB <- rbinom(nsim, 1, WF)
RF <- FF[rB == 1]; SF <- FF[rB == 0]
RAIN <- 10^qlmomco(f2flo(runif(length(RF)), xlo=XloRAIN), parRAIN)
SNOW <- 10^qlmomco(f2flo(runif(length(SF)), xlo=XloRAIN), parSNOW)
RAIN[RAIN < XloRAIN$thres] <- XloRAIN$thres
SNOW[SNOW < XloSNOW$thres] <- XloSNOW$thres
RAIN <- c(RAIN,rep(XloRAIN$thres, length(RF)-length(RAIN)))
SNOW <- c(SNOW,rep(XloSNOW$thres, length(SF)-length(SNOW)))
ALL <- c(RAIN,SNOW)
lines(qnorm(pp(ALL)), sort(ALL), cex=0.6, lwd=0.8, col=3)

RF <- FF[rB == 1]; SF <- FF[rB == 0]
RAIN <- 10^qlmomco(RF, parRAIN)
SNOW <- 10^qlmomco(SF, parSNOW)
RAIN[RAIN < XloRAIN$thres] <- XloRAIN$thres
SNOW[SNOW < XloSNOW$thres] <- XloSNOW$thres
RAIN <- c(RAIN,rep(XloRAIN$thres, length(RF)-length(RAIN)))
SNOW <- c(SNOW,rep(XloSNOW$thres, length(SF)-length(SNOW)))
ALL <- c(RAIN,SNOW)
lines(qnorm(pp(ALL)), sort(ALL), cex=0.6, lwd=0.8, col=3)

RF <- FF[rB == 1]; SF <- FF[rB == 0]
RAIN <- 10^qlmomco(f2flo(RF, xlo=XloRAIN), parRAIN)
SNOW <- 10^qlmomco(f2flo(SF, xlo=XloRAIN), parSNOW)
RAIN[RAIN < XloRAIN$thres] <- XloRAIN$thres
SNOW[SNOW < XloSNOW$thres] <- XloSNOW$thres
RAIN <- c(RAIN,rep(XloRAIN$thres, length(RF)-length(RAIN)))
SNOW <- c(SNOW,rep(XloSNOW$thres, length(SF)-length(SNOW)))
ALL <- c(RAIN,SNOW)
lines(qnorm(pp(ALL)), sort(ALL), cex=0.6, lwd=0.8, col=3) #
## End(Not run)
```

`par2vec` *Convert a Parameter Object to a Vector of Parameters*

Description

This function converts a parameter object to a vector of parameters using the `$para` component of the parameter list such as returned by `vec2par`.

Usage

```
par2vec(para, ...)
```

Arguments

`para`	A parameter object of a distribution.
`...`	Additional arguments should they even be needed.

Value

An R `vector` is returned in moment order.

Author(s)

W.H. Asquith

See Also

`vec2par`

Examples

```
para <- vec2par(c(12,123,0.5), type="gev")
par2vec(para)
#   xi alpha kappa
# 12.0 123.0   0.5
```

`paraep4` *Estimate the Parameters of the 4-Parameter Asymmetric Exponential Power Distribution*

Description

This function estimates the parameters of the 4-parameter Asymmetric Exponential Power distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relation between distribution parameters and L-moments is seen under `lmomaep4`. Relatively straightforward, but difficult to numerically achieve, optimization is needed to extract the parameters from the L-moments. If the τ_3 of the distribution is zero (symmetrical), then the distribution is known as the Exponential Power (see `lmrdia46`).

Delicado and Goria (2008) argue for numerical methods to use the following objective function

$$\epsilon(\alpha, \kappa, h) = \log(1 + \sum_{r=2}^{4}(\hat{\lambda}_r - \lambda_r)^2),$$

and subsequently solve directly for ξ. This objective function was chosen by Delicado and Goria because the solution surface can become quite flat for away from the minimum. The author of **lmomco** agrees with the findings of those authors from limited exploratory analysis and the development of the algorithms used here under the rubic of the “DG” method. This exploration resulted in an alternative algorithm using tabulated initial guesses described below. An evident drawback of the Delicado-Goria algorithm, is that precision in α is may be lost according to the observation that this parameter can be analytically computed given λ_2, κ, and h.

It is established practice in L-moment theory of four (and similarly three) parameter distributions to see expressions for τ_3 and τ_4 used for numerical optimization to obtain the two higher parameters (α and h) first and then see analytical expressions directly compute the two lower parameters (ξ and α). The author made various exploratory studies by optimizing on τ_3 and τ_4 through a least squares objective function. Such a practice seems to perform acceptably when compared to that recommended by Delicado and Goria (2008) when the initial guesses for the parameters are drawn from pretabulation of the relation between $\{\alpha, h\}$ and $\{\tau_3, \tau_4\}$.

Another optimization, referred to here as the "A" (Asquith) method, is available for parameter estimation using the following objective function

$$\epsilon(\kappa, h) = \sqrt{(\hat{\tau}_3 - \tau_3)^2 + (\hat{\tau}_4 - \tau_4)^2},$$

and subsequently solve directly for α and then ξ. The "A" method appears to perform better in κ and h estimation and quite a bit better in α and and ξ as seemingly expected because these last two are analytically computed (Asquith, 2014). The objective function of the "A" method defaults to use of the $\sqrt{x}$ but this can be removed by setting `sqrt.t3t4=FALSE`.

The initial guesses for the κ and h parameters derives from a hashed environment in in file 'sysdata.rda' (`.lmomcohash$AEPkh2lmrTable`) in which the $\{\kappa, h\}$ pair having the minimum $\epsilon(\kappa, h)$ in which τ_3 and τ_4 derive from the table as well. The file 'SysDataBuilder01.R' provides additional technical details on how the `AEPkh2lmrTable` was generated. The table represents a systematic double-loop sweep through `lmomaep4` for

$$\kappa \mapsto \{-3 \le \log(\kappa) \le 3, \Delta\log(\kappa) = 0.05\},$$

and

$$h \mapsto \{-3 \le \log(h) \le 3, \Delta\log(h) = 0.05\}.$$

The function will not return parameters if the following lower (estimated) bounds of τ_4 are not met: $\tau_4 \ge 0.77555(|\tau_3|) - 3.3355(|\tau_3|)^2 + 14.196(|\tau_3|)^3 - 29.909(|\tau_3|)^4 + 37.214(|\tau_3|)^5 - 24.741(|\tau_3|)^6 + 6.7998(|\tau_3|)^7$. For this polynomial, the residual standard error is RSE = 0.0003125 and the maximum absolute error for $\tau_3{:}[0,1] < 0.0015$. The actual coefficients in `paraep4` have additional significant figures. However, the argument `snap.tau4`, if set, will set τ_4 equal to the prediction from the polynomial. This value of τ_4 should be close enough numerically to the boundary because the optimization is made using a log-transformation to ensure that α, κ, and h remain in the positive domain—though the argument `nudge.tau4` is provided to offset τ_4 upward just incase of optimization problems.

Usage

```
paraep4(lmom, checklmom=TRUE, method=c("A", "DG", "ADG"),
        sqrt.t3t4=TRUE, eps=1e-4, checkbounds=TRUE, kapapproved=TRUE,
        snap.tau4=FALSE, nudge.tau4=0,
        A.guess=NULL, K.guess=NULL, H.guess=NULL, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the L-moments be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the

L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.

method
: Which method for parameter estimation should be used. The "A" or "DG" methods. The "ADG" method will run both methods and retains the salient optimization results of each but the official parameters in para are those from the "A" method. Lastly, all minimization is based on the optim function using the Nelder–Mead method and default arguments.

sqrt.t3t4
: If true and the method is "A", then the square root of the sum of square errors in τ_3 and τ_4 are used instead of sum of square differences alone.

eps
: A small term or threshold for which the square root of the sum of square errors in τ_3 and τ_4 is compared to to judge "good enough" for the alogrithm to set the ifail on return in addition to convergence flags coming from the optim function. Note that eps is only used if the "A" or "ADG" methods are triggered because the other method uses the scale parameter which in reality could be quite large relative to the other two shape parameters, and a reasonable default for such a secondary error threshold check would be ambiguous.

checkbounds
: Should the lower bounds of τ_4 be verified and if sample $\hat{\tau}_3$ and $\hat{\tau}_4$ are outside of these bounds, then NA are returned for the solutions.

kapapproved
: Should the Kappa distribution be fit by parkap if $\hat{\tau}_4$ is below the lower bounds of τ_4? This fitting is only possible if checkbounds is true. The Kappa and AEP4 overlap partially. The AEP4 extends τ_4 above Generalized Logistic and Kappa extends τ_4 below the lower bounds of τ_4 for AEP4 and extends all the way to the theoretical limits as used within are.lmom.valid.

snap.tau4
: A logical to "snap" the τ_4 upwards to the lower boundary if the given τ_4 is lower than the boundary described in the polynomial.

nudge.tau4
: An offset to the snapping of τ_4 intended to move τ_4 just above the lower bounds in case of optimization problems. (The absolute value of the nudge is made internally to ensure only upward adjustment by an addition operation.)

A.guess
: A user specified guess of the α parameter to provide to the optimization of any of the methods. This argument just superceeds the simple initial guess of $\alpha = 1$.

K.guess
: A user specified guess of the κ parameter to supercede that derived from the .lmomcohash$AEPkh2lmrTable in file 'sysdata.rda'.

H.guess
: A user specified guess of the h parameter to supercede that derived from the .lmomcohash$AEPkh2lmrTable in file 'sysdata.rda'.

...
: Other arguments to pass.

Value

An R list is returned.

type
: The type of distribution: aep4.

para
: The parameters of the distribution.

source
: The source of the parameters: "paraep4".

method
: The method as specified by the method.

`ifail` A numeric failure code.

`ifailtext` A text message for the failure code.

`L234` Optional and dependent on method “DG” or “ADG”. Another R `list` containing the optimization details by the “DG” method along with the estimated parameters in `para_L234`. The “_234” is to signify that optimization is made using λ_2, λ_3, and λ_4. The parameter values in `para` are those only when the “DG” method is used.

`T34` Optional and dependent on method “A” or “ADG”. Another R `list` containing the optimization details by the “A” method along with the estimated parameters in `para_T34`. The “_T34” is to signify that opimization is being conducted using τ_3 and τ_4 only. The parameter values in `para` are those by the “A” method.

The values for `ifail` or produced by three mechanisms. First, the convergence number emanating from the `optim` function itself. Second, the integer 1 is used when the failure is attributable to the `optim` function. Third, the interger 2 is a general attempt to have a singular failure by sometype of `eps` outside of `optim`. Fourth, the integer 3 is used to show that the parameters fail against a parameter validity check in `are.paraep4.valid`. And fifth, the integer 4 is used to show that the sample L-moments are below the lower bounds of the τ_4 polynomial shown here.

Additional and self explanatory elements on the returned list will be present if the Kappa distribution was fit instead.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Delicado, P., and Goria, M.N., 2008, A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution: Computational Statistics and Data Analysis, v. 52, no. 3, pp. 1661–1673.

See Also

`lmomaep4`, `cdfaep4`, `pdfaep4`, `quaaep4`, `quaaep4kapmix`

Examples

```
# As a general rule AEP4 optimization can be CPU intensive

## Not run:
lmr <- vec2lmom(c(305, 263, 0.815, 0.631))
plotlmrdia(lmrdia()); points(lmr$ratios[3], lmr$ratios[4], pch=16, cex=3)
PAR <- paraep4(lmr, snap.tau4=TRUE) # will just miss the default eps
FF <- nonexceeds(sig6=TRUE)
plot(FF, quaaep4(FF, PAR), type="l", log="y")
lmomaep4(PAR) # 305, 263, 0.8150952, 0.6602706 (compare to those in lmr)
```

```
## End(Not run)

## Not run:
PAR <- list(para=c(100, 1000, 1.7, 1.4), type="aep4")
lmr <- lmomaep4(PAR)
aep4 <- paraep4(lmr, method="ADG")
print(aep4) #
## End(Not run)

## Not run:
PARdg  <- paraep4(lmr, method="DG")
PARasq <- paraep4(lmr, method="A")
print(PARdg)
print(PARasq)
F <- c(0.001, 0.005, seq(0.01,0.99, by=0.01), 0.995, 0.999)
qF <- qnorm(F)
ylim <- range( quaaep4(F, PAR), quaaep4(F, PARdg), quaaep4(F, PARasq) )
plot(qF, quaaep4(F, PARdg), type="n", ylim=ylim,
     xlab="STANDARD NORMAL VARIATE", ylab="QUANTILE")
lines(qF, quaaep4(F, PAR), col=8, lwd=10) # the true curve
lines(qF, quaaep4(F, PARdg),  col=2, lwd=3)
lines(qF, quaaep4(F, PARasq), col=3, lwd=2, lty=2)
# See how the red curve deviates, Delicado and Goria failed
# and the ifail attribute in PARdg is TRUE. Note for lmomco 2.3.1+
# that after movement to log-exp transform to the parameters during
# optimization that this "error" as described does not appear to occur.

print(PAR$para)
print(PARdg$para)
print(PARasq$para)

ePAR1dg <- abs((PAR$para[1] - PARdg$para[1])/PAR$para[1])
ePAR2dg <- abs((PAR$para[2] - PARdg$para[2])/PAR$para[2])
ePAR3dg <- abs((PAR$para[3] - PARdg$para[3])/PAR$para[3])
ePAR4dg <- abs((PAR$para[4] - PARdg$para[4])/PAR$para[4])

ePAR1asq <- abs((PAR$para[1] - PARasq$para[1])/PAR$para[1])
ePAR2asq <- abs((PAR$para[2] - PARasq$para[2])/PAR$para[2])
ePAR3asq <- abs((PAR$para[3] - PARasq$para[3])/PAR$para[3])
ePAR4asq <- abs((PAR$para[4] - PARasq$para[4])/PAR$para[4])

MADdg  <- mean(ePAR1dg,  ePAR2dg,  ePAR3dg,  ePAR4dg)
MADasq <- mean(ePAR1asq, ePAR2asq, ePAR3asq, ePAR4asq)

# We see that the Asquith method performs better for the example
# parameters in PAR and inspection of the graphic will show that
# the Delicado and Goria solution is obviously off. (See Note above)
print(MADdg)
print(MADasq)

# Repeat the above with this change in parameter to
# PAR <- list(para=c(100, 1000, .7, 1.4), type="aep4")
# and the user will see that all three methods converged on the
```

```
# correct values.
## End(Not run)
```

---

parcau *Estimate the Parameters of the Cauchy Distribution*

Description

This function estimates the parameters of the Cauchy distribution from the trimmed L-moments (TL-moments) having trim level 1. The relations between distribution parameters and the TL-moments (trim=1) are seen under lmomcau.

Usage

```
parcau(lmom, ...)
```

Arguments

lmom	A TL-moment object from TLmoms with trim=1.
...	Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: cau.
para	The parameters of the distribution.
source	The source of the parameters: "parcau".

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

TLmoms, lmomcau, cdfcau, pdfcau, quacau

Examples

```
X1 <- rcauchy(20)
parcau(TLmoms(X1,trim=1))
```

paremu *Estimate the Parameters of the Eta-Mu Distribution*

Description

This function estimates the parameters (η and α) of the Eta-Mu ($\eta : \mu$) distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between distribution parameters and L-moments are seen under lmomemu.

The basic approach for parameter optimization is to extract initial guesses for the parameters from the table EMU_lmompara_byeta in the .lmomcohash environment. The parameters having a minimum Euclidean error as controlled by three arguments are used for initial guesses in a Nelder-Mead simplex multidimensional optimization using the R function optim and default arguments.

Limited testing indicates that of the "error term controlling options" that the default values as shown in the Usage section seem to provide superior performance in terms of recovering the *a priori known* parameters in experiments. It seems that only Euclidean optimization using L-skew and L-kurtosis is preferable, but experiments show the general algorithm to be slow.

Usage

```
paremu(lmom, checklmom=TRUE, checkbounds=TRUE,
         alsofitT3=FALSE, alsofitT3T4=FALSE, alsofitT3T4T5=FALSE,
         justfitT3T4=TRUE, boundary.tolerance=0.001,
         verbose=FALSE, trackoptim=TRUE, ...)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality).
checkbounds	Should the L-skew and L-kurtosis boundaries of the distribution be checked.
alsofitT3	Logical when true will add the error term $(\hat{\tau}_3 - \tau_3)^2$ to the sum of square errors for the mean and L-CV.
alsofitT3T4	Logical when true will add the error term $(\hat{\tau}_3 - \tau_3)^2 + (\hat{\tau}_4 - \tau_4)^2$ to the sum of square errors for the mean and L-CV.
alsofitT3T4T5	Logical when true will add the error term $(\hat{\tau}_3 - \tau_3)^2 + (\hat{\tau}_4 - \tau_4)^2 + (\hat{\tau}_5 - \tau_5)^2$ to the sum of square errors for the mean and L-CV.
justfitT3T4	Logical when true will only consider the sum of squares errors for L-skew and L-kurtosis as mathematically shown for alsofitT3T4.
boundary.tolerance	A fudge number to help guide how close to the boundaries an arbitrary list of τ_3 and τ_4 can be to consider them formally in or out of the attainable $\{\tau_3, \tau_4\}$ domain.
verbose	A logical to control a level of diagnostic output.

trackoptim — A logical to control specific messaging through each iteration of the objective function.

... — Other arguments to pass.

Value

An R `list` is returned.

type — The type of distribution: `emu`.

para — The parameters of the distribution.

source — The source of the parameters: "paremu".

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

`lmomemu`, `cdfemu`, `pdfemu`, `quaemu`

Examples

```
## Not run:
   par1 <- vec2par(c(.3, 2.15), type="emu")
   lmr1 <- lmomemu(par1, nmom=4)
   par2.1 <- paremu(lmr1, alsofitT3=FALSE, verbose=TRUE, trackoptim=TRUE)
   par2.1$para # correct parameters not found: eta=0.889 mu=3.54
   par2.2 <- paremu(lmr1, alsofitT3=TRUE, verbose=TRUE, trackoptim=TRUE)
   par2.2$para # correct parameters not found: eta=0.9063 mu=3.607
   par2.3 <- paremu(lmr1, alsofitT3T4=TRUE,  verbose=TRUE, trackoptim=TRUE)
   par2.3$para # correct parameters not found: eta=0.910 mu=3.62
   par2.4 <- paremu(lmr1, justfitT3T4=TRUE,  verbose=TRUE, trackoptim=TRUE)
   par2.4$para # correct parameters not found: eta=0.559 mu=3.69

   x <- seq(0,3,by=.01)
   plot(x,  pdfemu(x, par1), type="l", lwd=6, col=8, ylim=c(0,2))
   lines(x, pdfemu(x, par2.1), col=2, lwd=2, lty=2)
   lines(x, pdfemu(x, par2.2), col=4)
   lines(x, pdfemu(x, par2.3), col=3, lty=3, lwd=2)
   lines(x, pdfemu(x, par2.4), col=5, lty=2, lwd=2)

## End(Not run)
```

parexp — *Estimate the Parameters of the Exponential Distribution*

Description

This function estimates the parameters of the Exponential distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomexp`.

Usage

```
parexp(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `exp`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “parexp”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmomexp`, `cdfexp`, `pdfexp`, `quaexp`

Examples

```
lmr <- lmoms(rnorm(20))
parexp(lmr)
```

---

pargam *Estimate the Parameters of the Gamma Distribution*

Description

This function estimates the parameters of the Gamma distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. Both the two-parameter Gamma and three-parameter Generalized Gamma distributions are supported based on the desired choice of the user, and numerical-hybrid methods are required. The pdfgam documentation provides further details.

Usage

```
pargam(lmom, p=c("2", "3"), checklmom=TRUE, ...)
```

Arguments

lmom	A L-moment object created by lmoms or vec2lmom.
p	The number of parameters to estimate for the 2-p Gamma or 3-p Generalized Gamma.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
...	Other arguments to pass.

Value

An R `list` is returned.

type	The type of distribution: `gam`.
para	The parameters of the distribution.
source	The source of the parameters: "pargam".

Note

The two-parameter Gamma is supported by Hosking's code-based approximations to avoid direct numerical techniques. The three-parameter version is based on a dual approach to parameter optimization. The $\log(\sigma)$ and $\sqrt{\log(\lambda_1/\lambda_2)}$ conveniently has a relatively narrow range of variation. A polynomial approximation to provide a first estimate of σ (named σ') is used through the `optim()`

function to isolated the best estimates of μ' and ν' of the distribution holding σ constant at $\sigma = \sigma'$—a 2D approach is thus involved. Then, the initial parameter for a second three-dimensional optimization is made using the initial parameter estimates as the tuple μ', σ', ν'. This 2D approach seems more robust and effectively canvases more of the Generalized Gamma parameter domain, though a doubled-optimization is not quite as fast as a direct 3D optimization. The following code was used to derive the polynomial coefficients used for the first approximation of $sigma'$:

```
nsim <- 10000; mu <- sig <- nu <- l1 <- l2 <- t3 <- t4 <- rep(NA, nsim)
for(i in 1:nsim) {
  m <- exp(runif(1, min=-4, max=4)); s <- exp(runif(1, min=-8, max=8))
  n <- runif(1, min=-14, max=14); mu[i] <- m; sig[i] <- s; nu[i] <- n
  para <- vec2par(c(m,s,n), type="gam"); lmr <- lmomgam(para)
  if(is.null(lmr)) next
  lam <- lmr$lambdas[1:2]; rat <- lmr$ratios[3:4]
  l1[i]<-lam[1]; l2[i]<-lam[2];t3[i]<-rat[1]; t4[i]<-rat[2]
}
ZZ <- data.frame(mu=mu, sig=sig, nu=nu, l1=l1, l2=l2, t3=t3, t4=t4)
ZZ$ETA <- sqrt(log(ZZ$l1/ZZ$l2)); ZZ <- ZZ[complete.cases(ZZ), ]
ix <- 1:length(ZZ$ETA);  ix <- ix[(ZZ$ETA < 0.025 & log(ZZ$sig) < 1)]
ZZ <- ZZ[-ix,]
with(ZZ, plot(ETA, log(sig), xlim=c(0,4), ylim=c(-8,8)))
LM <- lm(log(sig)~
         I(1/ETA^1)+I(1/ETA^2)+I(1/ETA^3)+I(1/ETA^4)+I(1/ETA^5)+
             ETA   +I(  ETA^2)+I(  ETA^3)+I(  ETA^4)+I(  ETA^5), data=ZZ)
ETA <- seq(0,4,by=0.002) # so the line of fit can be seen
lines(ETA, predict(LM, newdata=list(ETA=ETA)), col=2)
The.Coefficients.In.pargam.Function <- LM$coefficients
```

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmomgam, cdfgam, pdfgam, quagam

Examples

```
pargam(lmoms(abs(rnorm(20, mean=10))))
```

```
## Not run:
pargam(lmomgam(vec2par(c(0.3,0.4,+1.2), type="gam")), p=3)$para
pargam(lmomgam(vec2par(c(0.3,0.4,-1.2), type="gam")), p=3)$para
#        mu      sigma         nu
# 0.2999994  0.3999990  1.1999696
# 0.2999994  0.4000020 -1.2000567
## End(Not run)
```

---

pargdd *Estimate the Parameters of the Gamma Difference Distribution*

Description

This function estimates the parameters of the Gamma Difference distribution given the L-moments of the data in an ordinary L-moment object (lmoms). The relations between distribution parameters and L-moments are complex (see lmomgdd). The distribution has four parameters. The vector para in the parameter object with a fifth parameter uses that as a trigger between a symmetrical distribution with para[3:4] equals para[1:2] if para[5] = 1. If para[5] is not present, then the distribution can be asymmetrical, or if para[5] is present and set to any value that is not 1, then the distribution can be asymmetrical.

Usage

```
pargdd(lmom, checklmom=TRUE, symgdd=FALSE, init.para=NULL, snap.tau4=FALSE,
          silent=FALSE, trace=FALSE, control=list(abstol=0.0001, maxit=1000), ...)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function.
symgdd	A logical to trigger a symmetrical distribution by $\alpha_2 = \alpha_1$ and $\beta_1 = \beta_1$ and the fifth element of para on the return will be set to 1.
init.para	Optional initial values for the parameters used for starting values for the optim function. If this argument is not set, then an unrigorous attempt is made to guess at the initial parameters using some poor admittedly heuristics. The fifth element, if present, and set to 1, then the symdd is internally set to true.
snap.tau4	A logical to trigger snapping τ_4 to a nudge above the $\{\tau_3, \tau_4\}$ trajectory of the Pearson Type III distribution. The Gamma Difference only has solution in $\{\tau_3, \tau_4\}$ domain above the Pearson.
silent	The argument silent for try().
trace	A logical to trigger a message in the main objective function.
control	The argument control for optim().
...	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: gdd.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “pargdd”.
`optim`	The results of the parameter optimization call.

Author(s)

W.H. Asquith

See Also

`lmomgdd`, `cdfgdd`, `pdfgdd`, `quagdd`

Examples

```
## Not run:
# Example of the symmetrical case, see lmomgdd-Note section.
x <- seq(-20, 20, by=0.1); para <- list(para=c(3, 0.4, NA, NA, 1), type="gdd")
slmr  <- lmomgdd(  para);  nara <- pargdd(slmr, symgdd=TRUE)
given <- pdfgdd(x, para);  fit  <- pdfgdd(x, nara)
plot( x, given, type="l", col=8, lwd=4, ylim=range(c(given, fit)))
lines(x, fit,   col="red") #
## End(Not run)

## Not run:
# Example of the asymmetrical case, and as of Summer 2024 experiments, it seems
# the author does not quite have limits of GDD implementation known. Though this
# example works, we do not always L-moment recreation from fitted parameters.
x <- seq(-5, 15, by=0.1); para <- list(para=c(3, 1, 1, 3), type="gdd")
slmr  <- lmomgdd(  para);  nara <- pargdd(slmr)
given <- pdfgdd(x, para);  fit  <- pdfgdd(x, nara)
plot( x, given, type="l", col=8, lwd=4, ylim=range(c(a, fit)))
lines(x, fit,   col="red") #
## End(Not run)
```

`pargep` *Estimate the Parameters of the Generalized Exponential Poisson Distribution*

Description

This function estimates the parameters of the Generalized Exponential Poisson distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between distribution parameters and L-moments are seen under lmomgep. However, the expectations of order statistic extrema are computed through numerical integration of the quantile function and the fundamental definition of L-moments (theoLmoms.max.ostat). The mean must be $\lambda_1 > 0$. The implementation here fits the first three L-moments. A distribution having two scale parameters produces more than one solution. The higher L-moments are not consulted as yet in an effort to further enhance functionality. This function has deterministic starting points but on subsequent iterations the starting points do change. If a solution is not forthcoming, try running the whole function again.

Usage

```
pargep(lmom, checklmom=TRUE, checkdomain=TRUE, maxit=10, verbose=FALSE, ...)
```

Arguments

`lmom`	An L-moment object created by lmoms or vec2lmom.
`checklmom`	Should the `lmom` be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`checkdomain`	A logical controlling whether the empirically derived (approximated) boundaries of the GEP in the τ_2 and τ_3 domain are used for early exiting if the `lmom` do not appear compatible with the distribution.
`maxit`	The maximum number of iterations. The default should be about twice as big as necessary.
`verbose`	A logical controlling intermediate results, which is useful given the experimental nature of GEP parameter estimation and if the user is evaluating results at each iteration. The verbosity is subject to change.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `gep`.
`para`	The parameters of the distribution.
`convergence`	A numeric code on covergence, a value of 0 means solution looks ok.
`error`	Sum of relative error: $\epsilon = |(\lambda'_2 - \hat{\lambda}'_2)/\hat{\lambda}'_2| + |(\lambda_3 - \hat{\lambda}_3)/\hat{\lambda}_3|$ for the fitted (prime) and sample (hat, given in `lmom`) 2nd and 3rd L-moments. A value of 10 means that the τ_2 and τ_3 values are outside the domain of the distribution as determined by brute force computations and custom polynomial fits.
`its`	Iteration count.
`source`	The source of the parameters: "pargep".

Note

There are various inequalities and polynomials demarcating the τ_2 and τ_3 of the distribution. These were developed during a protracted period of investigation into the numerical limits of the distribution with a specific implementation in **lmomco**. Some of these bounds may or may not be optimal as empirically-arrived estimates of theoretical bounds. The polynomials where carefully assembled however. The straight inequalities are a bit more ad hoc following supervision of domain exploration. More research is needed but the domain constraint provided should generally produce parameter solutions.

Author(s)

W.H. Asquith

See Also

`lmomgep`, `cdfgep`, `pdfgep`, `quagep`

Examples

```
## Not run:
# Two examples well inside the domain but known to produce difficulty in
# the optimization process; pargep() engineered with flexibility to usually
# hit the proper solutions.
mygepA <- pargep(vec2lmom(c(1,0.305,0.270), lscale=FALSE))
mygepB <- pargep(vec2lmom(c(1,0.280,0.320), lscale=FALSE))

## End(Not run)
## Not run:
gep1 <- vec2par(c(2708, 3, 52), type="gep")
 lmr <- lmomgep(gep1);  print(lmr$lambdas)
gep2 <- pargep(lmr);    print(lmomgep(gep2)$lambdas)
# Note that we are close on matching the L-moments but we do
# not recover the parameters given because to shape parameters.
gep3 <- pargep(lmr, nk=1, nh=2);
x <- quagep(nonexceeds(), gep1)
x <- sort(c(x, quagep(nonexceeds(), gep2)))
plot(x, pdfgep(x, gep1), type="l", lwd=2)
lines(x, pdfgep(x, gep2), lwd=3, col=2)
lines(x, pdfgep(x, gep3), lwd=2, col=3)

## End(Not run)
```

---

pargev   *Estimate the Parameters of the Generalized Extreme Value Distribution*

Description

This function estimates the parameters of the Generalized Extreme Value distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomgev`.

Usage

```
pargev(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `gev`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "pargev".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmomgev`, `cdfgev`, `pdfgev`, `quagev`

Examples

```
lmr <- lmoms(rnorm(20))
pargev(lmr)
```

pargld *Estimate the Parameters of the Generalized Lambda Distribution*

Description

This function estimates the parameters of the Generalized Lambda distribution given the L-moments of the data in an ordinary L-moment object (`lmoms`) or a trimmed L-moment object (`TLmoms` for `t=1`). The relations between distribution parameters and L-moments are seen under `lmomgld`. There are no simple expressions for the parameters in terms of the L-moments. Consider that multiple parameter solutions are possible with the Generalized Lambda so some expertise in the distribution and other aspects are needed.

Usage

```
pargld(lmom, verbose=FALSE, initkh=NULL, eps=1e-3,
       aux=c("tau5", "tau6"), checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms`, `vec2lmom`, or `TLmoms` with `trim=0`.
`verbose`	A logical switch on the verbosity of output. Default is `verbose=FALSE`.
`initkh`	A vector of the initial guess of the κ and h parameters. No other regions of parameter space are consulted.
`eps`	A small term or threshold for which the square root of the sum of square errors in τ_3 and τ_4 is compared to to judge "good enough" for the alogrithm to order solutions based on smallest error as explained in next argument.
`aux`	Control the algorithm to order solutions based on smallest error in $\Delta\tau_5$ or $\Delta\tau_6$.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Details

Karian and Dudewicz (2000) summarize six regions of the κ and h space in which the Generalized Lambda distribution is valid for suitably choosen α. Numerical experimentation suggestions that the L-moments are not valid in Regions 1 and 2. However, initial guesses of the parameters within each region are used with numerous separate `optim` (the R function) efforts to perform a least sum-of-square errors on the following objective function

$$(\hat{\tau}_3 - \tilde{\tau}_3)^2 + (\hat{\tau}_4 - \tilde{\tau}_4)^2,$$

where $\hat{\tau}_r$ is the L-moment ratio of the data, $\tilde{\tau}_r$ is the estimated value of the L-moment ratio for the fitted distribution κ and h and τ_r is the actual value of the L-moment ratio.

For each optimization, a check on the validity of the parameters so produced is made—are the parameters consistent with the Generalized Lambda distribution? A second check is made on the validity of τ_3 and τ_4. If both validity checks return TRUE then the optimization is retained if its sum-of-square error is less than the previous optimum value. It is possible for a given solution to be found outside the starting region of the initial guesses. The surface generated by the τ_3 and τ_4 equations seen in lmomgld is complex–different initial guesses within a given region can yield what appear to be radically different κ and h. Users are encouraged to "play" with alternative solutions (see the verbose argument). A quick double check on the L-moments from the solved parameters using lmomgld is encouraged as well. Karvanen and others (2002, eq. 25) provide an equation expressing κ and h as equal (a symmetrical Generalized Lambda distribution) in terms of τ_4 and suggest that the equation be used to determine initial values for the parameters. The Karvanen equation is used on a semi-experimental basis for the final optimization attempt by pargld.

Value

An R list is returned if result='best'.

type	The type of distribution: gld.
para	The parameters of the distribution.
delTau5	Difference between the $\tilde{\tau}_5$ of the fitted distribution and true $\hat{\tau}_5$.
error	Smallest sum of square error found.
source	The source of the parameters: "pargld".
rest	An R data.frame of other solutions if found.

The rest of the solutions have the following:

xi	The location parameter of the distribution.
alpha	The scale parameter of the distribution.
kappa	The 1st shape parameter of the distribution.
h	The 2nd shape parameter of the distribution.
attempt	The attempt number that found valid TL-moments and parameters of GLD.
delTau5	The absolute difference between $\hat{\tau}_5^{(1)}$ of data to $\tilde{\tau}_5^{(1)}$ of the fitted distribution.
error	The sum of square error found.
initial_k	The starting point of the κ parameter.
initial_h	The starting point of the h parameter.
valid.gld	Logical on validity of the GLD—TRUE by this point.
valid.lmr	Logical on validity of the L-moments—TRUE by this point.
lowerror	Logical on whether error was less than eps—TRUE by this point.

Note

This function is a cumbersome method of parameter solution, but years of testing suggest that with supervision and the available options regarding the optimization that reliable parameter estimations result. The Tukey Lambda distribution is a special form of the GLD, see **Tukey Lambda Notes** section in **Details** of lmrdia46 for more details.

Author(s)

W.H. Asquith

Source

W.H. Asquith in Feb. 2006 with a copy of Karian and Dudewicz (2000) and again Feb. 2011.

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karvanen, J., Eriksson, J., and Koivunen, V., 2002, Adaptive score functions for maximum likelihood ICA: Journal of VLSI Signal Processing, v. 32, pp. 82–92.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

lmomgld, cdfgld, pdfgld, quagld, parTLgld

Examples

```
## Not run:
  X      <- sort( rgamma(202, 2) ) # simulate a skewed distribution
  lmr    <- lmoms(X)               # compute trimmed L-moments
  PARgld <- pargld(lmr)            # fit the GLD
  FF     <- pp(X)
  plot( FF,    X, col=8, cex=0.25)
  lines(FF, qlmomco(FF, PARgld)) # show the best estimate
  if(! is.null(PARgld$rest)) { #$
    n <- length(PARgld$rest$xi)
    other <- unlist(PARgld$rest[n, 1:4]) #$ # show alternative
    lines(FF, qlmomco(FF, vec2par(other, type="gld")), col="red")
  }
  # Note in the extraction of other solutions that no testing for whether
  # additional solutions were found is made. Also, it is quite possible
  # that the other solutions "[n,1:4]" is effectively another numerical
  # convergence on the primary solution. Some users of this example thus
  # might not see two separate lines. Users are encouraged to inspect the
  # rest of the solutions: print(PARgld$rest) #
## End(Not run)

## Not run:
  FF <- seq(0.01, 0.99, 0.01)
  plot(FF,  qlmomco(FF, vec2par(c(3.1446434, 2.943469, 7.4211316, 1.050537),
                                type="gld")), col="blue", type="l")
  lines(FF, qlmomco(FF, vec2par(c(0.4962471, 8.794038, 0.0082958, 0.228352),
                                type="gld")), col="red"           ) #
## End(Not run)
```

parglo *Estimate the Parameters of the Generalized Logistic Distribution*

Description

This function estimates the parameters of the Generalized Logistic distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomglo`.

Usage

```
parglo(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `glo`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "parglo".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmomglo`, `cdfglo`, `pdfglo`, `quaglo`

Examples

```
lmr <- lmoms(rnorm(20))
parglo(lmr)
## Not run:
# A then Ph.D. student, L. Read inquired in February 2014 about the relation between
# GLO and the "Log-Logistic" distributions:
par.glo  <- vec2par(c(10, .56, 0), type="glo")         # Define GLO parameters
par.lnlo <- c(exp(par.glo$para[1]), 1/par.glo$para[2]) # Equivalent LN-LO parameters
F <- nonexceeds(); qF <- qnorm(F) # use a real probability axis to show features
plot(qF, exp(quaglo(F, par.glo)), type="l", lwd=5, xaxt="n", log="y",
     xlab="", ylab="QUANTILE") # notice the exp() wrapper on the GLO quantiles
lines(qF, par.lnlo[1]*(F/(1-F))^(1/par.lnlo[2]), col=2, lwd=2) # eq. for LN-LO
add.lmomco.axis(las=2, tcl=0.5, side.type="RI", otherside.type="NPP")

## End(Not run)
```

---

pargno *Estimate the Parameters of the Generalized Normal Distribution*

Description

This function estimates the parameters of the Generalized Normal (Log-Normal3) distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomgno`.

Usage

```
pargno(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: gno.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "pargno".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmomgno, cdfgno, pdfgno, quagno, parln3

Examples

```
lmr <- lmoms(rnorm(20))
pargno(lmr)

## Not run:
x <- c(2.4, 2.7, 2.3, 2.5, 2.2, 62.4, 3.8, 3.1)
gno <- pargno(lmoms(x)) # triggers warning: Hosking's limit is Tau3=+-0.95
## End(Not run)
```

---

pargov *Estimate the Parameters of the Govindarajulu Distribution*

Description

This function estimates the parameters of the Govindarajulu distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between distribution parameters and L-moments also are seen under lmomgov. The β is estimated as

$$\beta = -\frac{(4\tau_3 + 2)}{(\tau_3 - 1)},$$

and α then ξ are estimated for *unknown* ξ as

$$\alpha = \lambda_2 \frac{(\beta + 2)(\beta + 3)}{2\beta}, \text{ and}$$

$$\xi = \lambda_1 - \frac{2\alpha}{(\beta + 2)},$$

and α is estimated for *known* ξ as

$$\alpha = (\lambda_1 - \xi)\frac{(\beta + 2)}{2}.$$

The shape preservation for this distribution is an ad hoc decision. It could be that for given ξ, that solutions could fall back to estimating ξ and α from λ_1 and λ_2 only. Such as solution would rely on $\tau_2 = \lambda_2/\lambda_1$ with β estimated as

$$\beta = \frac{3\tau_2}{(1-\tau_2)}, \text{ and}$$

$$\alpha = \lambda_1 \frac{(\beta+2)}{2},$$

but such a practice yields remarkable changes in shape for this distribution even if the provided ξ precisely matches that from a previous parameter estimation for which the ξ was treated as unknown.

Usage

```
pargov(lmom, xi=NULL, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`xi`	An optional lower limit of the distribution. If not `NULL`, the B is still uniquely determined by τ_3, the α is adjusted so that the given lower bounds is honored. It is generally accepted to let the distribution fitting process determine its own lower bounds so `xi=NULL` should suffice in many circumstances.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `gov`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "pargov".

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Nair, N.U., Sankaran, P.G., Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

Nair, N.U., Sankaran, P.G., and Vineshkumar, B., 2012, The Govindarajulu distribution—Some Properties and applications: Communications in Statistics, Theory and Methods, v. 41, no. 24, pp. 4391–4406.

See Also

lmomgov, cdfgov, pdfgov, quagov

Examples

```
lmr <- lmoms(rnorm(20))
pargov(lmr)

lmr <- vec2lmom(c(1391.8, 215.68, 0.01655, 0.09628))
pargov(lmr)$para             # see below
#         xi       alpha        beta
# 868.148125 1073.740595    2.100971
pargov(lmr, xi=868)$para     # see below
#         xi       alpha        beta
# 868.000000 1074.044324    2.100971
pargov(lmr, xi=100)$para     # see below
#         xi       alpha        beta
# 100.000000 2648.817215    2.100971
```

---

pargpa *Estimate the Parameters of the Generalized Pareto Distribution*

Description

This function estimates the parameters of the Generalized Pareto distribution given the L-moments of the data in an ordinary L-moment object (lmoms) or a trimmed L-moment object (TLmoms for t=1). The relations between distribution parameters and L-moments are seen under lmomgpa or lmomTLgpa.

Usage

```
pargpa(lmom, zeta=1, xi=NULL, checklmom=TRUE, ...)
```

Arguments

lmom	An L-moment object created by lmoms, TLmoms with trim=0, or vec2lmom.
zeta	The right censoring fraction. If less than unity then a dispatch to the pargpaRC is made and the lmom argument must contain the B-type L-moments. If the data are not right censored, then this value must be left alone to the default of unity.
xi	The lower limit of the distribution. If ξ is known, then alternative algorithms are used.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
...	Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: gpa.
para	The parameters of the distribution.
source	The source of the parameters: "pargpa".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmomgpa, cdfgpa, pdfgpa, quagpa

Examples

```
X   <- rexp(200)
lmr <- lmoms(X)
P1  <- pargpa(lmr)
P2  <- pargpa(lmr, xi=0.25)

## Not run:
```

```
F <- nonexceeds()
plot(pp(X), sort(X))
lines(F, quagpa(F,P1))         # black line
lines(F, quagpa(F,P2), col=2)  # red line

## End(Not run)
```

pargpaRC *Estimate the Parameters of the Generalized Pareto Distribution with Right-Tail Censoring*

Description

This function estimates the parameters (ξ, α, and κ) of the Generalized Pareto distribution given the "B"-type L-moments (through the B-type probability-weighted moments) of the data under right censoring conditions (see pwmRC). The relations between distribution parameters and L-moments are seen under lmomgpaRC.

Usage

```
pargpaRC(lmom, zeta=1, xi=NULL, lower=-1, upper=20, checklmom=TRUE, ...)
```

Arguments

lmom	A B-type L-moment object created by a function such as pwm2lmom from B-type probability-weighted moments from pwmRC.
zeta	The compliment of the right-tail censoring fraction. The number of samples observed (noncensored) divided by the total number of samples.
xi	The lower limit of the distribution. If ξ is known, then alternative algorithms are used.
lower	The lower value for κ for a call to the optimize function. For the L-moments of the distribution to be valid $\kappa > -1$.
upper	The upper value for κ for a call to the optimize function. Hopefully, a large enough default is chosen for real-world data sets.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
...	Other arguments to pass.

Details

The optimize R function is used to numerically solve for the shape parameter κ. No test or evaluation is made on the quality of the minimization. Users should consult the contents of the optim portion of the returned list. Finally, this function should return the same parameters if $\zeta = 1$ as the pargpa function.

Value

An R list is returned.

type	The type of distribution: gpa.
para	The parameters of the distribution.
zeta	The compliment of the right-tail censoring fraction.
source	The source of the parameters: "pargpaRC".
optim	The list returned by the R function optimize.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

lmomgpa, lmomgpaRC, pargpa, cdfgpa, pdfgpa, quagpa

Examples

```
n         <- 60 # samplesize
para      <- vec2par(c(1500,160,.3),type="gpa") # build a GPA parameter set
fakedata  <- quagpa(runif(n),para) # generate n simulated values
threshold <- 1700 # a threshold to apply the simulated censoring
fakedata  <- sapply(fakedata,function(x) { if(x > threshold)
                                          return(threshold) else return(x) })
lmr       <- lmoms(fakedata) # Ordinary L-moments without considering
                             # that the data is censored
estpara   <- pargpa(lmr) # Estimated parameters of parent

pwm2     <- pwmRC(fakedata,threshold=threshold) # compute censored PWMs
typeBpwm <- pwm2$Bbetas # the B-type PWMs
zeta     <- pwm2$zeta # the censoring fraction

cenpara <- pargpaRC(pwm2lmom(typeBpwm),zeta=zeta) # Estimated parameters
F       <- nonexceeds() # nonexceedance probabilities for plotting purposes

# Visualize some data
plot(F,quagpa(F,para), type='l', lwd=3) # The true distribution
lines(F,quagpa(F,estpara), col=3) # Green estimated in the ordinary fashion
lines(F,quagpa(F,cenpara), col=2) # Red, consider that the data is censored
# now add in what the drawn sample looks like.
```

```
PP <- pp(fakedata) # plotting positions of the data
points(PP,sort(fakedata)) # sorting is needed!
# Interpretation. You should see that the red line more closely matches
# the heavy black line. The green line should be deflected to the right
# and pass through the values equal to the threshold, which reflects the
# much smaller L-skew of the ordinary L-moments compared to the type-B
# L-moments.

# Assertion, given some PWMs or L-moments, if zeta=1 then the parameter
# estimates must be identical. The following provides a demonstration.
para1 <- pargpaRC(pwm2lmom(typeBpwm),zeta=1)
para2 <- pargpa(pwm2lmom(typeBpwm))
str(para1); str(para2)

# Assertion as previous assertion, let us trigger different optimizer
# algorithms with a non-NULL xi parameter and see if the two parameter
# lists are the same.
para1 <- pargpaRC(pwm2lmom(typeBpwm), zeta=zeta)
para2 <- pargpaRC(pwm2lmom(typeBpwm), xi=para1$para[1], zeta=zeta)
str(para1); str(para2)
```

---

pargum *Estimate the Parameters of the Gumbel Distribution*

Description

This function estimates the parameters of the Gumbel distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomgum`.

Usage

```
pargum(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `gum`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "pargum".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmomgum`, `cdfgum`, `pdfgum`, `quagum`

Examples

```
lmr <- lmoms(rnorm(20))
pargum(lmr)
```

`parkap` *Estimate the Parameters of the Kappa Distribution*

Description

This function estimates the parameters of the Kappa distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomkap`, but of relevance to this documentation, the upper bounds of L-kurtosis (τ_4) and a function of L-skew (τ_3) is given by

$$\tau_4 < \frac{5\tau_3^2 + 1}{6}$$

This bounds is equal to the Generalized Logistic distribution (`parglo`) and failure occurs if this upper bounds is exceeded. However, the argument `snap.tau4`, if set, will set τ_4 equal to the upper bounds of τ_4 of the distribution to the relation above. This value of τ_4 should be close enough numerically The argument `nudge.tau4` is provided to offset τ_4 downward just a little. This keeps the relation operator as "$<$" in the bounds above to match Hosking's tradition as his sources declare "$\geq$" as above the GLO. The nudge here hence is not zero, which is a little different compared to the conceptually similar snapping in `paraep4`.

Usage

```
parkap(lmom, checklmom=TRUE,
             snap.tau4=FALSE, nudge.tau4=sqrt(.Machine$double.eps), ...)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
snap.tau4	A logical to "snap" the τ_4 downwards to the lower boundary if the given τ_4 is greater than the boundary described as above.
nudge.tau4	An offset to the snapping of τ_4 intended to move τ_4 just below the upper bounds. (The absolute value of the nudge is made internally to ensure only downward adjustment by a subtraction operation.)
...	Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: kap.
para	The parameters of the distribution.
source	The source of the parameters: "parkap".
support	The support (or range) of the fitted distribution.
ifail	A numeric failure code.
ifailtext	A text message for the failure code.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1994, The four-parameter kappa distribution: IBM Journal of Reserach and Development, v. 38, no. 3, pp. 251–258.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmomkap, cdfkap, pdfkap, quakap

Examples

```
lmr <- lmoms(rnorm(20))
parkap(lmr)

## Not run:
parkap(vec2lmom(c(0,1,.3,.8)), snap.tau4=TRUE) # Tau=0.8 is way above the GLO.
## End(Not run)
```

---

parkmu — *Estimate the Parameters of the Kappa-Mu Distribution*

Description

This function estimates the parameters (ν and α) of the Kappa-Mu ($\kappa : \mu$) distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between distribution parameters and L-moments are seen under lmomkmu.

The basic approach for parameter optimization is to extract initial guesses for the parameters from the table KMU_lmompara_bykappa in the .lmomcohash environment. The parameters having a minimum Euclidean error as controlled by three arguments are used for initial guesses in a Nelder-Mead simplex multidimensional optimization using the R function optim and default arguments.

Limited testing indicates that of the "error term controlling options" that the default values as shown in the Usage section seem to provide superior performance in terms of recovering the *a priori known* parameters in experiments. It seems that only Euclidean optimization using L-skew and L-kurtosis is preferable, but experiments show the general algorithm to be slow.

Usage

```
parkmu(lmom, checklmom=TRUE, checkbounds=TRUE,
         alsofitT3=FALSE, alsofitT3T4=FALSE, alsofitT3T4T5=FALSE,
         justfitT3T4=TRUE, boundary.tolerance=0.001,
         verbose=FALSE, trackoptim=TRUE, ...)
```

Arguments

lmom	An L-moment object created by lmoms or pwm2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality).
checkbounds	Should the L-skew and L-kurtosis boundaries of the distribution be checked.
alsofitT3	Logical when true will add the error term $(\hat{\tau}_3 - \tau_3)^2$ to the sum of square errors for the mean and L-CV.
alsofitT3T4	Logical when true will add the error term $(\hat{\tau}_3 - \tau_3)^2 + (\hat{\tau}_4 - \tau_4)^2$ to the sum of square errors for the mean and L-CV.
alsofitT3T4T5	Logical when true will add the error term $(\hat{\tau}_3 - \tau_3)^2 + (\hat{\tau}_4 - \tau_4)^2 + (\hat{\tau}_5 - \tau_5)^2$ to the sum of square errors for the mean and L-CV.

justfitT3T4 — Logical when true will only consider the sum of squares errors for L-skew and L-kurtosis as mathematically shown for alsofitT3T4.

boundary.tolerance — A fudge number to help guide how close to the boundaries an arbitrary list of τ_3 and τ_4 can be to consider them formally in or out of the attainable $\{\tau_3, \tau_4\}$ domain.

verbose — A logical to control a level of diagnostic output.

trackoptim — A logical to control specific messaging through each iteration of the objective function.

... — Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: kmu.
para	The parameters of the distribution.
source	The source of the parameters: "parkmu".

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

lmomkmu, cdfkmu, pdfkmu, quakmu

Examples

```
## Not run:
   par1 <- vec2par(c(0.7, 0.2), type="kmu")
   lmr1 <- lmomkmu(par1, nmom=4)
   par2.1 <- parkmu(lmr1, alsofitT3=TRUE,   verbose=TRUE, trackoptim=TRUE)
   par2.1$para
   par2.2 <- parkmu(lmr1, alsofitT3T4=TRUE, verbose=TRUE, trackoptim=TRUE)
   par2.2$para
   par2.3 <- parkmu(lmr1, alsofitT3=FALSE,  verbose=TRUE, trackoptim=TRUE)
   par2.3$para
   par2.4 <- parkmu(lmr1, justfitT3T4=TRUE, verbose=TRUE, trackoptim=TRUE)
   par2.4$para
   x <- seq(0,3,by=.01)
   plot(x,  pdfkmu(x, par1), type="l", lwd=6, col=8, ylim=c(0,5))
   lines(x, pdfkmu(x, par2.1), col=2, lwd=2, lty=2)
   lines(x, pdfkmu(x, par2.2), col=4)
```

```
   lines(x, pdfkmu(x, par2.3), col=3, lty=3, lwd=2)
   lines(x, pdfkmu(x, par2.4), col=5, lty=2, lwd=2)

## End(Not run)
## Not run:
   par1 <- vec2par(c(1, 0.65), type="kmu")
   lmr1 <- lmomkmu(par1, nmom=4)
   par2.1 <- parkmu(lmr1, alsofitT3=TRUE,   verbose=TRUE, trackoptim=TRUE)
   par2.1$para # eta=1.0  mu=0.65
   par2.2 <- parkmu(lmr1, alsofitT3T4=TRUE, verbose=TRUE, trackoptim=TRUE)
   par2.2$para # eta=1.0  mu=0.65
   par2.3 <- parkmu(lmr1, alsofitT3=FALSE,  verbose=TRUE, trackoptim=TRUE)
   par2.3$para # eta=8.5779  mu=0.2060
   par2.4 <- parkmu(lmr1, justfitT3T4=TRUE, verbose=TRUE, trackoptim=TRUE)
   par2.4$para # eta=1.0 mu=0.65
   x <- seq(0,3,by=.01)
   plot(x,  pdfkmu(x, par1), type="l", lwd=6, col=8, ylim=c(0,1))
   lines(x, pdfkmu(x, par2.1), col=2, lwd=2, lty=2)
   lines(x, pdfkmu(x, par2.2), col=4)
   lines(x, pdfkmu(x, par2.3), col=3, lty=3, lwd=2)
   lines(x, pdfkmu(x, par2.4), col=5, lty=2, lwd=2)
   lines(x, dlmomco(x, lmom2par(lmr1, type="gam")),  lwd=2, col=2)
   lines(x, dlmomco(x, lmom2par(lmr1, type="ray")),  lwd=2, col=2, lty=2)
   lines(x, dlmomco(x, lmom2par(lmr1, type="rice")), lwd=2, col=4, lty=2)

## End(Not run)
```

---

parkur *Estimate the Parameters of the Kumaraswamy Distribution*

Description

This function estimates the parameters of the Kumaraswamy distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomkur`.

Usage

```
parkur(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `kur`.
`para`	The parameters of the distribution.
`err`	The convergence error.
`convergence`	Logical showing whether error convergence occurred.
`source`	The source of the parameters: “parkur”.

Author(s)

W.H. Asquith

References

Jones, M.C., 2009, Kumaraswamy’s distribution—A beta-type distribution with some tractability advantages: Statistical Methodology, v. 6, pp. 70–81.

See Also

`lmomkur`, `cdfkur`, `pdfkur`, `quakur`

Examples

```
lmr <- lmoms(runif(20)^2)
parkur(lmr)

kurpar <- list(para=c(1,1), type="kur");
lmr <- lmomkur(kurpar)
parkur(lmr)

kurpar <- list(para=c(0.1,1), type="kur");
lmr <- lmomkur(kurpar)
parkur(lmr)

kurpar <- list(para=c(1,0.1), type="kur");
lmr <- lmomkur(kurpar)
parkur(lmr)

kurpar <- list(para=c(0.1,0.1), type="kur");
lmr <- lmomkur(kurpar)
parkur(lmr)
```

parlap — *Estimate the Parameters of the Laplace Distribution*

Description

This function estimates the parameters of the Laplace distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between distribution parameters and sample L-moments are simple, but there are two methods. The first method, which is the only one implemented in **lmomco**, jointly uses $\lambda_1, \lambda_2, \lambda_3$, and λ_4. The mathematical expressions are

$$\xi = \lambda_1 - 50/31 \times \lambda_3 \text{and}$$

$$\alpha = 1.4741\lambda_2 - 0.5960\lambda_4.$$

The alternative and even simpler method only uses λ_1 and λ_2. The mathematical expressions are

$$\xi = \lambda_1 \text{ and}$$

$$\alpha = \frac{4}{3}\lambda_2.$$

The user could easily estimate the parameters from the L-moments and use vec2par to create a parameter object.

Usage

```
parlap(lmom, checklmom=TRUE, ...)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
...	Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: lap.
para	The parameters of the distribution.
source	The source of the parameters: “parlap”.

Note

The decision to use only one of the two systems of equations for Laplace fitting is largely arbitrary, but it seems most fitting to use four L-moments instead of two.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: IBM Research Report RC12210, T.J. Watson Research Center, Yorktown Heights, New York.

See Also

lmomlap, cdflap, pdflap, qualap

Examples

```
lmr <- lmoms(rnorm(20))
parlap(lmr)
```

---

parlmrq *Estimate the Parameters of the Linear Mean Residual Quantile Function Distribution*

Description

This function estimates the parameters of the Linear Mean Residual Quantile Function distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under lmomlmrq.

Usage

```
parlmrq(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by lmoms or vec2lmom.
`checklmom`	Should the `lmom` be checked for validity using the are.lmom.valid function. Normally this should be left as the default.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `lmrq`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “parlmrq”.

Author(s)

W.H. Asquith

References

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

See Also

lmomlmrq, cdflmrq, pdflmrq, qualmrq

Examples

```
lmr <- lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2))
parlmrq(lmr)
```

---

parln3 *Estimate the Parameters of the 3-Parameter Log-Normal Distribution*

Description

This function estimates the parameters (ζ, lower bounds; μ_{log}, location; and σ_{log}, scale) of the Log-Normal3 distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between distribution parameters and L-moments are seen under lmomln3. The function uses algorithms of the Generalized Normal for core computations. Also, if $\tau_3 \leq 0$, then the Log-Normal3 distribution can not be fit, however reversing the data alleviates this problem.

Usage

```
parln3(lmom, zeta=NULL, checklmom=TRUE, ...)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
zeta	Lower bounds, if NULL then solved for.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
...	Other arguments to pass.

Details

Let the L-moments by in variable `lmr`, if the ζ (lower bounds) is unknown, then the algorithms return the same fit as the Generalized Normal will attain. However, `pargno` does not have intrinsic control on the lower bounds and `parln3` does. The λ_1, λ_2, and τ_3 are used in the fitting for `pargno` and `parln3` but only λ_1 and λ_2 are used when the ζ is provided as in `parln3(lmr, zeta=0)`. In otherwords, if ζ is known, then τ_3 is not used and shaping comes from the choice of ζ.

Value

An R `list` is returned.

`type`	The type of distribution: `ln3`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "parln3".

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`lmomln3`, `cdfln3`, `pdfln3`, `qualn3`, `pargno`

Examples

```
lmr <- lmoms(rnorm(20))
parln3(lmr)

## Not run:
# Handling condition of negative L-skew
# Data reversal looks like: Y <- -X, but let us use an example
# on the L-moments themselves.
lmr.pos <- vec2lmom(c(100, 45, -0.1)) # parln3(lmr.pos) fails
lmr.neg <- lmr.pos
lmr.neg$lambdas[1] <- -lmr.neg$lambdas[1]
lmr.neg$ratios[3]  <- -lmr.neg$ratios[3]
F <- nonexceeds()
plot(F, -qualn3(1-F, parln3(lmr.neg)), type="l", lwd=3, col=2) # red line
lines(F, quagno(F, pargno(lmr.pos))) # black line
## End(Not run)
```

parnor *Estimate the Parameters of the Normal Distribution*

Description

This function estimates the parameters of the Normal distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relation between distribution parameters and L-moments is seen under `lmomnor`.

There are interesting parallels between λ_2 (L-scale) and σ (standard deviation). The σ estimated from this function will not necessarily equal the output of the `sd` function of R, and in fact such equality is not expected. This disconnect between the parameters of the Normal distribution and the moments (sample) of the same name can be most confusing to young trainees in statistics. The Pearson Type III is similar. See the extended example for further illustration.

Usage

```
parnor(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `nor`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "parnor".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmomnor, cdfnor, pdfnor, quanor

Examples

```
lmr <- lmoms(rnorm(20))
parnor(lmr)

# A more extended example to explore the differences between an
# L-moment derived estimate of the standard deviation and R's sd()
true.std <- 15000 # select a large standard deviation
std         <- vector(mode = "numeric") # vector of sd()
std.by.lmom <- vector(mode = "numeric") # vector of L-scale values
sam <- 7   # number of samples to simulate
sim <- 100 # perform simulation sim times
for(i in seq(1,sim)) {
  Q <- rnorm(sam,sd=15000) # draw random normal variates
  std[i] <- sd(Q) # compute standard deviation
  lmr <- lmoms(Q) # compute the L-moments
  std.by.lmom[i] <- lmr$lambdas[2] # save the L-scale value
}
# convert L-scale values to equivalent standard deviations
std.by.lmom      <- sqrt(pi)*std.by.lmom

# compute the two biases and then output
# see how the standard deviation estimated through L-scale
# has a smaller bias than the usual (product moment) standard
# deviation. The unbiasness of L-moments is demonstrated.
std.bias         <- true.std - mean(std)
std.by.lmom.bias <- true.std - mean(std.by.lmom)
cat(c(std.bias,std.by.lmom.bias,"\n"))
```

---

parpdq3 *Estimate the Parameters of the Polynomial Density-Quantile3 Distribution*

Description

This function estimates the parameters of the Polynomial Density-Quantile3 distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between the distribution parameters and L-moments are seen under lmompdq3.

Usage

```
parpdq3(lmom, checklmom=TRUE)
```

Arguments

- `lmom` — An L-moment object created by `lmoms` or `vec2lmom`.
- `checklmom` — Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is unlikely that the L-moments will not be viable. However, for some circumstances or large simulation exercises then one might want to bypass this check.

Value

An R `list` is returned.

- `type` — The type of distribution: `pdq3`.
- `para` — The parameters of the distribution.
- `ifail` — A numeric field connected to the `ifailtext`; a value of 0 indicates fully successful operation of the function.
- `ifailtext` — A message, instead of a warning, about the internal operations or operational limits of the function.
- `source` — The source of the parameters: "parpdq3".

Note

The following is a study of the performance of `parpdq3` as the upper limit of the shape parameter κ is approached. The algorithms have the ability to estimate the κ reliabily, it is the scale parameter α that breaks down and hence there is a hard-wired setting of $|\kappa| > 0.98$ in which a warning is issue in `parpdq3` about α reliability:

```
A <- 10
K <- seq(0.8, 1, by=0.0001)
K <- sort(c(-K, K))
As <- Ks <- rep(NA, length(K))
for(i in 1:length(K)) {
  para <- list(para=c(0, A, K[i]), type="pdq3")
  As[i] <- parpdq3( lmompdq3(para) )$para[2]
  Ks[i] <- parpdq3( lmompdq3(para) )$para[3]
}
plot( K, (As-A)/A, type="l", col="red")
abline(v=c(-0.98, +0.98)) # heuristically determined threshold
```

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

lmompdq3, cdfpdq3, pdfpdq3, quapdq3

Examples

```
para <- list(para=c(0, 0.4332, -0.7029), type="pdq3")
parpdq3(lmompdq3(para))$para

para <- list(para=c(0, 0.4332, 0.7029), type="pdq3")
parpdq3(lmompdq3(para))$para

para <- list(para=c(0, 0.4332, 1-sqrt(.Machine$double.eps)), type="pdq3")
parpdq3(lmompdq3(para))$para

para <- list(para=c(0, 0.4332, -1+sqrt(.Machine$double.eps)), type="pdq3")
parpdq3(lmompdq3(para))$para

para <- list(para=c(0, 0.4332, +0.0001), type="pdq3")
parpdq3(lmompdq3(para))$para

para <- list(para=c(0, 0.4332, -0.0001), type="pdq3")
parpdq3(lmompdq3(para))$para

para <- list(para=c(0, 0.4332, 0), type="pdq3")
parpdq3(lmompdq3(para))$para
```

---

parpdq4 — *Estimate the Parameters of the Polynomial Density-Quantile4 Distribution*

Description

This function estimates the parameters of the Polynomial Density-Quantile4 distribution given the L-moments of the data in an L-moment object such as that returned by lmoms. The relations between the distribution parameters and L-moments are seen under lmompdq4.

Usage

```
parpdq4(lmom, checklmom=TRUE, snapt4uplimit=TRUE)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is unlikely that the L-moments will not be viable. However, for some circumstances or large simulation exercises then one might want to bypass this check.
snapt4uplimit	A logical controlling the behavior of the function for τ_4 exceeding an operational upper margin and whether the incoming τ_4 can be snapped down to this margin (see **Note**).

Value

An R `list` is returned.

`type`	The type of distribution: `pdq4`.
`para`	The parameters of the distribution.
`ifail`	A numeric field connected to the `ifailtext`; a value of 0 indicates fully successful operation of the function.
`ifailtext`	A message, instead of a warning, about the internal operations or operational limits of the function.
`source`	The source of the parameters: “parpdq4”.

Note

Upper Limit of the Shape Parameter—The following is a study of the performance of `parpdq4` as the upper limit of the shape parameter κ is approached. The algorithms have the ability to estimate the κ reliabily, it is the scale parameter α that breaks down and hence there is a hard-wired setting of $\kappa > 0.99$ in which a message is issued to `ifail` about α reliability:

```
A <- 100
K <- seq(0.8, 1, by=0.0001)
As <- Ks <- rep(NA, length(K))
for(i in 1:length(K)) {
  para  <- list(para=c(0, A, K[i]), type="pdq4")
  pdq4  <- parpdq4(lmompdq4(para), snapt4uplimit=FALSE)
  As[i] <- pdq4$para[2]
  Ks[i] <- pdq4$para[3]
}
plot( K, (As-A)/A, type="l", col="red")
abline(v=0.99) # heuristically determined threshold
```

Lower Limit of the Shape Parameter—The lower limit of κ does not really exist but as $\kappa \rightarrow -\infty$, the qualty of the α operation will degrade. The approach in the code involves an R function `uniroot()` operation and the lower limit is not set to `-Inf` but is set within sources as the value
`-.Machine$double.xmax^(1/64)`,
which is not too small of a number, but the τ_4 associated with this limit is -0.2499878576145593, which is extremely close to $\tau_4 > -1/4$ lower limit. The implementation here will snap incoming τ_4 to a threshold towards zero as

```
TAU4 <- "users tau4"
smallTAU4 <- -0.2499878576145593
if(TAU4 < smallTAU4) TAU4 <- smallTAU4 + sqrt(.Machine$double.eps)
print(TAU4, 16) # -0.2499878427133981
```

and this snapping produces an operational lower bounds of κ of -65455.6715146775. This topic can be explored by operations such as

```
# Have tau4 but with internals to protect quality of the
# alpha estimation and speed root-solving the kappa, there
# is an operational lower bounds of tau4. Here lower limit
# tau4 = -0.25 and the operations below return -0.2499878.
lmompdq4(parpdq4(vec2lmom(c(0, 100, 0, -1/4))))$ratios[4]
```

Upper Operational Limit of L-kurtosis—The script below explores the operational limit of τ_4 within the algorithms themselves. It is seen in the computations that breakdown in the reverse computation of the τ_4 from the parameters begins at $\tau_4 >= 0.867$. As a result, the argument snapt4upmargin by default and convenience could trigger snapping the solution to this upper limit (see section **Even Lower Maximum Operational Limit of L-kurtosis**).

```
T4s <- seq(0.8, 0.9, by=0.001) # sweeping through very high Tau4
unit_std <- 1/sqrt(pi)
FF <- pnorm(seq(-6, 6, by=0.01))
plot(0,0, type="n", xlim=range(qnorm(FF)), ylim=c(-6, 6),
          xlab="Standard Normal Variate", ylab="Quantile")
for(i in 1:length(T4s)) {
  lmr  <- vec2lmom(c(0, unit_std, 0, T4s[i]))
  pdq4 <- parpdq4(lmr, snapt4uplimit=FALSE)
  lmr4 <- lmompdq4(pdq4)
  lines(qnorm(FF), quapdq4(FF, pdq4))
  err1 <- theoLmoms(pdq4)$lambdas[2] - unit_std
  err2 <-            lmr4$lambdas[2] - unit_std
  vals <- c(T4s[i], pdq4$para[3], err1, err2)
  names(vals) <- c("Tau4", "Kappa", "Err1(theoLmoms)", "Err2(lmompdq4)")
  print(vals) # both methods of Lambda2 estimation
} # working and degenerates at Tau4 >= 0.867, so use 0.866 as a margin
```

The problem geometrically is, as the τ_4 becomes very "large", that the distribution is become so peaked that its variation will be degenerating to zero, which is not compatible with the infinite limits of the distribution. Presumably beyond $\tau_4 >= 0.867$, the TL-moments could be used with further algorithmic development. There are other difficulties though in the next example as τ_4 gets large.

Even Lower Maximum Operational Limit of L-kurtosis—Further study of the limits of maximum operational limit of τ_4 can be made for reliable use of the basic internal functions of R. Consider the following code:

```
T4s <- seq(0.4, 0.9, by=0.002)
errs <- vector(mode="numeric", length(T4s))
for(i in 1:length(T4s)) {
  lmra <- vec2lmom(c(0, 1, 0, T4s[i]))
  para <- parpdq4(lmra, snapt4uplimit=FALSE)
  lmrb <- lmompdq4(para)
  errs[i] <- abs(lmra$lambdas[4] - lmrb$lambdas[4])/lmra$lambdas[4]
  print(c(T4s[i], errs[i], para$para[3]))
}
plot(T4s, errs, ylab="abs(Lambda4 - EstLambda4)/Lambda4", col="red")
abline(v=0.845) # so use 0.845 as a lower margin
```

The $\tau_4 >= 0.845$ is therefore a more defensive upper limit for operational purposes of the **lmomco** package.

Lower Limit Performance of L-kurtosis—The lower limit of $\tau_4 = -1/4$ for the distribution is a statement of pure bimodality (two sides of a coin, as a matter of speaking). Visualization of the quantile function at the lower limit of τ_4 in the recipe that follows shows this fact with two flat-line segments of solid red lines with the change over at right angles at standard normal variate of zero. Then the τ_4 is nudged away from the lower limit just a little and replotted as the dashed line. Two other lines, but still for $\tau_4 < 0$, are shown in red and dark green. Finally, the demonstration ends with a magenta line for $\tau_4 = 0$.

```
FF <- pnorm(seq(-6, 6, by=0.01))
plot(0,0, type="n", xlim=range(qnorm(FF)), ylim=c(-6, 6),
          xlab="Standard Normal Variate", ylab="Quantile")
pdq4 <- parpdq4(vec2lmom(c(0, 1/sqrt(pi), 0, -1/4     )))
lines(qnorm(FF), quapdq4(FF, pdq4), col="red"   )
pdq4 <- parpdq4(vec2lmom(c(0, 1/sqrt(pi), 0, -1/4+0.03)))
lines(qnorm(FF), quapdq4(FF, pdq4), col="red", lty=2) # dashed
pdq4 <- parpdq4(vec2lmom(c(0, 1/sqrt(pi), 0, -1/8     )))
lines(qnorm(FF), quapdq4(FF, pdq4), col="darkgreen")
pdq4 <- parpdq4(vec2lmom(c(0, 1/sqrt(pi), 0, -1/16    )))
lines(qnorm(FF), quapdq4(FF, pdq4), col="blue"   )
pdq4 <- parpdq4(vec2lmom(c(0, 1/sqrt(pi), 0, 0        )))
lines(qnorm(FF), quapdq4(FF, pdq4), col="magenta")
```

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

lmompdq4, cdfpdq4, pdfpdq4, quapdq4

Examples

```
# Normal, Hosking (2007, p.2883)
para <- list(para=c(0, 0.4332, -0.7029), type="pdq4")
parpdq4(lmompdq4(para))$para
# parameter reversion shown

para <- list(para=c(0, 0.4332,  0.7029), type="pdq4")
parpdq4(lmompdq4(para))$para
# parameter reversion shown with sign change kappa

## Not run:
```

```
  # other looks disabled for check --timings
  para <- list(para=c(0, 0.4332, 0.97), type="pdq4")
  parpdq4(lmompdq4(para))$para
  # see now that alpha changing in fourth decimal as kappa
  # approaches the 0.98 threshold (see Note)

  # make two quick checks near zero and then zero
  para <- list(para=c(0, 0.4332, +0.0001), type="pdq4")
  parpdq4(lmompdq4(para))$para
  para <- list(para=c(0, 0.4332, -0.0001), type="pdq4")
  parpdq4(lmompdq4(para))$para
  para <- list(para=c(0, 0.4332, 0), type="pdq4")
  parpdq4(lmompdq4(para))$para #
## End(Not run)
```

---

parpe3 *Estimate the Parameters of the Pearson Type III Distribution*

Description

This function estimates the parameters of the Pearson Type III distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The L-moments in terms of the parameters are complicated and solved numerically. For the implementation in **lmomco**, the three parameters are μ, σ, and γ for the mean, standard deviation, and skew, respectively.

Usage

```
parpe3(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `pe3`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “parpe3”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmompe3`, `cdfpe3`, `pdfpe3`, `quape3`

Examples

```
lmr <- lmoms(rnorm(20))
parpe3(lmr)
```

`parray` *Estimate the Parameters of the Rayleigh Distribution*

Description

This function estimates the parameters of the Rayleigh distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are

$$\alpha = \frac{2\lambda_2\sqrt{\pi}}{\sqrt{2}-1},$$

and

$$\xi = \lambda_1 - \alpha\sqrt{\pi/2}.$$

Usage

```
parray(lmom, xi=NULL, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`xi`	The lower limit of the distribution. If ξ is known then alternative algorithms are triggered and only the first L-moment is required for fitting.

checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
...	Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: ray.
para	The parameters of the distribution.
source	The source of the parameters: "parray".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

See Also

lmomray, cdfray, pdfray, quaray

Examples

```
lmr <- lmoms(rnorm(20))
parray(lmr)
```

---

parrevgum — *Estimate the Parameters of the Reverse Gumbel Distribution*

Description

This function estimates the parameters of the Reverse Gumbel distribution given the type-B L-moments of the data in an L-moment object such as that returned by pwmRC using pwm2lmom. This distribution is important in the analysis of censored data. It is the distribution of a logarithmically transformed 2-parameter Weibull distribution. The relations between distribution parameters and L-moments are

$$\alpha = \lambda_2^B/\{\log(2) + \mathrm{Ei}(-2\log(1-\zeta)) - \mathrm{Ei}(-\log(1-\zeta))\}$$

and

$$\xi = \lambda_1^B + \alpha\{\mathrm{Ei}(-\log(1-\zeta))\},$$

where ζ is the compliment of the right-tail censoring fraction of the sample or the nonexceedance probability of the right-tail censoring threshold, and $\mathrm{Ei}(x)$ is the exponential integral defined as

$$\mathrm{Ei}(X) = \int_X^{\infty} x^{-1} e^{-x} \mathrm{d}x,$$

where $\mathrm{Ei}(-\log(1-\zeta)) \rightarrow 0$ as $\zeta \rightarrow 1$ and $\mathrm{Ei}(-\log(1-\zeta))$ can not be evaluated as $\zeta \rightarrow 0$.

Usage

```
parrevgum(lmom, zeta=1, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` through `pwmRC` or other L-moment type object. The user intervention of the `zeta` differentiates this distribution (and this function) from similar parameter estimation functions in the **lmomco** package.
`zeta`	The compliment of the right censoring fraction. Number of samples observed (noncensored) divided by the total number of samples.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `revgum`.
`para`	The parameters of the distribution.
`zeta`	The compliment of the right censoring fraction. Number of samples observed (noncensored) divided by the total number of samples.
`source`	The source of the parameters: “parrevgum”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

`lmomrevgum`, `cdfrevgum`, `pdfrevgum`, `quarevgum`, `pwm2lmom`, `pwmRC`

Examples

```
# See p. 553 of Hosking (1995)
# Data listed in Hosking (1995, table 29.3, p. 553)
D <- c(-2.982, -2.849, -2.546, -2.350, -1.983, -1.492, -1.443,
       -1.394, -1.386, -1.269, -1.195, -1.174, -0.854, -0.620,
       -0.576, -0.548, -0.247, -0.195, -0.056, -0.013,  0.006,
        0.033,  0.037,  0.046,  0.084,  0.221,  0.245,  0.296)
D <- c(D,rep(.2960001,40-28)) # 28 values, but Hosking mentions
                              # 40 values in total
z <-  pwmRC(D,threshold=.2960001)
str(z)
# Hosking reports B-type L-moments for this sample are
# lamB1 = -.516 and lamB2 = 0.523
btypelmoms <- pwm2lmom(z$Bbetas)
# My version of R reports lamB1 = -0.5162 and lamB2 = 0.5218
str(btypelmoms)
rg.pars <- parrevgum(btypelmoms,z$zeta)
str(rg.pars)
# Hosking reports xi = 0.1636 and alpha = 0.9252 for the sample
# My version of R reports xi = 0.1635 and alpha = 0.9254
```

---

parrice *Estimate the Parameters of the Rice Distribution*

Description

This function estimates the parameters (ν and α) of the Rice distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are complex and tabular lookup is made using a relation between τ and a form of signal-to-noise ratio SNR defined as ν/α and a relation between τ and precomputed Laguerre polynomial (`LaguerreHalf`).

The λ_1 (mean) is most straightforward

$$\lambda_1 = \alpha \times \sqrt{\pi/2} \times L_{1/2}(-\nu^2/[2\alpha^2]),$$

for which the terms to the right of the multiplication symbol are uniquely a function of τ and precomputed for tabular lookup and interpolation from ‘`sysdata.rdb`’ (`.lmomcohash$RiceTable`). Parameter estimation also relies directly on tabular lookup and interpolation to convert τ to SNR. The file ‘`SysDataBuilder01.R`’ provides additional technical details.

Usage

```
parrice(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.

checklmom — Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check. However, the end point of the Rice distribution for high ν/α is not determined here, so it is recommended to leave `checklmom` turned on.

... — Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `rice`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "parrice".
`ifail`	A numeric failure mode.
`ifailtext`	A helpful message on the failure.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`lmomrice`, `cdfrice`, `pdfrice`, `quarice`

Examples

```
## Not run:
  parrice(lmomrice(vec2par(c(10,50),   type="rice"))) # Within Rician limits
  parrice(lmomrice(vec2par(c(100,0.1), type="rice"))) # Beyond Rician limits

plotlmrdia(lmrdia(), xlim=c(0,0.2), ylim=c(-0.1,0.22),
           autolegend=TRUE, xleg=0.05, yleg=0.05)
lines(.lmomcohash$RiceTable$TAU3, .lmomcohash$RiceTable$TAU4, lwd=5, col=8)
legend(0.1,0, "RICE DISTRIBUTION", lwd=5, col=8, bty="n")
text(0.14, -0.04,  "Normal distribution limit on left end point"   )
text(0.14, -0.055, "Rayleigh distribution limit on right end point")

# check parrice against a Maximum Likelihood method in VGAM
set.seed(1)
library(VGAM) # now example from riceff() of VGAM
vee <- exp(2); sigma <- exp(1); y <- rrice(n <- 1000, vee, sigma)
fit <- vglm(y ~ 1, riceff, trace=TRUE, crit="c")
Coef(fit)
```

```
# NOW THE MOMENT OF TRUTH, USING L-MOMENTS
parrice(lmoms(y))
# VGAM package 0.8-1 reports
#     vee    sigma
# 7.344560 2.805877
# lmomco 1.2.2 reports
#      nu    alpha
# 7.348784 2.797651
## End(Not run)
```

---

pars2x *Estimate Quantiles from an Ensemble of Parameters*

Description

This function acts as a frontend to estimate quantiles from an ensemble of parameters from the methods of L-moments (lmr2par), maximum likelihood (MLE, mle2par), and maximum product of spacings (MPS, mps2par) for nonexceedance probabilities. The mean, standard deviation, and number of unique quantiles for each nonexceedance probability are computed too. The unique quantiles are used because the MLE and MPS methods could fall back to L-moments or other and thus it should be considered that one of the methods might have failed.

Usage

```
pars2x(f, paras, na.rm=FALSE, ...)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
paras	An ensemble of parameters from x2pars.
na.rm	A logical to pass to the mean and standard deviation computations.
...	The additional arguments, if ever used.

Value

A data.frame having, if at least one of the parameter estimation methods is not NULL, the following columns in addition to attributes that are demonstrated in the **Examples** section:

lmr	Quantiles based on parameters from method of L-moments.
mle	Quantiles based on parameters from MLE.
mps	Quantiles based on parameters from MPS.
f	The nonexceedance probabilities.
mean	The mean of the unique quantiles (usually three) seen for each probability. Results can be affected by na.rm.
sd	The standard deviation of the unique quantiles (usually three) seen for each probability. Results can be affected by na.rm.
n	The number of unique quantiles (usually three) seen for each probability and quantiles computed as NA are not counted.

Author(s)

W.H. Asquith

See Also

x2pars

Examples

```
## Not run:
# Simulate from GLO and refit it. Occasionally, the simulated data
# will result in MLE or MPS failing to converge, just a note to users.
# This example also shows the use of the attributes of the Results.
set.seed(3237)
x <- rlmomco(32, vec2par(c(2.5, 0.7, -0.39), type="glo"))
three.para.est <- x2pars(x, type="glo")
FF <- nonexceeds() # a range in nonexceedance probabilities
# In the event of MLE or MPS failure, one will see NA's in the Results.
Results <- pars2x(FF, three.para.est, na.rm=FALSE)
sum <- attr(Results, "all.summary")
plot(pp(x), sort(x), type="n", ylim=range(sum), log="y")
polygon(attr(Results, "f.poly"), attr(Results, "x.poly"), col=8, lty=0)
points(pp(x), sort(x), col=3)
lines(Results$f, Results$lmr,  col=1) # black line
lines(Results$f, Results$mle,  col=2) # red   line
lines(Results$f, Results$mps,  col=4) # blue  line
lines(Results$f, Results$mean, col=6, lty=2, lwd=2) # purple mean #
## End(Not run)
```

---

parsla *Estimate the Parameters of the Slash Distribution*

Description

This function estimates the parameters of the Slash distribution from the trimmed L-moments (TL-moments) having trim level 1. The relations between distribution parameters and TL-moments are shown under lmomsla.

Usage

```
parsla(lmom, ...)
```

Arguments

lmom	A TL-moment object from TLmoms with trim=1.
...	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `sla`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “parsla”.

Author(s)

W.H. Asquith

References

Rogers, W.H., and Tukey, J.W., 1972, Understanding some long-tailed symmetrical distributions: Statistica Neerlandica, v. 26, no. 3, pp. 211–226.

See Also

`TLmoms`, `lmomsla`, `cdfsla`, `pdfsla`, `quasla`

Examples

```
## Not run:
par1 <- vec2par(c(-100, 30), type="sla")
X   <- rlmomco(500, par1)
lmr <- TLmoms(X, trim=1)
par2 <- parsla(lmr)
F <- seq(0.001,.999, by=0.001)
plot(qnorm(pp(X)), sort(X), pch=21, col=8,
     xlab="STANDARD NORMAL VARIATE",
     ylab="QUANTILE")
lines(qnorm(F), quasla(F, par1), lwd=3)
lines(qnorm(F), quasla(F, par2), col=2)

## End(Not run)
```

`parsmd` *Estimate the Parameters of the Singh–Maddala Distribution*

Description

This function estimates the parameters of the Singh–Maddala (Burr Type XII) distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The L-moments in terms of the parameters are complicated and solved numerically. Extensive study of the computational limits of the R implementation are incorporated within the source code of the function. The file `lmomco/inst/doc/domain_of_smd.R` contains the algorithmic sweep used to compute the L-skew and L-kurtosis attainable domain of the distribution.

Usage

```
parsmd(lmom, checklmom=TRUE, checkbounds=TRUE, snap.tau4=TRUE, ...)
```

Arguments

lmom	An L-moment object created by lmoms or vec2lmom.
checklmom	Should the lmom be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_3 and τ_4 inequality, are.lmom.valid). However, for some circumstances or large simulation exercises then one might want to bypass this check.
checkbounds	Should the lower bounds of τ_4 be verified and if sample $\hat{\tau}_3$ and $\hat{\tau}_4$ are outside of these bounds, then NA are returned for the solutions.
snap.tau4	A logical to trigger the application of the empirical limits of the distribution in terms of τ_3 and τ_4 wherein parameter estimation appears numerically possible and such parameters return the given values of these L-moment ratios. The lower and upper limits of τ_4 are defined by separate polynomials as functions of τ_3. If the logical is true, then τ_4 in excess of the upper bounds are assigned to the upper bounds and τ_4 in deficit of the lower bounds are assigned to the lower bounds. Messages within the returned parameter object are provided if the snapping occurs.
...	Other arguments to pass.

Value

An R list is returned.

type	The type of distribution: smd.
para	The parameters of the distribution.
last_para	The last or final iteration of the parameters that are the same as para if ifail is zero. This provides a way to preserve where the parameter left off or gave up.
source	The source of the parameters: "parsmd".
iter	The number of iteration attempts looping on the optim() call.
rt	The output of the optim() call.
message	A message from parsmd, which generally involves checkbounds=TRUE and snap.tau4=TRUE on the resetting or snapping of the τ_3 and τ_4 to the computational bounds for the distribution.
ifail	A interger flag to status of the operations: -1 means that the L-moments are invalid (if they are checked), 0 means that the parameter estimation appears successful, and 1 means that the parameter estimation appears to have failed.

Author(s)

W.H. Asquith

References

Shahzad, M.N., and Zahid, A., 2013, Parameter estimation of Singh Maddala distribution by moments: International Journal of Advanced Statistics and Probability, v. 1, no. 3, pp. 121–131, doi:10.14419/ijasp.v1i3.1206.

See Also

`lmomsmd`, `cdfsmd`, `pdfsmd`, `quasmd`

Examples

```
lmr <- lmoms(rnorm(20))
parsmd(lmr, snap.tau4=TRUE)
```

`parst3` *Estimate the Parameters of the 3-Parameter Student t Distribution*

Description

This function estimates the parameters of the 3-parameter Student t distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomst3`. The largest value of ν recognized is $10^5.5$, which is the near the Normal distribution and the smallest value recognized is 1.001, which is near the Cauchy. As $\nu \rightarrow 1$ the distribution limits to the Cauchy, but the implementation here does not switch over to the Cauchy. Therefore in **lmomco** $1.001 \leq \nu \leq 10^5.5$. The ν is the "degrees of freedom" parameter that is well-known with the 1-parameter Student t distribution. The nu limits are studied in the `inst/doc/t4t6/studyST3.R` script and the `theoTLmoms` function and its performance on the quantile function of the distribution provide the guidance including range of numerically computable τ_6. The τ_4 value can be set as low as 0.1226 as short-hand for the true lower L-kurtosis limit, which is that of the Normal ($30/\pi \times \mathrm{atan}(\sqrt{2}) - 9 = 0.1226017$ and additional decimals). Internally, the given $0.1226 \leq \tau_4 \leq 0.1226017$ is snapped to that of the Normal with an internal small positive nudge up. The $\tau_4 > 0.998$ are set to $\tau_4 = 0.998$.

Usage

```
parst3(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `st3`.
`para`	The parameters of the distribution.
`rt`	The returned list of the `uniroot()` call to estimate ν.
`source`	The source of the parameters: "parst3".

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`lmomst3`, `cdfst3`, `pdfst3`, `quast3`

Examples

```
parst3(vec2lmom(c(10, 2, 0, 0.1226)))$para
parst3(vec2lmom(c(10, 2, 0, 0.14  )))$para
parst3(vec2lmom(c(10, 2, 0, 0.4   )))$para
parst3(vec2lmom(c(10, 2, 0, 0.9   )))$para
parst3(vec2lmom(c(10, 2, 0, 0.998 )))$para
```

`partexp` *Estimate the Parameters of the Truncated Exponential Distribution*

Description

This function estimates the parameters of the Truncated Exponential distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The parameter ψ is the right truncation of the distribution, and α is a scale parameter, letting $\beta = 1/\alpha$ to match nomenclature of Vogel and others (2008), and recalling the L-moments in terms of the parameters and letting $\eta = \exp(-\beta\psi)$ are

$$\lambda_1 = \frac{1 - \eta + \eta\log(\eta)}{\beta(1-\eta)},$$

$$\lambda_2 = \frac{1 + 2\eta\log(\eta) - \eta^2}{2\beta(1-\eta)^2}, \text{ and}$$

$$\tau_2 = \frac{\lambda_2}{\lambda_1} = \frac{1 + 2\eta\log(\eta) - \eta^2}{2(1-\eta)[1 - \eta + \eta\log(\eta)]},$$

and τ_2 is a monotonic function of η is decreasing from $\tau_2 = 1/2$ at $\eta = 0$ to $\tau_2 = 1/3$ at $\eta = 1$ the parameters are readily solved given $\tau_2 = [1/3, 1/2]$, the R function `uniroot` can be used to solve for η with a starting interval of $(0, 1)$, then the parameters in terms of the parameters are

$$\alpha = \frac{1 - \eta + \eta \log(\eta)}{(1 - \eta)\lambda_1}, \text{ and}$$

$$\psi = -\log(\eta)/\alpha.$$

If the η is rooted as equaling zero, then it is assumed that $\hat{\tau}_2 \equiv \tau_2$ and the exponential distribution triggered, or if the η is rooted as equaling unity, then it is assumed that $\hat{\tau}_2 \equiv \tau_2$ and the uniform distribution triggered (see below).

The distribution is restricted to a narrow range of L-CV ($\tau_2 = \lambda_2/\lambda_1$). If $\tau_2 = 1/3$, the process represented is a stationary Poisson for which the probability density function is simply the uniform distribution and $f(x) = 1/\psi$. If $\tau_2 = 1/2$, then the distribution is represented as the usual exponential distribution with a location parameter of zero and a scale parameter $1/\beta$. Both of these limiting conditions are supported.

If the distribution shows to be uniform ($\tau_2 = 1/3$), then the third element in the returned parameter vector is used as the ψ parameter for the uniform distribution, and the first and second elements are `NA` of the returned parameter vector.

If the distribution shows to be exponential ($\tau_2 = 1/2$), then the second element in the returned parameter vector is the inverse of the rate parameter for the exponential distribution, and the first element is `NA` and the third element is `0` (a numeric `FALSE`) of the returned parameter vector.

Usage

```
partexp(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `texp`.
`para`	The parameters of the distribution.
`ifail`	A logical value expressed in numeric form indicating the failure or success state of the parameter estimation. A value of two indicates that the $\tau_2 < 1/3$ whereas a value of three indicates that the $\tau_2 > 1/2$; for each of these inequalities a fuzzy tolerance of one part in one million is used. Successful parameter estimation,

which includes the uniform and exponential boundaries, is indicated by a value of zero.

`ifail.message` Various messages for successful and failed parameter estimations are reported. In particular, there are two conditions for which each distributional boundary (uniform or exponential) can be obtained. First, for the uniform distribution, one message would indicate if the $\tau_2 = 1/3$ is assumed within a one part in one million will be identified or if η is rooted to 1. Second, for the exponential distribution, one message would indicate if the $\tau_2 = 1/2$ is assumed within a one part in one million will be identified or if η is rooted to 0.

`eta` The value for η. The value is set to either unity or zero if the τ_2 fuzzy tests as being equal to 1/3 or 1/2, respectively. The value is set to the rooted value of η for all other valid solutions. The value is set to `NA` if τ_2 tests as being outside the 1/3 and 1/2 limits.

`source` The source of the parameters: "partexp".

Author(s)

W.H. Asquith

References

Vogel, R.M., Hosking, J.R.M., Elphick, C.S., Roberts, D.L., and Reed, J.M., 2008, Goodness of fit of probability distributions for sightings as species approach extinction: Bulletin of Mathematical Biology, DOI 10.1007/s11538-008-9377-3, 19 p.

See Also

`lmomtexp`, `cdftexp`, `pdftexp`, `quatexp`

Examples

```
# truncated exponential is a nonstationary poisson process
A  <- partexp(vec2lmom(c(100, 1/2),   lscale=FALSE)) # pure exponential
B  <- partexp(vec2lmom(c(100, 0.499), lscale=FALSE)) # almost exponential
BB <- partexp(vec2lmom(c(100, 0.45),  lscale=FALSE)) # truncated exponential
C  <- partexp(vec2lmom(c(100, 1/3),   lscale=FALSE)) # stationary poisson process
D  <- partexp(vec2lmom(c(100, 40))) # truncated exponential
```

`parTLgld` *Estimate the Parameters of the Generalized Lambda Distribution using Trimmed L-moments (t=1)*

Description

This function estimates the parameters of the Generalized Lambda distribution given the trimmed L-moments (TL-moments) for $t = 1$ of the data in a TL-moment object with a trim level of unity (`trim=1`). The relations between distribution parameters and TL-moments are seen under `lmomTLgld`. There are no simple expressions for the parameters in terms of the L-moments. Consider that multiple parameter solutions are possible with the Generalized Lambda distribution so some expertise with this distribution and other aspects is advised.

Usage

```
parTLgld(lmom, verbose=FALSE, initkh=NULL, eps=1e-3,
         aux=c("tau5", "tau6"), checklmom=TRUE, ...)
```

Arguments

`lmom`	A TL-moment object created by `TLmoms`.
`verbose`	A logical switch on the verbosity of output. Default is `verbose=FALSE`.
`initkh`	A vector of the initial guess of the κ and h parameters. No other regions of parameter space are consulted.
`eps`	A small term or threshold for which the square root of the sum of square errors in τ_3 and τ_4 is compared to to judge "good enough" for the alogrithm to order solutions based on smallest error as explained in next argument.
`aux`	Control the algorithm to order solutions based on smallest error in trimmed $\Delta\tau_5$ or $\Delta\tau_6$.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Details

Karian and Dudewicz (2000) summarize six regions of the κ and h space in which the Generalized Lambda distribution is valid for suitably choosen α. Numerical experimentation suggestions that the L-moments are not valid in Regions 1 and 2. However, initial guesses of the parameters within each region are used with numerous separate `optim` (the R function) efforts to perform a least sum-of-square errors on the following objective function.

$$(\hat{\tau}^{(1)}_3 - \tilde{\tau}^{(1)}_3)^2 + (\hat{\tau}^{(1)}_4 - \tilde{\tau}^{(1)}_4)^2,$$

where $\tilde{\tau}^{(1)}_r$ is the L-moment ratio of the data, $\hat{\tau}^{(1)}_r$ is the estimated value of the TL-moment ratio for the current pairing of κ and h and $\tau^{(1)}_r$ is the actual value of the L-moment ratio.

For each optimization a check on the validity of the parameters so produced is made–are the parameters consistent with the Generalized Lambda distribution and a second check is made on the validity of $\tau^{(1)}_3$ and $\tau^{(1)}_4$. If both validity checks return `TRUE` then the optimization is retained if its sum-of-square error is less than the previous optimum value. It is possible for a given solution to be found outside the starting region of the initial guesses. The surface generated by the $\tau_3^{(1)}$ and $\tau_4^{(1)}$ equations seen in lmomTLgld is complex; different initial guesses within a given region can yield what appear to be radically different κ and h. Users are encouraged to "play" with alternative solutions (see the verbose argument). A quick double check on the L-moments (not TL-moments) from the solved parameters using lmomTLgld is encouraged as well.

Value

An R list is returned if result='best'.

type	The type of distribution: gld.
para	The parameters of the distribution.
delTau5	Difference between $\tilde{\tau}_5^{(1)}$ of the fitted distribution and true $\hat{\tau}_5^{(1)}$.
error	Smallest sum of square error found.
source	The source of the parameters: "parTLgld".
rest	An R data.frame of other solutions if found.

The rest of the solutions have the following:

xi	The location parameter of the distribution.
alpha	The scale parameter of the distribution.
kappa	The 1st shape parameter of the distribution.
h	The 2nd shape parameter of the distribution.
attempt	The attempt number that found valid TL-moments and parameters of GLD.
delTau5	The absolute difference between $\hat{\tau}_5^{(1)}$ of data to $\tilde{\tau}_5^{(1)}$ of the fitted distribution.
error	The sum of square error found.
initial_k	The starting point of the κ parameter.
initial_h	The starting point of the h parameter.
valid.gld	Logical on validity of the GLD—TRUE by this point.
valid.lmr	Logical on validity of the L-moments—TRUE by this point.
lowerror	Logical on whether error was less than eps—TRUE by this point.

Note

This function is a cumbersome method of parameter solution, but years of testing suggest that with supervision and the available options regarding the optimization that reliable parameter estimations result.

Author(s)

W.H. Asquith

Source

W.H. Asquith in Feb. 2006 with a copy of Karian and Dudewicz (2000) and again Feb. 2011.

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

TLmoms, lmomTLgld, cdfgld, pdfgld, quagld, pargld

Examples

```
# As of version 1.6.2, it is felt that in spirit of CRAN CPU
# reduction that the intensive operations of parTLgld() should
# be kept a bay.

## Not run:
X <- rgamma(202,2) # simulate a skewed distribution
lmr <- TLmoms(X, trim=1) # compute trimmed L-moments
PARgldTL <- parTLgld(lmr) # fit the GLD

F <- pp(X) # plotting positions for graphing
plot(F,sort(X), col=8, cex=0.25)
lines(F, qlmomco(F,PARgldTL)) # show the best estimate
if(! is.null(PARgldTL$rest)) {
  n <- length(PARgldTL$rest$xi)
  other <- unlist(PARgldTL$rest[n,1:4]) # show alternative
  lines(F, qlmomco(F,vec2par(other, type="gld")), col=2)
}
# Note in the extraction of other solutions that no testing for whether
# additional solutions were found is made. Also, it is quite possible
# that the other solutions "[n,1:4]" is effectively another numerical
# convergence on the primary solution. Some users of this example thus
# might not see two separate lines. Users are encouraged to inspect the
# rest of the solutions: print(PARgld$rest)

# For one run of the above example, the GLD results follow
#print(PARgldTL)
#$type
#[1] "gld"
#$para
#         xi       alpha       kappa           h
# 1.02333964 -3.86037875 -0.06696388 -0.22100601
#$delTau5
#[1] -0.02299319
#$error
#[1] 7.048409e-08
#$source
#[1] "pargld"
#$rest
#         xi      alpha       kappa          h attempt     delTau5        error
```

```
#1  1.020725 -3.897500 -0.06606563 -0.2195527       6 -0.02302222 1.333402e-08
#2  1.021203 -3.895334 -0.06616654 -0.2196020       4 -0.02304333 8.663930e-11
#3  1.020684 -3.904782 -0.06596204 -0.2192197       5 -0.02306065 3.908918e-09
#4  1.019795 -3.917609 -0.06565792 -0.2187232       2 -0.02307092 2.968498e-08
#5  1.023654 -3.883944 -0.06668986 -0.2198679       7 -0.02315035 2.991811e-07
#6 -4.707935 -5.044057  5.89280906 -0.3261837      13  0.04168800 2.229672e-10

## End(Not run)

## Not run:
F <- seq(.01,.99,.01)
plot(F,qlmomco(F, vec2par(c( 1.02333964, -3.86037875,
                            -0.06696388, -0.22100601), type="gld")),
     type="l")
lines(F,qlmomco(F, vec2par(c(-4.707935, -5.044057,
                              5.89280906, -0.3261837), type="gld")))

## End(Not run)
```

---

parTLgpa — *Estimate the Parameters of the Generalized Pareto Distribution using Trimmed L-moments*

Description

This function estimates the parameters of the Generalized Pareto distribution given the the trimmed L-moments (TL-moments) for $t = 1$ of the data in TL-moment object with a trim level of unity (`trim=1`). The parameters are computed as

$$\kappa = \frac{10 - 45\tau_3^{(1)}}{9\tau_3^{(1)} + 10},$$

$$\alpha = \frac{1}{6}\lambda_2^{(1)}(\kappa+2)(\kappa+3)(\kappa+4), \text{ and}$$

$$\xi = \lambda_1^{(1)} - \frac{\alpha(\kappa+5)}{(\kappa+2)(\kappa+3)}.$$

Usage

```
parTLgpa(lmom, ...)
```

Arguments

`lmom`	A TL-moment object created by `TLmoms`.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: gpa.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “parTLgpa”.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

`TLmoms`, `lmomTLgpa`, `cdfgpa`, `pdfgpa`, `quagpa`

Examples

```
TL <- TLmoms(rnorm(20),trim=1)
parTLgpa(TL)
```

`partri` *Estimate the Parameters of the Asymmetric Triangular Distribution*

Description

This function estimates the parameters of the Asymmetric Triangular distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomtri`.

The estimtion by the `partri` function is built around simultaneous numerical optimization of an objective function defined as

$$\epsilon = \left(\frac{\lambda_1 - \hat{\lambda}_1}{\hat{\lambda}_1}\right)^2 + \left(\frac{\lambda_2 - \hat{\lambda}_2}{\hat{\lambda}_2}\right)^2 + \left(\frac{\tau_3 - \hat{\tau}_3}{1}\right)^2$$

for estimation of the three parameters (ν, minimum; ω, mode; and ψ, maximum) from the sample L-moments ($\hat{\lambda}_1$, $\hat{\lambda}_2$, $\hat{\tau}_3$). The divisions shown in the objective function are used for scale removal to help make each L-moment order somewhat similar in its relative contribution to the solution. The coefficient of L-variation is not used because the distribution implementation by the **lmomco** package supports entire real number line and the loss of definition of τ_2 at $x = 0$, in particular, causes untidiness in coding.

The function is designed to support both left- or right-hand right triangular shapes because of (1) `paracheck` argument availability in `lmomtri`, (2) the sorting of the numerical estimates if the mode is no compatable with either of the limits, and (3) the snapping of $\nu = \omega \equiv (\nu^\star + \omega^\star)/2$ when $\hat{\tau}_3 > 0.142857$ or $\psi = \omega \equiv (\psi^\star + \omega^\star)/2$ when $\hat{\tau}_3 < 0.142857$ where the $\star$ versions are the optimized values if the τ_3 is very near to its numerical bounds.

Usage

```
partri(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `tri`.
`para`	The parameters of the distribution.
`obj.val`	The value of the objective function, which is the error of the optimization.
`source`	The source of the parameters: "partri".

Author(s)

W.H. Asquith

See Also

`lmomtri`, `cdftri`, `pdftri`, `quatri`

Examples

```
lmr <- lmomtri(vec2par(c(10,90,100), type="tri"))
partri(lmr)

partri(lmomtri(vec2par(c(-11, 67,67), type="tri")))$para
partri(lmomtri(vec2par(c(-11,-11,67), type="tri")))$para
```

parwak — *Estimate the Parameters of the Wakeby Distribution*

Description

This function estimates the parameters of the Wakeby distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The relations between distribution parameters and L-moments are seen under `lmomwak`.

Usage

```
parwak(lmom, checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `wak`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: "parwak".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`lmomwak`, `cdfwak`, `pdfwak`, `quawak`

Examples

```
lmr <- lmoms(rnorm(20))
parwak(lmr)
```

`parwei` *Estimate the Parameters of the Weibull Distribution*

Description

This function estimates the parameters of the Weibull distribution given the L-moments of the data in an L-moment object such as that returned by `lmoms`. The Weibull distribution is a reverse Generalized Extreme Value distribution. As result, the Generalized Extreme Value algorithms are used for implementation of the Weibull in this package. The relations between the Generalized Extreme Value parameters (ξ, α, and κ) and the Weibull parameters are

$$\kappa = 1/\delta,$$

$$\alpha = \beta/\delta, \text{ and}$$

$$\xi = \zeta - \beta.$$

These relations are taken from Hosking and Wallis (1997). The relations between the distribution parameters and L-moments are seen under `lmomgev`.

Usage

```
parwei(lmom,checklmom=TRUE, ...)
```

Arguments

`lmom`	An L-moment object created by `lmoms` or `vec2lmom`.
`checklmom`	Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.
`...`	Other arguments to pass.

Value

An R `list` is returned.

`type`	The type of distribution: `wei`.
`para`	The parameters of the distribution.
`source`	The source of the parameters: “parwei”.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

lmomwei, cdfwei, pdfwei, quawei

Examples

```
parwei(lmoms(rnorm(20)))
## Not run:
str(parwei(lmoms(rweibull(3000,1.3, scale=340)-1200))) #
## End(Not run)
```

---

pdfaep4 *Probability Density Function of the 4-Parameter Asymmetric Exponential Power Distribution*

Description

This function computes the probability density of the 4-parameter Asymmetric Exponential Power distribution given parameters (ξ, α, κ, and h) computed by paraep4. The probability density function is

$$f(x) = \frac{\kappa\, h}{\alpha(1+\kappa^2)\,\Gamma(1/h)} \exp\left(- \left[\kappa^{\mathrm{sign}(x-\xi)} \left(\frac{|x-\xi|}{\alpha} \right) \right]^h \right)$$

where $f(x)$ is the probability density for quantile x, ξ is a location parameter, α is a scale parameter, κ is a shape parameter, and h is another shape parameter. The range is $-\infty < x < \infty$. If the τ_3 of the distribution is zero (symmetrical), then the distribution is known as the Exponential Power (see lmrdia46).

Usage

```
pdfaep4(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from paraep4 or vec2par.
paracheck	A logical controlling whether the parameters and checked for validity.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Delicado, P., and Goria, M.N., 2008, A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution: Computational Statistics and Data Analysis, v. 52, no. 3, pp. 1661–1673.

See Also

cdfaep4, quaaep4, lmomaep4, paraep4

Examples

```
aep4 <- vec2par(c(1000,15000,0.5,0.4), type='aep4');
F <- nonexceeds();
x <- quaaep4(F,aep4);
check.pdf(pdfaep4,aep4,plot=TRUE);
## Not run:
delx <- .01;
x <- seq(-10,10, by=delx);
K <- 3;
PAR <- list(para=c(0,1, K, 0.5), type="aep4");
plot(x,pdfaep4(x, PAR), type="n",
     ylab="PROBABILITY DENSITY",
     ylim=c(0,0.6), xlim=range(x));
lines(x,pdfaep4(x,PAR), lwd=2);

PAR <- list(para=c(0,1, K, 1), type="aep4");
lines(x,pdfaep4(x, PAR), lty=2, lwd=2);

PAR <- list(para=c(0,1, K, 2), type="aep4");
lines(x,pdfaep4(x, PAR), lty=3, lwd=2);

PAR <- list(para=c(0,1, K, 4), type="aep4");
lines(x,pdfaep4(x, PAR), lty=4, lwd=2);

## End(Not run)
```

pdfcau *Probability Density Function of the Cauchy Distribution*

Description

This function computes the probability density of the Cauchy distribution given parameters (ξ and α) provided by parcau. The probability density function is

$$f(x) = \left(\pi\alpha\left[1 + \left(\frac{x-\xi}{\alpha}\right)^2\right]\right)^{-1},$$

where $f(x)$ is the probability density for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
pdfcau(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parcau or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, vol. 43, pp. 299–314.

Evans, Merran, Hastings, Nicholas, Peacock, J.B., 2000, Statistical distributions: 3rd ed., Wiley, New York.

Gilchrist, W.G., 2000, Statistical modeling with quantile functions: Chapman and Hall/CRC, Boca Raton, FL.

See Also

cdfcau, quacau, lmomcau, parcau, vec2par

Examples

```
cau <- vec2par(c(12,12),type='cau')
x <- quacau(0.5,cau)
pdfcau(x,cau)
```

`pdfemu` *Probability Density Function of the Eta-Mu Distribution*

Description

This function computes the probability density of the Eta-Mu ($\eta : \mu$) distribution given parameters (η and μ) computed by `paremu`. The probability density function is

$$f(x) = \frac{4\sqrt{\pi}\mu^{\mu-1/2}h^{\mu}}{\gamma(\mu)H^{\mu-1/2}}\, x^{2\mu}\, \exp(-2\mu h x^2)\, I_{\mu-1/2}(2\mu H x^2),$$

where $f(x)$ is the nonexceedance probability for quantile x, and the modified Bessel function of the first kind is $I_k(x)$, and the h and H are

$$h = \frac{1}{1-\eta^2},$$

and

$$H = \frac{\eta}{1-\eta^2},$$

for "Format 2" as described by Yacoub (2007). This format is exclusively used in the algorithms of the **lmomco** package.

If $\mu = 1$, then the Rice distribution results, although `pdfrice` is not used. If $\kappa \rightarrow 0$, then the exact Nakagami-m density function results with a close relation to the Rayleigh distribution.

Define m as

$$m = 2\mu\left[1 + \left(\frac{H}{h}\right)^2\right],$$

where for a given m, the parameter μ must lie in the range

$$m/2 \le \mu \le m.$$

The $I_k(x)$ for real $x > 0$ and noninteger k is

$$I_k(x) = \frac{1}{\pi}\int_0^{\pi} \exp(x\cos(\theta))\cos(k\theta)\; \mathrm{d}\theta - \frac{\sin(k\pi)}{\pi}\int_0^{\infty} \exp(-x\cosh(t) - kt)\; \mathrm{d}t.$$

Usage

```
pdfemu(x, para, paracheck=TRUE)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `paremu` or `vec2par`.
`paracheck`	A logical controlling whether the parameters and checked for validity.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

cdfemu, quaemu, lmomemu, paremu

Examples

```
## Not run:
x <- seq(0,4, by=.1)
para <- vec2par(c(.5, 1.4), type="emu")
F <- cdfemu(x, para);         X <- quaemu(F, para)
plot(F, X, type="l", lwd=8);  lines(F, x, col=2)

delx <- 0.005
x <- seq(0,3, by=delx)
plot(c(0,3), c(0,1), xaxs="i", yaxs="i",
     xlab="RHO", ylab="pdfemu(RHO)", type="n")
mu <- 0.6
# Note that in order to produce the figure correctly using the etas
# shown in the figure that it must be recognized that these are the etas
# for format1, but all of the algorithms in lmomco are built around
# format2
etas.format1 <- c(0, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 1)
etas.format2 <- (1 - etas.format1)/(1+etas.format1)
H <- etas.format2 / (1 - etas.format2^2)
h <-            1 / (1 - etas.format2^2)
for(eta in etas.format2) {
   lines(x, pdfemu(x, vec2par(c(eta, mu), type="emu")),
         col=rgb(eta^2,0,0))
}
mtext("Yacoub (2007, figure 5)")

plot(c(0,3), c(0,2), xaxs="i", yaxs="i",
     xlab="RHO", ylab="pdfemu(RHO)", type="n")
eta.format1 <- 0.5
eta.format2 <- (1 - eta.format1)/(1 + eta.format1)
mus <- c(0.25, 0.3, 0.5, 0.75, 1, 1.5, 2, 3)
for(mu in mus) {
   lines(x, pdfemu(x, vec2par(c(eta, mu), type="emu")))
}
mtext("Yacoub (2007, figure 6)")
```

```
plot(c(0,3), c(0,1), xaxs="i", yaxs="i",
     xlab="RHO", ylab="pdfemu(RHO)", type="n")
m <- 0.75
mus <- c(0.7425, 0.75, 0.7125, 0.675, 0.45, 0.5, 0.6)
for(mu in mus) {
   eta <- sqrt((m / (2*mu))^-1 - 1)
   print(eta)
   lines(x, pdfemu(x, vec2par(c(eta, mu), type="emu")))
}
mtext("Yacoub (2007, figure 7)") #
## End(Not run)
```

---

pdfexp — *Probability Density Function of the Exponential Distribution*

Description

This function computes the probability density of the Exponential distribution given parameters (ξ and α) computed by parexp. The probability density function is

$$f(x) = \alpha^{-1}\exp(Y),$$

where Y is

$$Y = \left(\frac{-(x-\xi)}{\alpha}\right),$$

where $f(x)$ is the probability density for the quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
pdfexp(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parexp or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfexp, quaexp, lmomexp, parexp

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
expp <- parexp(lmr)
x <- quaexp(.5,expp)
pdfexp(x,expp)
```

---

pdfgam — *Probability Density Function of the Gamma Distribution*

Description

This function computes the probability density function of the Gamma distribution given parameters (α, shape, and β, scale) computed by pargam. The probability density function has no explicit form, but is expressed as an integral

$$f(x|\alpha,\beta)^{\mathrm{lmomco}} = \frac{1}{\beta^\alpha\,\Gamma(\alpha)}\,x^{\alpha-1}\exp(-x/\beta),$$

where $f(x)$ is the probability density for the quantile x, α is a shape parameter, and β is a scale parameter.

Alternatively, a three-parameter version is available for this package following the parameterization of the Generalized Gamma distribution used in the **gamlss.dist** package and is

$$f(x|\mu,\sigma,\nu)^{\mathrm{lmomco}}_{\mathrm{gamlss.dist}} = \frac{\theta^\theta\,|\nu|}{\Gamma(\theta)}\,\frac{z^\theta}{x}\exp(-z\theta),$$

where $z = (x/\mu)^\nu$, $\theta = 1/(\sigma^2\,|\nu|^2)$ for $x > 0$, location parameter $\mu > 0$, scale parameter $\sigma > 0$, and shape parameter $-\infty < \nu < \infty$. Note that for $\nu = 0$ the distribution is log-Normal. The three parameter version is automatically triggered if the length of the `para` element is three and not two.

Usage

```
pdfgam(x, para)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `pargam` or `vec2par`.

Value

Probability density (f) for x.

Note

Two Parameter $\equiv$ Three Parameter
For $\nu = 1$, the parameter conversion between the two gamma forms is $\alpha = \sigma^{-2}$ and $\beta = \mu\sigma^2$ and this can be readily verified:

```
mu <- 5; sig <- 0.7; nu <- 0
x <- exp(seq(-3,3,by=.1))
para2 <- vec2par(c(1/sig^2, (mu*sig^2)  ), type="gam")
para3 <- vec2par(c(      mu,    sig,   1), type="gam")
plot(x, pdfgam(x, para2), ylab="Gamma Density"); lines(x, pdfgam(x, para3))
```

Package flexsurv Generalized Gamma
The **flexsurv** package provides an "original" (`GenGamma.orig`) and "preferred" parameterization (`GenGamma`) of the Generalized Gamma distribution and discusses parameter conversion between the two. Here the parameterization of the preferred form is compared to that in **lmomco**. The probability density function of `dgengamma()` from **flexsurv** is

$$f(x|\mu_2, \sigma_2, Q)_{\mathrm{flexsurv}} = \frac{\eta^\eta |Q|}{\sigma_2 \Gamma(\eta)} \frac{1}{x} \exp\{\eta \times [wQ - \exp(wQ)]\},$$

where $\eta = Q^{-2}$, $w = \log(g/\eta)/Q$ for $g \sim \mathrm{Gamma}(\eta, 1)$ where Gamma is the cumulative distribution function (presumably, need to verify this) of the Gamma distribution, and

$$x \sim \exp(\mu_2 + w\sigma_2),$$

where $\mu_2 > 0$, $\sigma_2 > 0$, and $-\infty < Q < \infty$, and the log-Normal distribution results for $Q = 0$. These definitions for **flexsurv** seem incomplete to this author and further auditing is needed.

Additional Generalized Gamma Comparison
The default **gamlss.dist** package version uses so-called *log.links* for μ and σ, and so-called *identity.link* for ν and these links are implicit for **lmomco**. The parameters can be converted to **flexsurv** package equivalents by $\mu_2 = \log(\mu)$, $\sigma_2 = \sigma$, and $Q = \sigma\nu$, which is readily verified by

```
mu <- 2; sig <- 0.8; nu <- 0.2; x <- exp(seq(-3,1,by=0.1))
para <- vec2par(c(mu,sig,nu), type="gam")
dGG <- gamlss.dist::dGG(x, mu=mu, sigma=sig, nu=nu)
plot( x, dGG, ylab="density", lwd=0.8, cex=2)
lines(x, flexsurv::dgengamma(x, log(mu), sig, Q=sig*nu), col=8, lwd=5)
lines(x, pdfgam(x, para), col=2)
```

What complicates the discussion further is that seemingly only the *log.link* concept is manifested in the use of $\log(mu)$ to provide the μ_2 for `flexsurv::dgengamma`.

On the Log-Normal via Generalized Gamma
The **gamlss.dist** package uses an $|\nu| < 1\mathrm{e}{-6}$ trigger for the log-Normal calls. Further testing and the initial independent origin of **lmomco** code suggests that a primary trigger though can be based on the finiteness of the `lgamma(theta)` for θ. This is used in `pdfgam` as well as `cdfgam` and `quagam`.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`cdfgam`, `quagam`, `lmomgam`, `pargam`

Examples

```
  lmr <- lmoms(c(123,34,4,654,37,78))
  gam <- pargam(lmr)
  x <- quagam(0.5,gam)
  pdfgam(x,gam)

## Not run:
# 3-p Generalized Gamma Distribution and gamlss.dist package parameterization
gg <- vec2par(c(7.4, 0.2, 14), type="gam"); X <- seq(0.04,9, by=.01)
GGa <- gamlss.dist::dGG(X, mu=7.4, sigma=0.2, nu=14)
GGb <- pdfgam(X, gg) # We now compare the two densities.
plot( X, GGa, type="l", xlab="X", ylab="PROBABILITY DENSITY", col=3, lwd=6)
lines(X, GGb, col=2, lwd=2) #
## End(Not run)

## Not run:
# 3-p Generalized Gamma Distribution and gamlss.dist package parameterization
gg <- vec2par(c(1.7, 3, -4), type="gam"); X <- seq(0.04,9, by=.01)
GGa <- gamlss.dist::dGG(X, mu=1.7, sigma=3, nu=-4)
GGb <- pdfgam(X, gg) # We now compare the two densities.
plot( X, GGa, type="l", xlab="X", ylab="PROBABILITY DENSITY", col=3, lwd=6)
lines(X, GGb, col=2, lwd=2) #
## End(Not run)
```

pdfgdd *Probability Density Function of the Gamma Difference Distribution*

Description

This function computes the probability density of the Gamma Difference distribution (Klar, 2015) given parameters ($\alpha_1 > 0$, $\beta_1 > 0$, $\alpha_2 > 0$, $\beta_2 > 0$) computed by pargdd.

$$f(x, x > 0) = ce^{+\beta_2 x} \int_{+x}^{\infty} z^{\alpha_1 - 1}(z - x)^{\alpha_2 - 1} e^{-(\beta_1 + \beta_2) z}\, \mathrm{d}z,$$

and

$$f(x, x < 0) = ce^{-\beta_1 x} \int_{-x}^{\infty} z^{\alpha_2 - 1}(z + x)^{\alpha_1 - 1} e^{-(\beta_1 + \beta_2) z}\, \mathrm{d}z,$$

where c is defined as

$$c = \frac{\beta_1^{\alpha_1} \beta_2^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)},$$

where $\Gamma(y)$ is the complete gamma function.

Usage

```
pdfgdd(x, para, paracheck=TRUE, silent=TRUE, ...)
```

Arguments

`x`	A real value vector.
`para`	The parameters from pargdd or vec2par.
`paracheck`	A logical controlling whether the parameters are checked for validity.
`silent`	The argument of `silent` for the `try()` operation wrapped on `integrate()`.
`...`	Additional argument to pass.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Klar, B., 2015, A note on gamma difference distributions: Journal of Statistical Computation and Simulation v. 85, no. 18, pp. 1–8, doi:10.1080/00949655.2014.996566.

See Also

cdfgdd, quagdd, lmomgdd, pargdd

Examples

```
## Not run:
x <- seq(-8, 8, by=0.01) # the operations on x are to center
para <- list(para=c(3,   1, 1, 1), type="gdd")
plot(x-(3  /1 - 1/1), pdfgdd(x, para), type="l", xlim=c(-6,6), ylim=c(0, 0.7),
     xlab="x", ylab="density of gamma difference distribution")
para <- list(para=c(2,   1, 1, 1), type="gdd")
lines(x-(2  /1 - 1/1), pdfgdd(x, para), lty=2)
para <- list(para=c(1,   1, 1, 1), type="gdd")
lines(x-(1  /1 - 1/1), pdfgdd(x, para), lty=3)
para <- list(para=c(0.5, 1, 1, 1), type="gdd")
lines(x-(0.5/1 - 1/1), pdfgdd(x, para), lty=4) #
## End(Not run)
```

---

pdfgep *Probability Density Function of the Generalized Exponential Poisson Distribution*

Description

This function computes the probability density of the Generalized Exponential Poisson distribution given parameters (β, κ, and h) computed by pargep. The probability density function is

$$f(x) = \frac{\kappa h \eta}{[1-\exp(-h)]^\kappa} 1 - \exp[-h + h\exp(-\eta x) \times \exp[-h - \eta x + h\exp(-\eta x)],$$

where $F(x)$ is the nonexceedance probability for quantile $x > 0$, $\eta = 1/\beta$, $\beta > 0$ is a scale parameter, $\kappa > 0$ is a shape parameter, and $h > 0$ is another shape parameter.

Usage

```
pdfgep(x, para)
```

Arguments

x	A real value vector.
para	The parameters from pargep or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Barreto-Souza, W., and Cribari-Neto, F., 2009, A generalization of the exponential-Poisson distribution: Statistics and Probability, 79, pp. 2493–2500.

See Also

pdfgep, quagep, lmomgep, pargep

Examples

```
pdfgep(0.5, vec2par(c(10,2.9,1.5), type="gep"))
## Not run:
x <- seq(0,3, by=0.01); ylim <- c(0,1.5)
plot(NA,NA, xlim=range(x), ylim=ylim, xlab="x", ylab="f(x)")
mtext("Barreto-Souza and Cribari-Neto (2009, fig. 1)")
K <- c(0.1, 1, 5, 10)
for(i in 1:length(K)) {
   gep <- vec2par(c(2,K[i],1), type="gep"); lines(x, pdfgep(x, gep), lty=i)
}

## End(Not run)
```

---

pdfgev — *Probability Density Function of the Generalized Extreme Value Distribution*

Description

This function computes the probability density of the Generalized Extreme Value distribution given parameters (ξ, α, and κ) computed by pargev. The probability density function is

$$f(x) = \alpha^{-1} \exp[-(1-\kappa)Y - \exp(-Y)],$$

where Y is

$$Y = -\kappa^{-1} \log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$, and

$$Y = (x-\xi)/\alpha,$$

for $\kappa = 0$, where $f(x)$ is the probability density for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of x is $-\infty < x \leq \xi + \alpha/\kappa$ if $k > 0$; $\xi + \alpha/\kappa \leq x < \infty$ if $\kappa \leq 0$. Note that the shape parameter κ parameterization of the distribution herein follows that in tradition by the greater L-moment community and others use a sign reversal on κ. (The **evd** package is one example.)

Usage

```
pdfgev(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from `pargev` or `vec2par`.
paracheck	A logical switch as to whether the validity of the parameters should be checked.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`cdfgev`, `quagev`, `lmomgev`, `pargev`

Examples

```
  lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
  gev <- pargev(lmr)
  x <- quagev(0.5, gev)
       pdfgev(  x, gev)

## Not run:
  # We explore using maximum likelihood for GEV estimation on its density function.
  # We check the convergence and check on parameters back estimating the mean.
  small <- .Machine$double.eps
  for(k in c(-2, -1/2, -small, 0, +small, 1/2, 2)) {
    names(k) <- "myKappa"
    gev <- vec2par(c(2, 2, k), type="gev")
    x <- rlmomco(1000, gev)
    mu1 <- mean(x); names(mu1) <- "mean"
    cv1 <-       NA; names(cv1) <- "converge"
    mle <- mle2par(x, type="gev", init.para=pargev(lmoms(x)),
              ptransf=function(t) { c(t[1], log(t[2]), t[3]) },
            pretransf=function(t) { c(t[1], exp(t[2]), t[3]) },
                         null.on.not.converge=FALSE)
    mu2 <- lmomgev(mle)$lambdas[1]; names(mu2) <- "backMean"
    cv2 <- mle$optim$convergence;   names(cv2) <- "converge"
```

```
    print(round(c(k, cv1, mu1, gev$para), digits=5))
    print(round(c(k, cv2, mu2, mle$para), digits=5))
  } #
## End(Not run)
```

---

pdfgld *Probability Density Function of the Generalized Lambda Distribution*

Description

This function computes the probability density function of the Generalized Lambda distribution given parameters (ξ, α, κ, and h) computed by `pargld` or similar. The probability density function is

$$f(x) = [(\kappa[F(x)^{\kappa-1}] + h[1 - F(x)])^{h-1}) \times \alpha]^{-1},$$

where $f(x)$ is the probability density function at x, $F(x)$ is the cumulative distribution function at x.

Usage

```
pdfgld(x, para, paracheck)
```

Arguments

`x`	A real value vector.
`para`	The parameters from `pargld` or `vec2par`.
`paracheck`	A logical switch as to whether the validity of the parameters should be checked. Default is `paracheck=TRUE`. This switch is made so that the root solution needed for `cdfgld` exhibits an extreme speed increase because of the repeated calls to `quagld`.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

`cdfgld`, `quagld`, `lmomgld`, `pargld`

Examples

```
## Not run:
# Using Karian and Dudewicz, 2000, p. 10
gld <- vec2par(c(0.0305,1/1.3673,0.004581,0.01020),type='gld')
quagld(0.25,gld) # which equals about 0.028013 as reported by K&D
pdfgld(0.028013,gld) # which equals about 43.04 as reported by K&D
F <- seq(.001,.999,by=.001)
x <- quagld(F,gld)
plot(x, pdfgld(x,gld), type='l', xlim=c(0,.1))

## End(Not run)
```

---

pdfglo *Probability Density Function of the Generalized Logistic Distribution*

Description

This function computes the probability density of the Generalized Logistic distribution given parameters (ξ, α, and κ) computed by parglo. The probability density function is

$$f(x) = \frac{\alpha^{-1}\exp(-(1-\kappa)Y)}{[1+\exp(-Y)]^2},$$

where Y is

$$Y = -\kappa^{-1}\log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$, and

$$Y = (x-\xi)/\alpha,$$

for $\kappa = 0$, and where $f(x)$ is the probability density for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
pdfglo(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parglo or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfglo, quaglo, lmomglo, parglo

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
glo <- parglo(lmr)
x <- quaglo(0.5,glo)
pdfglo(x,glo)
```

---

pdfgno *Probability Density Function of the Generalized Normal Distribution*

Description

This function computes the probability density of the Generalized Normal distribution given parameters (ξ, α, and κ) computed by pargno. The probability density function function is

$$f(x) = \frac{\exp(\kappa Y - Y^2/2)}{\alpha\sqrt{2\pi}},$$

where Y is

$$Y = -\kappa^{-1} \log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$, and

$$Y = (x-\xi)/\alpha,$$

for $\kappa = 0$, where $f(x)$ is the probability density for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
pdfgno(x, para)
```

Arguments

x	A real value vector.
para	The parameters from pargno or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgno, quagno, lmomgno, pargno, pdfln3

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
gno <- pargno(lmr)
x <- quagno(0.5,gno)
pdfgno(x,gno)
```

---

pdfgov *Probability Density Function of the Govindarajulu Distribution*

Description

This function computes the probability density of the Govindarajulu distribution given parameters (ξ, α, and β) computed by pargov. The probability density function is

$$f(x) = [\alpha\beta(\beta+1)]^{-1}[F(x)]^{1-\beta}[1-F(x)]^{-1},$$

where $f(x)$ is the probability density for quantile x, $F(x)$ the cumulative distribution function or nonexceedance probability at x, ξ is a location parameter, α is a scale parameter, and β is a shape parameter.

Usage

```
pdfgov(x, para)
```

Arguments

x	A real value vector.
para	The parameters from pargov or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Nair, N.U., Sankaran, P.G., Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

Nair, N.U., Sankaran, P.G., and Vineshkumar, B., 2012, The Govindarajulu distribution—Some Properties and applications: Communications in Statistics, Theory and Methods, v. 41, no. 24, pp. 4391–4406.

See Also

cdfgov, quagov, lmomgov, pargov

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
gov <- pargov(lmr)
x <- quagov(0.5,gov)
pdfgov(x,gov)
```

---

pdfgpa *Probability Density Function of the Generalized Pareto Distribution*

Description

This function computes the probability density of the Generalized Pareto distribution given parameters (ξ, α, and κ) computed by pargpa. The probability density function is

$$f(x) = \alpha^{-1}\exp(-(1-\kappa)Y),$$

where Y is

$$Y = -\kappa^{-1}\log\left(1 - \frac{\kappa(x-\xi)}{\alpha}\right),$$

for $\kappa \neq 0$, and

$$Y = (x - \xi)/\alpha,$$

for $\kappa = 0$, where $f(x)$ is the probability density for quantile x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of x is $\xi \leq x \leq \xi + \alpha/\kappa$ if $k > 0$; $\xi \leq x < \infty$ if $\kappa \leq 0$. Note that the shape parameter κ parameterization of the distribution herein follows that in tradition by the greater L-moment community and others use a sign reversal on κ. (The **evd** package is one example.)

Usage

```
pdfgpa(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from pargpa or vec2par.
paracheck	A logical switch as to whether the validity of the parameters should be checked.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgpa, quagpa, lmomgpa, pargpa

Examples

```
  lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
  gpa <- pargpa(lmr)
  x   <- quagpa(0.5, gpa)
         pdfgpa(  x, gpa)

## Not run:
  # We explore using maximum likelihood for GPA estimation on its density function
```

```
  # with stress testing near the K > -1 lower limit, K near zero, and then large K
  # producing extreme densities. We check the convergence and check on parameters
  # back estimating the mean. The experiment is designed that with repeated
  # operations that convergence "failures" in stats::optim()
  #   1  'indicates that the iteration limit maxit had been reached'
  #   10 'indicates degeneracy of the Nelder-Mead simplex.'
  # With the 10 being a bit more common and 1 but still for many runs convergence
  # at K = 8 is still attainable. Also, note the care in the construction of the
  # ptransf and pretransf for the honoring the GPA parameter space.
  small <- .Machine$double.eps; n <- 1000 # samples
  for(k in c(-1+small, -0.99, -1/2, -small, 0, 1/2, 8)) {
    names(k) <- "myKappa"
    gpa <- vec2par(c(2, 2, k), type="gpa")
    x <- rlmomco(n, gpa)
    mu1 <- mean(x); names(mu1) <- "mean"
    cv1 <-      NA; names(cv1) <- "converge"
    mle <- mle2par(x, type="gpa", init.para=pargpa(lmoms(x)),
             ptransf=function(t) { c(t[1], log(t[2]), log(t[3] +1)) },
           pretransf=function(t) { c(t[1], exp(t[2]), exp(t[3])-1)  },
                     null.on.not.converge=FALSE)
    mu2 <- lmomgpa(mle)$lambdas[1]; names(mu2) <- "backMean"
    cv2 <- mle$optim$convergence;   names(cv2) <- "converge"
    print(round(c(k, cv1, mu1, gpa$para), digits=5))
    print(round(c(k, cv2, mu2, mle$para), digits=5))
  } #
## End(Not run)
```

---

pdfgum *Probability Density Function of the Gumbel Distribution*

Description

This function computes the probability density of the Gumbel distribution given parameters (ξ and α) computed by pargum. The probability density function is

$$f(x) = \alpha^{-1} \exp(Y) \exp[-\exp(Y)],$$

where

$$Y = -\frac{x - \xi}{\alpha},$$

where $f(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
pdfgum(x, para)
```

Arguments

x	A real value vector.
para	The parameters from pargum or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgum, quagum, lmomgum, pargum

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
gum <- pargum(lmr)
x <- quagum(0.5,gum)
pdfgum(x,gum)
```

---

pdfkap *Probability Density Function of the Kappa Distribution*

Description

This function computes the probability density of the Kappa distribution given parameters (ξ, α, κ, and h) computed by parkap. The probability density function is

$$f(x) = \alpha^{-1}[1 - \kappa(x - \xi)/\alpha]^{1/k-1} \times [F(x)]^{1-h}$$

where $f(x)$ is the probability density for quantile x, $F(x)$ is the cumulative distribution function or nonexceedance probability at x, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
pdfkap(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parkap or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

Sourced from written communication with Dr. Hosking in October 2007.

See Also

cdfkap, quakap, lmomkap, parkap

Examples

```
kap <- vec2par(c(1000,15000,0.5,-0.4),type='kap')
F <- nonexceeds()
x <- quakap(F,kap)
check.pdf(pdfkap,kap,plot=TRUE)
```

---

pdfkmu *Probability Density Function of the Kappa-Mu Distribution*

Description

This function computes the probability density of the Kappa-Mu ($\kappa : \mu$) distribution given parameters (κ and μ) computed by parkmu. The probability density function is

$$f(x) = \frac{2\mu(1+\kappa)^{(\mu+1)/2}}{\kappa^{(\mu-1)/2}\mathrm{exp}(\mu\kappa)}\, x^{\mu}\ \exp(-\mu(1+\kappa)x^2)\, I_{\mu-1}(2\mu\sqrt{\kappa(1+\kappa)}x),$$

where $f(x)$ is the nonexceedance probability for quantile x, and the modified Bessel function of the first kind is $I_k(x)$, and define m as

$$m = \frac{\mu(1+\kappa)^2}{1+2\kappa}.$$

and for a given m, the new parameter μ must lie in the range

$$0 \le \mu \le m.$$

The definition of $I_k(x)$ is seen under pdfemu. Lastly, if $\kappa = \infty$, then there is a Dirca Delta function of probability at $x = 0$.

Usage

```
pdfkmu(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from parkmu or vec2par.
paracheck	A logical controlling whether the parameters and checked for validity.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

cdfkmu, quakmu, lmomkmu, parkmu

Examples

```
## Not run:
x <- seq(0,4, by=.1)
para <- vec2par(c(.5, 1.4), type="kmu")
F <- cdfkmu(x, para)
X <- quakmu(F, para, quahi=pi)
plot(F, X, type="l", lwd=8)
lines(F, x, col=2)

## End(Not run)
## Not run:
# Note that in this example very delicate steps are taken to show
# how one interacts with the Dirac Delta function (x=0) when the m
# is known but mu == 0. For x=0, the fraction of total probability
# is recorded, but when one is doing numerical summation to evaluate
# whether the total probability under the PDF is unity some algebraic
# manipulations are needed as shown for the conditional when kappa
# is infinity.

delx <- 0.001
x <- seq(0,3, by=delx)

plot(c(0,3), c(0,1), xlab="RHO", ylab="pdfkmu(RHO)", type="n")
m <- 1.25
```

```
mus <- c(0.25, 0.50, 0.75, 1, 1.25, 0)
for(mu in mus) {
   kappa <- m/mu - 1 + sqrt((m/mu)*((m/mu)-1))
   para <- vec2par(c(kappa, mu), type="kmu")
   if(! is.finite(kappa)) {
      para <- vec2par(c(Inf,m), type="kmu")
      density <- pdfkmu(x, para)
      lines(x, density, col=2, lwd=3)
      dirac <- 1/delx - sum(density[x != 0])
      cumulant <- (sum(density) + density[1]*(1/delx - 1))*delx
      density[x == 0] <- rep(dirac, length(density[x == 0]))
      message("Total integrated probability is ", cumulant, "\n")
   }
   lines(x, pdfkmu(x, para))
}
mtext("Yacoub (2007, figure 3)")

## End(Not run)
```

---

pdfkur *Probability Density Function of the Kumaraswamy Distribution*

Description

This function computes the probability density of the Kumaraswamy distribution given parameters (α and β) computed by parkur. The probability density function is

$$f(x) = \alpha\beta x^{\alpha-1}(1 - x^{\alpha})^{\beta-1},$$

where $f(x)$ is the nonexceedance probability for quantile x, α is a shape parameter, and β is a shape parameter.

Usage

```
pdfkur(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parkur or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Jones, M.C., 2009, Kumaraswamy's distribution—A beta-type distribution with some tractability advantages: Statistical Methodology, v. 6, pp. 70–81.

See Also

cdfkur, quakur, lmomkur, parkur

Examples

```
lmr <- lmoms(c(0.25, 0.4, 0.6, 0.65, 0.67, 0.9))
kur <- parkur(lmr)
x <- quakur(0.5,kur)
pdfkur(x,kur)
```

---

pdflap *Probability Density Function of the Laplace Distribution*

Description

This function computes the probability density of the Laplace distribution given parameters (ξ and α) computed by parlap. The probability density function is

$$f(x) = (2\alpha)^{-1} \exp(Y),$$

where Y is

$$Y = \left(\frac{-|x - \xi|}{\alpha}\right).$$

Usage

```
pdflap(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parlap or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: IBM Research Report RC12210, T.J. Watson Research Center, Yorktown Heights, New York.

See Also

cdflap, qualap, lmomlap, parlap

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
lap <- parlap(lmr)
x <- qualap(0.5,lap)
pdflap(x,lap)
```

---

pdflmrq *Probability Density Function of the Linear Mean Residual Quantile Function Distribution*

Description

This function computes the probability density function of the Linear Mean Residual Quantile Function distribution given parameters computed by parlmrq. The probability density function is

$$f(x) = \frac{1 - F(x)}{2\alpha\, F(x) + (\mu - \alpha)},$$

where $f(x)$ is the nonexceedance probability for quantile x, $F(x)$ is the cumulative distribution function or nonexceedance probability at x, μ is a location parameter, and α is a scale parameter.

Usage

```
pdflmrq(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parlmrq or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

See Also

cdflmrq, qualmrq, lmomlmrq, parlmrq

Examples

```
lmr <- lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2))
pdflmrq(3,parlmrq(lmr))
## Not run:
para.lmrq <- list(para=c(2.1043, 0.4679), type="lmrq")
para.wei  <- vec2par(c(0,2,0.9), type="wei") # note switch from Midhu et al. ordering.
F <- seq(0.01,0.99,by=.01); x <- qualmrq(F, para.lmrq)
plot(x, pdflmrq(x, para.lmrq), type="l", ylab="", lwd=2, lty=2, col=2,
     xlab="The p.d.f. of Weibull and p.d.f. of LMRQD", xaxs="i", yaxs="i",
     xlim=c(0,9), ylim=c(0,0.8))
lines(x, pdfwei(x, para.wei))
mtext("Midhu et al. (2013, Statis. Meth.)")

## End(Not run)
```

`pdfln3` *Probability Density Function of the 3-Parameter Log-Normal Distribution*

Description

This function computes the probability density of the Log-Normal3 distribution given parameters (ζ, lower bounds; μ_{log}, location; and σ_{log}, scale) computed by parln3. The probability density function function (same as Generalized Normal distribution, pdfgno) is

$$f(x) = \frac{\exp(\kappa Y - Y^2/2)}{\alpha\sqrt{2\pi}},$$

where Y is

$$Y = \frac{\log(x - \zeta) - \mu_{\mathrm{log}}}{\sigma_{\mathrm{log}}},$$

where ζ is the lower bounds (real space) for which $\zeta < \lambda_1 - \lambda_2$ (checked in are.parln3.valid), μ_{log} be the mean in natural logarithmic space, and σ_{log} be the standard deviation in natural logarithm space for which $\sigma_{\mathrm{log}} > 0$ (checked in are.parln3.valid) is obvious because this parameter has an analogy to the second product moment. Letting $\eta = \exp(\mu_{\mathrm{log}})$, the parameters of the Generalized Normal are $\zeta + \eta$, $\alpha = \eta\sigma_{\mathrm{log}}$, and $\kappa = -\sigma_{\mathrm{log}}$. At this point, the algorithms (pdfgno) for the Generalized Normal provide the functional core.

Usage

```
pdfln3(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parln3 or vec2par.

Value

Probability density (f) for x.

Note

The parameterization of the Log-Normal3 results in ready support for either a known or unknown lower bounds. Details regarding the parameter fitting and control of the ζ parameter can be seen under the Details section in parln3.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

cdfln3, qualn3, lmomln3, parln3, pdfgno

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
ln3 <- parln3(lmr); gno <- pargno(lmr)
x <- qualn3(0.5,ln3)
pdfln3(x,ln3) # 0.008053616
pdfgno(x,gno) # 0.008053616 (the distributions are the same, but see Note)
```

`pdfnor` *Probability Density Function of the Normal Distribution*

Description

This function computes the probability density function of the Normal distribution given parameters computed by `parnor`. The probability density function is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right),$$

where $f(x)$ is the probability density for quantile x, μ is the arithmetic mean, and σ is the standard deviation. The R function `pnorm` is used.

Usage

```
pdfnor(x, para)
```

Arguments

`x`	A real value.
`para`	The parameters from `parnor` or `vec2par`.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`cdfnor`, `quanor`, `lmomnor`, `parnor`

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
pdfnor(50,parnor(lmr))
```

pdfpdq3 *Probability Density Function of the Polynomial Density-Quantile3 Distribution*

Description

This function computes the probability density of the Polynomial Density-Quantile3 distribution given parameters (α and β) computed by parpdq3. The probability density function has not explicit form. The implementation here simply uses a *five-point stencil* to approciate the derivative of the cumulative distribution function cdfpdq3 and hence an eps term is used and multipled to the scale parameter (α) of the distribution. The distribution's canonical definition is in terms of the quantile function (quapdq3).

Usage

```
pdfpdq3(x, para, paracheck=TRUE, h=NA, hfactor=0.2)
```

Arguments

x	A real value vector.
para	The parameters from parpdq4 or vec2par.
paracheck	A logical switch as to whether the validity of the parameters should be checked. Default is paracheck=TRUE. This switch is made so that the root solution needed for cdfpdq3 shows an extreme speed increase because of the repeated calls to quapdq3.
h	The differential element of the stencil, if provided, otherwise hfactor used.
hfactor	A term multiplied to the α parameter to set the h in the numerical derivative. Not optimal, but seems to work for a variety of chosen parameters for plotting the density function.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

cdfpdq3, quapdq3, lmompdq3, parpdq3, pdfpdq4

Examples

```
## Not run:
  para <- list(para=c(0.6933, 1.5495, 0.5488), type="pdq3")
  X <- seq(-5, +12, by=(12 - -5) / 1000)
  plot( X, pdfpdq3(X, para), type="l", col=grey(0.8), lwd=4, ylim=c(0, 0.3))
  lines(X, c(NA, diff(pf(exp(X), df1=7, df2=1))/((12 - -5) / 1000)), lty=2)
  legend("topleft", c("log F(7,1) distribution with same L-moments",
                      "PDQ3 distribution with same L-moments as the log F(7,1)"),
                    lwd=c(1, 4), lty=c(2, 1), col=c(1, grey(0.8)), cex=0.8)
  mtext("Mimic Hosking (2007, fig. 2 [left])")
  check.pdf(pdfpdq3, para) #
## End(Not run)

## Not run:
  para <- list(para=c(100, 43.32, -0.7029), type="pdq3")
  minX <- quapdq3(0.0001, para)
  maxX <- quapdq3(0.9999, para)
  X <- seq(minX, maxX, by=(maxX - minX) / 1000)
  plot( X, pdfpdq3(X, para), type="l", col=grey(0.8), lwd=4)
  check.pdf(pdfpdq3, para) #
## End(Not run)

## Not run:
  para <- vec2par(c(0.4729820, 3.0242067, 0.9880701), type="pdq3")
  print(lmom2par(par2lmom(para), type="pdq3"))
  # "|kappa| > 0.98, alpha (yes alpha) results could be unreliable"
  # So, we are entering into a problem for which the kappa parameter is
  # very large and instabilities in the algorithm will result, but
  # vec2par() has not mechanism for determining this type of situation.
  # Ultimately, things will manifest with a check.pdf() that fails.
  sup <- lmomco::supdist(para)$support
  xx <- seq(sup[1], sup[2], by=diff(range(sup)) / 2000)
  plot(xx, pdfpdq3(xx, para), type="l", col=grey(0.8))
  plot(xx, pdfpdq3(xx, para), type="l", col=grey(0.8), xlim=c(-1,10))
  # See hints of instability in the density shape in the second plot
  check.pdf(pdfpdq3, para) # non-finite function value
## End(Not run)
```

---

pdfpdq4 *Probability Density Function of the Polynomial Density-Quantile4 Distribution*

Description

This function computes the probability density of the Polynomial Density-Quantile4 distribution given parameters (α and β) computed by parpdq4. The probability density function has not explicit form. The implementation here simply uses a *five-point stencil* to approciate the derivative of the cumulative distribution function cdfpdq4 and hence an eps term is used and multipled to the scale parameter (α) of the distribution. The distribution's canonical definition is in terms of the quantile function (quapdq4).

Usage

```
pdfpdq4(x, para, paracheck=TRUE, h=NA, hfactor=0.2)
```

Arguments

x	A real value vector.
para	The parameters from parpdq4 or vec2par.
paracheck	A logical switch as to whether the validity of the parameters should be checked. Default is paracheck=TRUE. This switch is made so that the root solution needed for cdfpdq4 shows an extreme speed increase because of the repeated calls to quapdq4.
h	The differential element of the stencil, if provided, otherwise hfactor used.
hfactor	A term multiplied to the α parameter to set the h in the numerical derivative. Not optimal, but seems to work for a variety of chosen parameters for plotting the density function.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

cdfpdq4, quapdq4, lmompdq4, parpdq4, pdfpdq3

Examples

```
## Not run:
  para <- list(para=c(0, 0.4332, -0.7029), type="pdq4")
  X <- seq(-4, +4, by=(4 - -4) / 1000)
  plot( X, pdfpdq4(X, para), type="l", col=grey(0.8), lwd=4, ylim=c(0, 0.5))
  lines(X, dnorm(  X, sd=1), lty=2)
  legend("topleft", c("Standard normal distribution",
                      "PDQ4 distribution with same L-moments as the standard normal"),
                    lwd=c(1, 4), lty=c(2, 1), col=c(1, grey(0.8)), cex=0.8)
  mtext("Mimic Hosking (2007, fig. 3 [left])")
  check.pdf(pdfpdq4, para, hfactor=0.3)
## End(Not run)

## Not run:
```

```
  para <- list(para=c(100, 43.32, -0.7029), type="pdq4")
  minX <- quapdq4(0.0001, para)
  maxX <- quapdq4(0.9999, para)
  X <- seq(minX, maxX, by=(maxX - minX) / 1000)
  plot( X, pdfpdq4(X, para), type="l", col=grey(0.8), lwd=4)

  check.pdf(pdfpdq4, para, hfactor=0.3)
## End(Not run)
```

pdfpe3 *Probability Density Function of the Pearson Type III Distribution*

Description

This function computes the probability density of the Pearson Type III distribution given parameters (μ, σ, and γ) computed by parpe3. These parameters are equal to the product moments (pmoms): mean, standard deviation, and skew. The probability density function for $\gamma \neq 0$ is

$$f(x) = \frac{Y^{\alpha-1}\exp(-Y/\beta)}{\beta^{\alpha}\,\Gamma(\alpha)},$$

where $f(x)$ is the probability density for quantile x, Γ is the complete gamma function in R as gamma, ξ is a location parameter, β is a scale parameter, α is a shape parameter, and $Y = x - \xi$ for $\gamma > 0$ and $Y = \xi - x$ for $\gamma < 0$. These three "new" parameters are related to the product moments (μ, mean; σ, standard deviation; γ, skew) by

$$\alpha = 4/\gamma^2,$$

$$\beta = \frac{1}{2}\sigma|\gamma|, \text{ and}$$

$$\xi = \mu - 2\sigma/\gamma.$$

If $\gamma = 0$, the distribution is symmetrical and simply is the probability density Normal distribution with mean and standard deviation of μ and σ, respectively. Internally, the $\gamma = 0$ condition is implemented by R function dnorm. The **PearsonDS** package supports the Pearson distribution system including the Type III (see **Examples**).

Usage

```
pdfpe3(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parpe3 or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfpe3, quape3, lmompe3, parpe3

Examples

```
  lmr <- lmoms(c(123,34,4,654,37,78))
  pe3 <- parpe3(lmr)
  x <- quape3(0.5,pe3)
  pdfpe3(x,pe3)
## Not run:
# Demonstrate Pearson Type III between lmomco and PearsonDS
qlmomco.pearsonIII <- function(f, para) {
   MU    <- para$para[1] # product moment mean
   SIGMA <- para$para[2] # product moment standard deviation
   GAMMA <- para$para[3] # product moment skew
   L <- para$para[1] - 2*SIGMA/GAMMA # location
   S <- (1/2)*SIGMA*abs(GAMMA)       # scale
   A <- 4/GAMMA^2                    # shape
   return(PearsonDS::qpearsonIII(f, A, L, S)) # shape comes first!
}
FF <- nonexceeds(); para <- vec2par(c(6,.4,.7), type="pe3")
plot( FF, qlmomco(FF, para), xlab="Probability", ylab="Quantile", cex=3)
lines(FF, qlmomco.pearsonIII(FF, para), col=2, lwd=3) #
## End(Not run)

## Not run:
# Demonstrate forced Pearson Type III parameter estimation via PearsonDS package
para <- vec2par(c(3, 0.4, 0.6), type="pe3"); X <- rlmomco(105, para)
lmrpar <- lmom2par(lmoms(X), type="pe3")
mpspar <- mps2par(X, type="pe3"); mlepar <- mle2par(X, type="pe3")
PDS <- PearsonDS:::pearsonIIIfitML(X) # force function exporting
if(PDS$convergence != 0) {
  warning("convergence failed"); PDS <- NULL # if null, rerun simulation [new data]
} else {
  # This is a list() mimic of PearsonDS::pearsonFitML()
  PDS   <- list(type=3, shape=PDS$par[1], location=PDS$par[2], scale=PDS$par[3])
  skew  <- sign(PDS$shape) * sqrt(4/PDS$shape)
  stdev <-    2*PDS$scale  / abs(skew); mu <- PDS$location + 2*stdev/skew
  PDS <- vec2par(c(mu,stdev,skew), type="pe3") # lmomco form of parameters
}
```

```
print(lmrpar$para); print(mpspar$para); print(mlepar$para); print(PDS$para)
#        mu     sigma     gamma
# 2.9653380 0.3667651 0.5178592 # L-moments (by lmomco, of course)
# 2.9678021 0.3858198 0.4238529 # MPS by lmomco
# 2.9653357 0.3698575 0.4403525 # MLE by lmomco
# 2.9653379 0.3698609 0.4405195 # MLE by PearsonDS
# So we can see for this simulation that the two MLE approaches are similar.
## End(Not run)
```

---

pdfray *Probability Density Function of the Rayleigh Distribution*

Description

This function computes the probability density of the Rayleigh distribution given parameters (ξ and α) computed by parray. The probability density function is

$$f(x) = \frac{x - \xi}{\alpha^2} \exp\left(\frac{-(x - \xi)^2}{2\alpha^2}\right),$$

where $f(x)$ is the nonexceedance probability for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
pdfray(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parray or similar.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

See Also

cdfray, quaray, lmomray, parray

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
ray <- parray(lmr)
x <- quaray(0.5,ray)
pdfray(x,ray)
```

---

pdfrevgum *Probability Density Function of the Reverse Gumbel Distribution*

Description

This function computes the probability density of the Reverse Gumbel distribution given parameters (ξ and α) computed by parrevgum. The probability density function is

$$f(x) = \alpha^{-1} \exp(Y)[\exp(\exp[-\exp(Y)])],$$

where

$$Y = \frac{x - \xi}{\alpha},$$

where $f(x)$ is the probability density for quantile x, ξ is a location parameter, and α is a scale parameter.

Usage

```
pdfrevgum(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parrevgum or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

cdfrevgum, quarevgum, lmomrevgum, parrevgum

Examples

```
# See p. 553 of Hosking (1995)
# Data listed in Hosking (1995, table 29.3, p. 553)
D <- c(-2.982, -2.849, -2.546, -2.350, -1.983, -1.492, -1.443,
       -1.394, -1.386, -1.269, -1.195, -1.174, -0.854, -0.620,
       -0.576, -0.548, -0.247, -0.195, -0.056, -0.013,  0.006,
        0.033,  0.037,  0.046,  0.084,  0.221,  0.245,  0.296)
D <- c(D,rep(.2960001,40-28)) # 28 values, but Hosking mentions
                              # 40 values in total
z <-  pwmRC(D,threshold=.2960001)
str(z)
# Hosking reports B-type L-moments for this sample are
# lamB1 = -0.516 and lamB2 = 0.523
btypelmoms <- pwm2lmom(z$Bbetas)
# My version of R reports lamB1 = -0.5162 and lamB2 = 0.5218
str(btypelmoms)
rg.pars <- parrevgum(btypelmoms,z$zeta)
str(rg.pars)
# Hosking reports xi=0.1636 and alpha=0.9252 for the sample
# My version of R reports xi = 0.1635 and alpha = 0.9254
# Now one can continue one with a plotting example.
## Not run:
F  <- nonexceeds()
PP <- pp(D) # plotting positions of the data
D  <- sort(D)
plot(D,PP)
lines(D,cdfrevgum(D,rg.pars))
# Now finally do the PDF
F <- seq(0.01,0.99,by=.01)
x <- quarevgum(F,rg.pars)
plot(x,pdfrevgum(x,rg.pars),type='l')

## End(Not run)
```

---

pdfrice *Probability Density Function of the Rice Distribution*

Description

This function computes the probability density of the Rice distribution given parameters (ν and SNR) computed by parrice. The probability density function is

$$f(x) = \frac{x}{\alpha^2}\,\exp\!\left(\frac{-(x^2+\nu^2)}{2\alpha^2}\right)\,I_0(x\nu/\alpha^2),$$

where $f(x)$ is the nonexceedance probability for quantile x, ν is a parameter, and ν/α is a form of signal-to-noise ratio SNR, and $I_k(x)$ is the modified Bessel function of the first kind, which for integer $k = 0$ is seen under LaguerreHalf. If $\nu = 0$, then the Rayleigh distribution results and pdfray is used. If $24 < \mathrm{SNR} < 52$ is used, then the Normal distribution functions are used with appropriate parameter estimation for μ and σ that include the Laguerre polynomial LaguerreHalf. If $\mathrm{SNR} > 52$, then the Normal distribution functions continue to be used with $\mu = \alpha \times \mathrm{SNR}$ and $\sigma = A$.

Usage

```
pdfrice(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parrice or vec2par.

Value

Probability density (f) for x.

Note

The **VGAM** package provides a pdf of the Rice for reference:

```
"drice" <- function(x, vee, sigma, log = FALSE) { # From the VGAM package
    if(!is.logical(log.arg <- log)) stop("bad input for argument 'log'")
    rm(log)
    N = max(length(x), length(vee), length(sigma))
    x = rep(x, len=N); vee = rep(vee, len=N); sigma = rep(sigma, len=N)
    logdensity = rep(log(0), len=N)
    xok = (x > 0)
    x.abs = abs(x[xok]*vee[xok]/sigma[xok]^2)
    logdensity[xok] = log(x[xok]) - 2 * log(sigma[xok]) +
                      (-(x[xok]^2+vee[xok]^2)/(2*sigma[xok]^2)) +
                      log(besselI(x.abs, nu=0, expon.scaled = TRUE)) + x.abs
    logdensity[sigma <= 0] <- NaN; logdensity[vee < 0] <- NaN
    if(log.arg) logdensity else exp(logdensity)
}
```

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

cdfrice, quarice, lmomrice, parrice

Examples

```
lmr <- lmoms(c(10, 43, 27, 26, 49, 26, 62, 39, 51, 14))
rice <- parrice(lmr)
x <- quarice(nonexceeds(),rice)
plot(x,pdfrice(x,rice), type="b")


# For SNR=v/a > 24 or 240.001/10 > 24, the Normal distribution is
# used by the Rice as implemented here.
rice1 <- vec2par(c(239.9999,10), type="rice")
rice2 <- vec2par(c(240.0001,10), type="rice")
x <- 200:280
plot( x, pdfrice(x, rice1), type="l", lwd=5, lty=3) # still RICIAN code
lines(x, dnorm(  x, mean=240.0001, sd=10), lwd=3, col=2) # NORMAL obviously
lines(x, pdfrice(x, rice2), lwd=1, col=3) # NORMAL distribution code is triggered

# For SNR=v/a > 52 or 521/10 > 52, the Normal distribution
# used by the Rice as implemented here with simple parameter estimation
# because this high of SNR is beyond limits of Bessel function in Laguerre
# polynomial
rice1 <- vec2par(c(520,10), type="rice")
rice2 <- vec2par(c(521,10), type="rice")
x <- 10^(log10(520) - 0.05):10^(log10(520) + 0.05)
plot( x, pdfrice(x, rice1), type="l", lwd=5, lty=3)
lines(x, pdfrice(x, rice2), lwd=1, col=3) # NORMAL code triggered
```

---

pdfsla *Probability Density Function of the Slash Distribution*

Description

This function computes the probability density of the Slash distribution given parameters (ξ and α) provided by parsla. The probability density function is

$$f(x) = \frac{\phi(0) - \phi(y)}{y^2},$$

where $f(x)$ is the probability density for quantile x, $y = (x - \xi)/\alpha$, ξ is a location parameter, and α is a scale parameter. The function $\phi(y)$ is the probability density function of the Standard Normal distribution.

Usage

```
pdfsla(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parsla or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Rogers, W.H., and Tukey, J.W., 1972, Understanding some long-tailed symmetrical distributions: Statistica Neerlandica, v. 26, no. 3, pp. 211–226.

See Also

cdfsla, quasla, lmomsla, parsla

Examples

```
sla <- vec2par(c(12, 1.2), type="sla")
x <- quasla(0.5, sla)
pdfsla(x, sla)
```

---

pdfsmd *Probability Density Function of the Singh–Maddala Distribution*

Description

This function computes the probability density of the Singh–Maddala (Burr Type XII) distribution given parameters (a, b, and q) computed by parsmd. The probability density function is

$$f(x) = \frac{b \cdot q \cdot x^{b-1}}{a^b \biggl(1 + \bigl[(x - \xi)/a\bigr]^b\biggr)^{q+1}},$$

where $f(x)$ is the probability density for quantile x with $0 \le x \le \infty$, ξ is a location parameter, a is a scale parameter ($a > 0$), b is a shape parameter ($b > 0$), and q is another shape parameter ($q > 0$).

Usage

```
pdfsmd(x, para)
```

Arguments

x	A real value vector.

para	The parameters from parsmd or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Kumar, D., 2017, The Singh–Maddala distribution—Properties and estimation: International Journal of System Assurance Engineering and Management, v. 8, no. S2, 15 p., doi:10.1007/s13198-01706001.

Shahzad, M.N., and Zahid, A., 2013, Parameter estimation of Singh Maddala distribution by moments: International Journal of Advanced Statistics and Probability, v. 1, no. 3, pp. 121–131, doi:10.14419/ijasp.v1i3.1206.

See Also

cdfsmd, quasmd, lmomsmd, parsmd

Examples

```
# The SMD approximating the normal and use x=0
tau4_of_normal <- 30 * pi^-1 * atan(sqrt(2)) - 9 # from theory
pdfsmd(0, parsmd( vec2lmom( c( -pi, pi, 0, tau4_of_normal ) ) ) ) # 0.061953
dnorm( 0, mean=-pi, sd=pi*sqrt(pi))                               # 0.06110337

## Not run:
LMlo <- vec2lmom(c(10000, 1500, 0.3, 0.1))
LMhi <- vec2lmom(c(10000, 1500, 0.3, 0.6))
SMDlo <- parsmd(LMlo, snap.tau4=TRUE) # Tau4 snapped to 0.15077
SMDhi <- parsmd(LMhi, snap.tau4=TRUE) # Tau4 snapped to 0.25360
FF <- pnorm(seq(-6, 3, by=.01))
x <- sort(c(quasmd(FF, SMDlo), quasmd(FF, SMDhi)))
plot( x, pdfsmd(x, SMDlo), col="red", xlim=range(x), type="l")
lines(x, pdfsmd(x, SMDhi), col="blue") #
## End(Not run)
```

pdfst3 *Probability Density Function of the 3-Parameter Student t Distribution*

Description

This function computes the probability density of the 3-parameter Student t distribution given parameters (ξ, α, ν) computed by parst3. The probability density function is

$$f(x) = \frac{\Gamma(\frac{1}{2} + \frac{1}{2}\nu)}{\alpha\nu^{1/2}\,\Gamma(\frac{1}{2})\Gamma(\frac{1}{2}\nu)}(1 + t^2/\nu)^{-(\nu+1)/2},$$

where $f(x)$ is the probability density for quantile x, t is defined as $t = (x - \xi)/\alpha$, ξ is a location parameter, α is a scale parameter, and ν is a shape parameter in terms of the degrees of freedom as for the more familiar Student t distribution in R.

For value X, the built-in R functions can be used. For U = ξ and A=α for $1.001 \le \nu \le 10^5.5$, one can use dt((X-U)/A, N)/A for N=ν. The R function dt is used for the 1-parameter Student t density. The limits for ν stem from study of ability for theoretical integration of the quantile function to produce viable τ_4 and τ_6 (see inst/doc/t4t6/studyST3.R).

Usage

```
pdfst3(x, para, paracheck=TRUE)
```

Arguments

x	A real value vector.
para	The parameters from parst3 or vec2par.
paracheck	A logical on whether the parameter should be check for validity.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

cdfst3, quast3, lmomst3, parst3

Examples

```
## Not run:
xs <- -200:200
  para <- vec2par(c(37, 25,  114), type="st3")
plot(xs, pdfst3(xs, para), type="l")
  para <- vec2par(c(11, 36, 1000), type="st3")
lines(xs, pdfst3(xs, para), lty=2)
  para <- vec2par(c(-7, 60,   40), type="st3")
lines(xs, pdfst3(xs, para), lty=3)

## End(Not run)
```

---

pdftexp *Probability Density Function of the Truncated Exponential Distribution*

Description

This function computes the probability density of the Truncated Exponential distribution given parameters (ψ and α) computed by partexp. The parameter ψ is the right truncation, and α is a scale parameter. The probability density function, letting $\beta = 1/\alpha$ to match nomenclature of Vogel and others (2008), is

$$f(x) = \frac{\beta\ \exp(-\beta t)}{1 - \exp(-\beta\psi)},$$

where $x(x)$ is the probability density for the quantile $0 \leq x \leq \psi$ and $\psi > 0$ and $\alpha > 0$. This distribution represents a nonstationary Poisson process.

The distribution is restricted to a narrow range of L-CV ($\tau_2 = \lambda_2/\lambda_1$). If $\tau_2 = 1/3$, the process represented is a stationary Poisson for which the probability density function is simply the uniform distribution and $f(x) = 1/\psi$. If $\tau_2 = 1/2$, then the distribution is represented as the usual exponential distribution with a location parameter of zero and a scale parameter $1/\beta$. Both of these limiting conditions are supported.

Usage

```
pdftexp(x, para)
```

Arguments

x	A real value vector.
para	The parameters from partexp or vec2par.

Value

Probability density (F) for x.

Author(s)

W.H. Asquith

References

Vogel, R.M., Hosking, J.R.M., Elphick, C.S., Roberts, D.L., and Reed, J.M., 2008, Goodness of fit of probability distributions for sightings as species approach extinction: Bulletin of Mathematical Biology, DOI 10.1007/s11538-008-9377-3, 19 p.

See Also

cdftexp, quatexp, lmomtexp, partexp

Examples

```
lmr <- vec2lmom(c(40,0.38), lscale=FALSE)
pdftexp(0.5,partexp(lmr))
## Not run:
F <- seq(0,1,by=0.001)
A <- partexp(vec2lmom(c(100, 1/2), lscale=FALSE))
x <- quatexp(F, A)
plot(x, pdftexp(x, A), pch=16, type='l')
by <- 0.01; lcvs <- c(1/3, seq(1/3+by, 1/2-by, by=by), 1/2)
reds <- (lcvs - 1/3)/max(lcvs - 1/3)
for(lcv in lcvs) {
    A <- partexp(vec2lmom(c(100, lcv), lscale=FALSE))
    x <- quatexp(F, A)
    lines(x, pdftexp(x, A),
          pch=16, col=rgb(reds[lcvs == lcv],0,0))
}

## End(Not run)
```

---

pdftri — *Probability Density Function of the Asymmetric Triangular Distribution*

Description

This function computes the probability density of the Asymmetric Triangular distribution given parameters (ν, ω, and ψ) computed by partri. The probability density function is

$$f(x) = \frac{2(x-\nu)}{(\omega-\nu)(\psi-\nu)},$$

for $x < \omega$,

$$f(x) = \frac{2(\psi-x)}{(\psi-\omega)(\psi-\nu)},$$

for $x > \omega$, and

$$f(x) = \frac{2}{(\psi-\nu)},$$

for $x = \omega$ where $x(F)$ is the quantile for nonexceedance probability F, ν is the minimum, ψ is the maximum, and ω is the mode of the distribution.

Usage

```
pdftri(x, para)
```

Arguments

x	A real value vector.
para	The parameters from partri or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

See Also

pdftri, quatri, lmomtri, partri

Examples

```
tri <- vec2par(c(-120, 102, 320), type="tri")
x <- quatri(nonexceeds(),tri)
pdftri(x,tri)
```

---

pdfwak — *Probability Density Function of the Wakeby Distribution*

Description

This function computes the probability density of the Wakeby distribution given parameters (ξ, α, β, γ, and δ) computed by parwak. The probability density function is

$$f(x) = (\alpha[1 - F(x)]^{\beta - 1} + \gamma[1 - F(x)]^{-\delta - 1})^{-1},$$

where $f(x)$ is the probability density for quantile x, $F(x)$ is the cumulative distribution function or nonexceedance probability at x, ξ is a location parameter, α and β are scale parameters, and γ, and δ are shape parameters. The five returned parameters from parwak in order are ξ, α, β, γ, and δ.

Usage

```
pdfwak(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parwak or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

Sourced from written communication with Dr. Hosking in October 2007.

See Also

cdfwak, quawak, lmomwak, parwak

Examples

```
## Not run:
lmr <- vec2lmom(c(1,0.5,.4,.3,.15))
wak <- parwak(lmr)
F <- nonexceeds()
x <- quawak(F,wak)
check.pdf(pdfwak,wak,plot=TRUE)

## End(Not run)
```

---

pdfwei *Probability Density Function of the Weibull Distribution*

Description

This function computes the probability density of the Weibull distribution given parameters (ζ, β, and δ) computed by parwei. The probability density function is

$$f(x) = \delta Y^{\delta-1} \exp(-Y^{\delta})/\beta$$

where $f(x)$ is the probability density, $Y = (x - \zeta)/\beta$, quantile x, ζ is a location parameter, β is a scale parameter, and δ is a shape parameter.

The Weibull distribution is a reverse Generalized Extreme Value distribution. As result, the Generalized Extreme Value algorithms are used for implementation of the Weibull in **lmomco**. The relations between the Generalized Extreme Value parameters (ξ, α, and κ) are $\kappa = 1/\delta$, $\alpha = \beta/\delta$, and $\xi = \zeta - \beta$. These relations are available in Hosking and Wallis (1997).

In R, the probability distribution function of the Weibull distribution is `pweibull`. Given a Weibull parameter object `para`, the R syntax is `pweibull(x+para$para[1], para$para[3], scale=para$para[2])`. For the **lmomco** implmentation, the reversed Generalized Extreme Value distribution pdfgev is used and again in R syntax is `pdfgev(-x,para)`.

Usage

```
pdfwei(x, para)
```

Arguments

x	A real value vector.
para	The parameters from parwei or vec2par.

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Hosking, J.R.M. and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfwei, quawei, lmomwei, parwei

Examples

```
  # Evaluate Weibull deployed here and built-in function (pweibull)
  lmr <- lmoms(c(123,34,4,654,37,78))
  WEI <- parwei(lmr)
  F1  <- cdfwei(50,WEI)
  F2  <- pweibull(50+WEI$para[1],shape=WEI$para[3],scale=WEI$para[2])
  if(F1 == F2) EQUAL <- TRUE
## Not run:
  # The Weibull is a reversed generalized extreme value
  Q <- sort(rlmomco(34,WEI)) # generate Weibull sample
  lm1 <- lmoms( Q)   # regular L-moments
  lm2 <- lmoms(-Q)   # L-moment of negated (reversed) data
  WEI <- parwei(lm1) # parameters of Weibull
  GEV <- pargev(lm2) # parameters of GEV
  F <- nonexceeds()  # Get a vector of nonexceedance probabilities
  plot(pp(Q),Q)
  lines(cdfwei(Q,WEI),Q,lwd=5,col=8)
  lines(1-cdfgev(-Q,GEV),Q,col=2) # line overlaps previous distribution

## End(Not run)
```

pfactor.bernstein *Estimation of Optimal p-factor of Distributional Support Estimation for Smoothed Quantiles from the Bernstein or Kantorovich Polynomials*

Description

Compute the optimal p-factor through numerical integration of the smoothed empirical quantile function to estimate the L-moments of the distribution. This function attempts to report an optimal "p-factor" (author's term) for the given parent distribution in `para` based on estimating the crossing of the origin of an error between the given L-moment ratio τ_r for 3, 4, and 5 that will come from either the distribution parameter object or given as an argument in `lmr.dist`. The estimated support of the distribution is that shown by Turnbull and Ghosh (2014) and is computed as follows

$$\biggl(x_{0:n},\, x_{n+1:n}\biggr) = \biggl(x_{1:n} - \frac{(x_{2:n} - x_{1:n})}{(1-p)^{-2} - 1},\, x_{n:n} + \frac{(x_{n:n} - x_{n-1:n})}{(1-p)^{-2} - 1}\biggr),$$

where p is the p-factor. The support will honor natural bounds if given by either `fix.lower` or `fix.upper`. The polynomial type for smooth is provided in `poly.type`. These three arguments are the same as those for `dat2bernqua` and `lmoms.bernstein`. The statistic type used to measure central tendency of the errors for the `nsim` simulations per p. The function has its own hardwired p-factors to compute but these can be superseded by the `pfactors` argument. The `p.lo` and `p.hi` are the lower and upper bounds to truncate on immediately after the p-factors to use are assembled. These are made for three purposes: (1) protection against numerical problems for mathematical upper limits (unity), (2) to potentially provide for much faster execution if the user already knows the approximate optimal value for the p-factor, and (3) to potentially use this function in a direct optimization framework using the R functions `optim` or `uniroot`. It is strongly suggested to keep `plot.em` set so the user can inspect the computations.

Usage

```
pfactor.bernstein(para, x=NULL, n=NULL,
                        bern.control=NULL,
                        poly.type=c("Bernstein", "Kantorovich"),
                        stat.type=c("Mean", "Median"),
                        fix.lower=NULL, fix.upper=NULL,
                        lmr.dist=NULL, lmr.n=c("3", "4", "5"),
                        nsim=500, plot.em=TRUE, pfactors=NULL,
                        p.lo=.Machine$double.eps, p.hi=1)
```

Arguments

`para` A mandatory "parent" distribution defined by a usual **lmomco** distribution parameter object for a distribution. The simulations are based on this distribution, although optimization for p can be set to a different L-moment value by `lmr.dist`.

`x` An optional vector of data values.

n
: An optional sample size to run the simulations on. This value is computed by `length(x)` if `x` is provided. If set by argument, then that size supersedes the length of the optional observed sample.

bern.control
: A `list` that holds `poly.type`, `stat.type`, `fix.lower`, and `fix.upper`. And this list will supersede the respective values provided as separate arguments. There is an implicit `bound.type` of `"Carv"`.

poly.type
: Same argument as for `dat2bernqua`.

stat.type
: The central estimation statistic for each p-factor evaluated.

fix.lower
: Same argument as for `dat2bernqua`.

fix.upper
: Same argument as for `dat2bernqua`.

lmr.dist
: This is the value for the `lmr.n` of the distribution in `para` unless explicitly set through `lmr.dist`.

lmr.n
: The L-moment ratio number for p-factor optimization.

nsim
: The number of simulations to run. Experiments suggest the default is adequate for reasonably small sample sizes—the simulation count can be reduced as `n` becomes large.

plot.em
: A logical to trigger the diagnostic plot of the simulated errors and a smooth line through these errors.

pfactors
: An optional vector of p-factors to loop through for the simulations. The vector computing internall is this is set to `NULL` seems to be more than adequate.

p.lo
: An computational lower boundary for which the `pfactors` by argument or default are truncated to. The default for `lo` is to be quite small and does no truncate the default `pfactors`.

p.hi
: An computational upper boundary for which the `pfactors` by argument or default are truncated to. The default for `hi` is unity, which is the true upper limit that results in a 0 slope between the $x_{0:n}$ to $x_{1:n}$ or $x_{n:n}$ to $x_{n+1:n}$ order statistics.

Value

An R `list` or `real` is returned. If `pfactors` is a single value, then the single value for the error statistic is returned, otherwise the list described will be. If the returned `pfactor` is `NA`, then likely the smooth line did not cross zero and the reason the user should keep `plot.em=TRUE` and inspect the plot. Perhaps revisions to the arguments will become evident. The contents of the list are

pfactor
: The estimated value of p smoothed by `lowess` that has an error of zero, see `err.stat` as a function of `ps`.

bounds.type
: `Carv`, which is the same bound type as needed by `dat2bernqua` and `lmoms.bernstein`.

poly.type
: The given `poly.type`.

stat.type
: The given `stat.type`. The “Mean” seems to be preferable.

lmom.type
: A string of the L-moment type: “Tau3”, “Tau4”, “Tau5”.

fix.lower
: The given fixed lower boundary, which could stay `NULL`.

fix.upper
: The given fixed upper boundary, which could stay `NULL`.

source	An attribute identifying the computational source of the L-moments: "pfactor.bernstein".
ps	The p-factors actually evaluated.
err.stat	The error statistic computed by stat.type of the simulated $\hat{\tau}_r$ by integration provided by lmoms.bernstein minus the "true" value τ_r provided by either para or given by lmr.dist where r is lmr.n.
err.smooth	The lowess-smoothed values for err.stat and the pfactor comes from a linear interpolation of this smooth for the error being zero.

Note

Repeated application of this function for various n would result in the analyst having a vector of n and p (pfactor). The analyst could then fit a regression equation and refine the estimated $p(n)$. For example, a dual-logarithmic regression is suggested lm(log(p)~log(n)).

Also, symmetrical data likely see little benefit from optimizing on the symmetry-measuring L-moments Tau3 and Tau5; the analyst might prefer to optimize on peakedness measured by Tau4.

Note

This function is highly experimental and subject to extreme overhaul. Please contact the author if you are an interested party in Bernstein and Kantorovich polynomials.

Author(s)

W.H. Asquith

References

Turnbull, B.C., and Ghosh, S.K., 2014, Unimodal density estimation using Bernstein polynomials. Computational Statistics and Data Analysis, v. 72, pp. 13–29.

See Also

lmoms.bernstein, dat2bernqua, lmoms

Examples

```
## Not run:
pdf("pfactor_exampleB.pdf")
X <- exp(rnorm(200)); para <- parexp(lmoms(X))
# nsim is too small, but makes the following three not take too long
pfactor.bernstein(para, n=20, lmr.n="3", nsim=100, p.lo=.06, p.hi=.3)
pfactor.bernstein(para, n=20, lmr.n="4", nsim=100, p.lo=.06, p.hi=.3)
pfactor.bernstein(para, n=20, lmr.n="5", nsim=100, p.lo=.06, p.hi=.3)
dev.off()

## End(Not run)
## Not run:
# Try intra-sample p-factor optimization from two perspectives. The 3-parameter
# GEV "over fits" the data and provides the parent.  Then use Tau3 of the fitted
```

```
# GEV for peakedness restraint and then use Tau3 of the data. Then repeat but use
# the apparent "exact" value of Tau3 for the true exponential parent.
pdf("pfactor_exampleB.pdf")
lmr <- vec2lmom(c(60,20)); paraA <- parexp(lmr); n <- 40
tr <- lmorph(par2lmom(paraA))$ratios[3]
X <- rlmomco(n, paraA); para <- pargev(lmoms(X))
F <- seq(0.001,0.999, by=0.001)
plot(qnorm(pp(X, a=0.40)), sort(X), type="n", log="y",
      xlab="Standard normal variate", ylab="Quantile",
      xlim=qnorm(range(F)), ylim=range(qlmomco(F,paraA)))
lines(qnorm(F), qlmomco(F, paraA), col=8, lwd=2)
lines(qnorm(F), qlmomco(F, para), lty=2)
points(qnorm(pp(X, a=0.40)), sort(X))

# Make sure to fill in the p-factor when needed!
bc <- list(poly.type = "Bernstein", bound.type="Carv",
           stat.type="Mean", fix.lower=0, fix.upper=NULL, p=NULL)
kc <- list(poly.type = "Kantorovich", bound.type="Carv",
           stat.type="Mean", fix.lower=0, fix.upper=NULL, p=NULL)

# Bernstein
A <- pfactor.bernstein(para,      n=n, nsim=100,              bern.control=bc)
B <- pfactor.bernstein(para, x=X, n=n, nsim=100,              bern.control=bc)
C <- pfactor.bernstein(para,      n=n, nsim=100, lmr.dist=tr, bern.control=bc)
D <- pfactor.bernstein(para, x=X, n=n, nsim=100, lmr.dist=tr, bern.control=bc)
plot(qnorm(pp(X, a=0.40)), sort(X), type="n", log="y",
      xlab="Standard normal variate", ylab="Quantile",
      xlim=qnorm(range(F)), ylim=range(qlmomco(F,paraA)))
lines(qnorm(F), qlmomco(F, paraA), col=8, lwd=2)
lines(qnorm(F), qlmomco(F, para), lty=2)
points(qnorm(pp(X, a=0.40)), sort(X))
      bc$p <- A$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=bc), col=2)
      bc$p <- B$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=bc), col=3)
      bc$p <- C$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=bc), col=2, lty=2)
      bc$p <- D$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=bc), col=3, lty=2)
# Kantorovich
A <- pfactor.bernstein(para,      n=n, nsim=100,              bern.control=kc)
B <- pfactor.bernstein(para, x=X, n=n, nsim=100,              bern.control=kc)
C <- pfactor.bernstein(para,      n=n, nsim=100, lmr.dist=tr, bern.control=kc)
D <- pfactor.bernstein(para, x=X, n=n, nsim=100, lmr.dist=tr, bern.control=kc)
plot(qnorm(pp(X, a=0.40)), sort(X), type="n", log="y",
      xlab="Standard normal variate", ylab="Quantile",
      xlim=qnorm(range(F)), ylim=range(qlmomco(F,paraA)))
lines(qnorm(F), qlmomco(F, paraA), col=8, lwd=2)
lines(qnorm(F), qlmomco(F, para), lty=2)
points(qnorm(pp(X, a=0.40)), sort(X))
      kc$p <- A$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=kc), col=2)
      kc$p <- B$pfactor
```

```
lines(qnorm(F), dat2bernqua(F,X, bern.control=kc), col=3)
      kc$p <- C$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=kc), col=2, lty=2)
      kc$p <- D$pfactor
lines(qnorm(F), dat2bernqua(F,X, bern.control=kc), col=3, lty=2)
dev.off()

## End(Not run)
## Not run:
X <- exp(rnorm(200)); para <- parexp(lmoms(X))
"pfactor.root" <- function(para, p.lo, p.hi, ...) {
    afunc <- function(p, para=NULL, x=NULL, ...) {
      return(pfactor.bernstein(para=para, x=x, pfactors=p, ...)) }
    rt <- uniroot(afunc, c(p.lo, p.hi),
                  tol=0.001, maxiter=30, nsim=500, para=para, ...)
    return(rt)
}
pfactor.root(para, 0.05, 0.15, n=10, lmr.n="4")
pfactor.bernstein(para, n=10, lmr.n="4", nsim=200, p.lo=.05, p.hi=.15)

## End(Not run)
```

---

plmomco — *Cumulative Distribution Function of the Distributions*

Description

This function acts as an alternative front end to par2cdf. The nomenclature of the plmomco function is to mimic that of built-in R functions that interface with distributions.

Usage

```
plmomco(x, para)
```

Arguments

x	A real value.
para	The parameters from lmom2par or similar.

Value

Nonexceedance probability ($0 \leq F \leq 1$) for x.

Author(s)

W.H. Asquith

See Also

dlmomco, qlmomco, rlmomco, slmomco, add.lmomco.axis

Examples

```
para <- vec2par(c(0,1),type='nor') # Standard Normal parameters
nonexceed <- plmomco(1,para) # percentile of one standard deviation
```

---

plotlmrdia *Plot L-moment Ratio Diagram (Tau3 and Tau4)*

Description

Plot the Tau3-Tau4 L-moment ratio diagram of L-skew and L-kurtosis from a Tau3-Tau4 L-moment ratio diagram object returned by `lmrdia`. This diagram is useful for selecting a distribution to model the data. The application of L-moment diagrams is well documented in the literature. This function is intended to function as a demonstration of L-moment ratio diagram plotting with enough user settings for many practical applications.

Usage

```
plotlmrdia(lmr=NULL, nopoints=FALSE, nolines=FALSE, nolimits=FALSE,
           noaep4=FALSE, nogev=FALSE, noglo=FALSE,  nogno=FALSE, nogov=FALSE,
           nogpa=FALSE,  nope3=FALSE, nopdq3=FALSE, nowei=TRUE,
           nocau=TRUE,   noexp=FALSE, nonor=FALSE,  nogum=FALSE,
           noray=FALSE, nosla=TRUE, nouni=FALSE,
           xlab="L-skew (Tau3), dimensionless",
           ylab="L-kurtosis (Tau4), dimensionless", add=FALSE, empty=FALSE,
           autolegend=FALSE, xleg=NULL, yleg=NULL, legendcex=0.9,
           ncol=1, text.width=NULL, lwd.cex=1, expand.names=FALSE, ...)
```

Arguments

`lmr`	L-moment diagram object from `lmrdia`, if `NULL`, then `empty` is internally set to `TRUE`.
`nopoints`	If `TRUE` then point distributions are not drawn.
`nolines`	If `TRUE` then line distributions are not drawn.
`nolimits`	If `TRUE` then theoretical limits of L-moments are not drawn.
`noaep4`	If `TRUE` then the lower bounds line of Asymmetric Exponential Power distribution is not drawn.
`nogev`	If `TRUE` then line of Generalized Extreme Value distribution is not drawn.
`noglo`	If `TRUE` then line of Generalized Logistic distribution is not drawn.
`nogno`	If `TRUE` then line of Generalized Normal (Log-Normal3) distribution is not drawn.
`nogov`	If `TRUE` then line of Govindarajulu distribution is not drawn.
`nogpa`	If `TRUE` then line of Generalized Pareto distribution is not drawn.
`nope3`	If `TRUE` then line of Pearson Type III distribution is not drawn.
`nopdq3`	If `TRUE` then line of Polynomial Density-Quantile3 distribution is not drawn.

`nowei`	If `TRUE` then line of the Weibull distribution is not drawn. The Weibull is a reverse of the Generalized Extreme Value. Traditionally in the literature, the Tau3-Tau4 L-moment ratio diagram have usually included the Weibull distribution and therefore the default setting of this argument is to not plot the Weibull.
`nocau`	If `TRUE` then point (TL-moment [trim=1]) of the Cauchy distribution is not drawn.
`noexp`	If `TRUE` then point of Exponential distribution is not drawn.
`nonor`	If `TRUE` then point of Normal distribution is not drawn.
`nogum`	If `TRUE` then point of Gumbel distribution is not drawn.
`noray`	If `TRUE` then point of Rayleigh distribution is not drawn.
`nouni`	If `TRUE` then point of Uniform distribution is not drawn.
`nosla`	If `TRUE` then point (TL-moment [trim=1]) of the Slash distribution is not drawn.
`xlab`	Horizonal axis label passed to `xlab` of the `plot` function.
`ylab`	Vertical axis label passed to `ylab` of the `plot` function.
`add`	A logical to toggle a call to `plot` to start a new plot, otherwise, just the trajectories are otherwise plotted.
`empty`	A logical to return before any trajectories are plotted but after the condition of the `add` has been evaluated.
`autolegend`	Generate the legend by built-in algorithm.
`xleg`	X-coordinate of the legend. This argument is checked for being a character versus a numeric. If it is a character, then `yleg` is not needed and `xleg` and take on “location may also be specified by setting x to a single keyword” as per the functionality of `graphics::legend()` itself.
`yleg`	Y-coordinate of the legend.
`legendcex`	The `cex` to pass to `graphics::legend()`.
`ncol`	The number of columns in which to set the legend items (default is 1, which differs from `legend()` default of 1).
`text.width`	Argument of the same name for `legend`. Setting to 0.1 for `ncol` set to 2 seems to work pretty well when two columns are desired.
`lwd.cex`	Expansion factor on the line widths.
`expand.names`	Expand the distribution names in the legend.
`...`	Additional arguments passed into `plot()` and `legend()` functions..

Note

This function provides hardwired calls to `lines` and `points` to produce the diagram. The plot symbology for the shown distributions is summarized here. The Asymmetric Exponential Power and Kappa (four parameter) and Wakeby (five parameter) distributions are not well represented on the diagram as each constitute an area (Kappa) or hyperplane (Wakeby) and not a line (3-parameter distributions) or a point (2-parameter distributions). However, the Kappa demarks the area bounded by the Generalized Logistic (`glo`) on the top and the theoretical L-moment limits on the bottom. The Asymmetric Exponential Power demarks its own unique lower boundary and extends up in the τ_4 direction to $\tau_4 = 1$. However, parameter estimation with L-moments has lost considerable accuracy for τ_4 that large (see Asquith, 2014).

GRAPHIC TYPE	GRAPHIC NATURE
L-moment Limits	line width 2 and color a medium-dark grey
Asymmetric Exponential Power (4-p)	line width 1, line type 4 (dot), and color red
Generalized Extreme Value (GEV)	line width 1, line type 1 (solid), and color darkred
Generalized Logistic	line width 1 and color green
Generalized Normal	line width 1, line type 2 (dash), and color blue
Govindarajulu	line width 1, line type 2 (dash), and color 6 (magenta)
Generalized Pareto	line width 1, line type 1 (solid), and color blue
Pearson Type III	line width 1, line type 1 (solid), and color 6 (purple)
Polynomial Density-Quantile3	line width 1.3, line type 2 (dash), and color darkgreen
Weibull (reversed GEV)	line width 1, line type 1 (solid), and color darkorange
Exponential	symbol 16 (filled circle) and color red
Normal	symbol 15 (filled square) and color red
Gumbel	symbol 17 (filled triangle) and color red)
Rayleigh	symbol 18 (filled diamond) and color red
Uniform	symbol 12 (square and a plus sign) and color red
Cauchy	symbol 13 (circle with over lapping $\times$) and color turquoise4
Slash	symbol 10 (cicle containing $+$) and color turquoise4

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis–An approach based on L-moments: Cambridge University Press.

Vogel, R.M., and Fennessey, N.M., 1993, L moment diagrams should replace product moment diagrams: Water Resources Research, v. 29, no. 6, pp. 1745–1752.

See Also

`lmrdia`, `plotlmrdia46`, `plotradarlmr`

Examples

```
plotlmrdia(lmrdia()) # simplest of all uses

## Not run:
# A more complex example follows: for a given mean, L-scale, L-skew, and L-kurtosis,
# use sample size of 30, use 500 simulations, set L-moments, fit the Kappa distribution
T3 <- 0.34; T4 <- 0.21; n <- 30; nsim <- 500
lmr <- vec2lmom(c(10000, 7500, T3, T4)); kap <- parkap(lmr)

# create vectors for storing simulated L-skew (t3) and L-kurtosis (t4)
t3 <- t4 <- vector(mode="numeric")

# perform nsim simulations by randomly drawing from the Kappa distribution
# and compute the L-moments in sim.lmr and store the t3 and t4 of each sample
for(i in 1:nsim) {
  sim.lmr <- lmoms(rlmomco(n, kap))
  t3[i] <- sim.lmr$ratios[3]; t4[i] <- sim.lmr$ratios[4]
}

# plot the diagram and "zoom" by manually setting the axis limits
plotlmrdia(xlim=c(-0.1, 0.5), ylim=c(-0.1, 0.4), las=1, empty=TRUE)

# Follow up by plotting the {t3, t4} values and the mean of the values
points(t3, t4, pch=21, bg="white", lwd=0.8) # plot each simulation

# plot crossing dashed lines at true values of L-skew and L-kurtosis
abline(v=T3, col="salmon4", lty=2, lwd=3) # Theoretical values for the
abline(h=T4, col="salmon4", lty=2, lwd=3) # distribution as fit

points(mean(t3), mean(t4), pch=16, cex=3) # mean of simulations and
# should plot reasonably close to the salmon4-colored crossing lines

# plot the trajectories of the distributions
plotlmrdia(lmrdia(), add=TRUE, nopoints=TRUE, inset=0.01,
           autolegend=TRUE, xleg="topleft", lwd.cex=1.5) #
## End(Not run)
```

---

plotlmrdia46 *Plot L-moment Ratio Diagram (Tau4 and Tau6)*

Description

Plot the Tau4-Tau6 L-moment ratio diagram showing trajectories of τ_4 and τ_6 for strictly symmetrical distributions from a Tau4-Tau6 L-moment ratio diagram object returned by lmrdia46. This diagram is useful for selecting among symmetrical distributions to model the data. This function is intended to function as a demonstration of Tau4-Tau6 L-moment ratio diagram plotting with enough user settings for many practical applications.

Usage

```
plotlmrdia46(lmr=NULL, nopoints=FALSE, nolines=FALSE,
             noaep4=FALSE,  nogld_byt5opt=TRUE, nopdq4=FALSE,  nost3=FALSE,
             nosymgdd=TRUE, nosymstable=FALSE,  notukey=FALSE,
            nocau=TRUE,    nonor=FALSE, nosla=TRUE, trucate.tau4.to.gtzero=TRUE,
             xlab="L-kurtosis (Tau4), dimensionless",
             ylab="Sixth L-moment ratio (Tau6), dimensionless",
             add=FALSE, empty=FALSE,
             autolegend=FALSE, xleg=NULL, yleg=NULL, legendcex=0.9,
             ncol=1, text.width=NULL, lwd.cex=1, expand.names=FALSE, ...)
```

Arguments

`lmr`	L-moment diagram object from `lmrdia46`, if `NULL`, then `empty` is internally set to `TRUE`.
`nopoints`	If `TRUE` then point distributions are not drawn.
`nolines`	If `TRUE` then line distributions are not drawn.
`noaep4`	If `TRUE` then the Symmetric Exponential Power distribution is not drawn.
`nogld_byt5opt`	If `TRUE` then line of Generalized Lambda distribution through it solution optimization on $\tau_5 = 0$ is not drawn.
`nopdq4`	If `TRUE` then line of Polynomial Density-Quantile4 distribution is not drawn.
`nost3`	If `TRUE` then line of Student 3t distribution is not drawn.
`nosymgdd`	If `TRUE` then line of a symmetrical Gamma Difference distribution is not drawn.
`nosymstable`	If `TRUE` then line of Symmetric Stable distribution is not drawn.
`notukey`	If `TRUE` then line of Tukey Lambda distribution is not drawn.
`nocau`	If `TRUE` then point of Cauchy distribution (trim=1 L-moments) is not drawn.
`nonor`	If `TRUE` then point of Normal distribution is not drawn.
`nosla`	If `TRUE` then point of Slash distribution (trim=1 L-moments) is not drawn.
`trucate.tau4.to.gtzero`	Truncate the distributions that can extend to negative τ_4 to zero. This is a reasonable default and prevents line drawing to the left into a clipping region for easier handling of post processing of a graphic in vector editing software.
`xlab`	Horizonal axis label passed to `xlab` of the `plot` function.
`ylab`	Vertical axis label passed to `ylab` of the `plot` function.
`add`	A logical to toggle a call to `plot` to start a new plot, otherwise, just the trajectories are otherwise plotted.
`empty`	A logical to return before any trajectories are plotted but after the condition of the `add` has been evaluated.
`autolegend`	Generate the legend by built-in algorithm.
`xleg`	X-coordinate of the legend. This argument is checked for being a character versus a numeric. If it is a character, then `yleg` is not needed and `xleg` and take on "location may also be specified by setting `x` to a single keyword" as per the functionality of `graphics::legend()` itself.

yleg	Y-coordinate of the legend.
legendcex	The cex to pass to graphics::legend().
ncol	The number of columns in which to set the legend items (default is 1, which differs from legend() default of 1).
text.width	Argument of the same name for legend. Setting to 0.1 for ncol set to 2 seems to work pretty well when two columns are desired.
lwd.cex	Expansion factor on the line widths.
expand.names	Expand the distribution names in the legend.
...	Additional arguments passed into the plot() and legend() functions.

Note

This function provides hardwired calls to lines and points to produce the diagram. The plot symbology for the shown distributions is summarized here.

GRAPHIC TYPE	GRAPHIC NATURE
Symmetric Exponential Power	line width 1, line type 4 (dot), and color red
Generalized Lambda	line width 1, line type 1 (solid), and color purple
Polynomial Density-Quantile4	line width 1, line type 1 (solid), and color darkgreen
Student t	line width 1, line type 1 (solid), and color blue
Symmetric Gamma Difference	line width 2, line type 1 (solid), and color a darkorange2
Symmetric Stable	line width 2, line type 1 (solid), and color a medium-dark grey
Tukey Lambda (1-p)	line width 1, line type 2 (dash), and color purple
Normal	symbol 15 (filled square) and color red
Cauchy	symbol 13 (circle with over lapping $\times$) and color turquoise4
Slash	symbol 10 (cicle containing $+$) and color turquoise4

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

See Also

lmrdia46, plotlmrdia

Examples

```
plotlmrdia46(lmrdia46(), nogld_byt5opt=FALSE, nosymgdd=FALSE,
             autolegend=TRUE, xleg="topleft")

## Not run:
```

```
# A more complex example follows: for a given mean, L-scale, L-skew = 0 (symmetry), and
# L-kurtosis, use sample size of 30, use 500 simulations, set L-moments,
# fit the Asymmetric Exponential Power4 distribution, which is symmetrical when the
# L-skew is zero and thus the distribution is the Exponential Power.
T3  <- 0; T4 <- 0.21; n <- 30; nsim <- 500
lmr <- vec2lmom(c(10000, 7500, T3, T4, 0)); aep4 <- paraep4(lmr)
T6  <- theoLmoms(aep4, nmom=6)$ratios[6]

# create vectors for storing simulated L-kurtosis (t4) and Tau6 (t6)
t4 <- t6 <- vector(mode="numeric")

# perform nsim simulations by randomly drawing from the AEP4 distribution
# and compute the L-moments in sim.lmr and store the t4 and t6 of each sample
for(i in 1:nsim) {
  sim.lmr <- lmoms(rlmomco(n, aep4), nmom=6)
  t4[i] <- sim.lmr$ratios[4]; t6[i] <- sim.lmr$ratios[6]
}

# plot the diagram and "zoom" by manually setting the axis limits
plotlmrdia46(xlim=c(-0.05, 0.5), ylim=c(-0.1, 0.35), las=1, empty=TRUE)

# follow up by plotting the {t3, t4} values and the mean of the values
points(t4, t6, cex=0.8, pch=21, bg="white", lwd=0.8) # plot each simulation

# plot crossing dashed lines at true values of L-skew and L-kurtosis
abline(v=T4, col="salmon4", lty=2, lwd=3) # Theoretical values for the
abline(h=T6, col="salmon4", lty=2, lwd=3) # distribution as fit

points(mean(t4), mean(t6), pch=16, cex=3) # mean of simulations and
# should plot reasonably close to the salmon4-colored crossing lines

# plot the trajectories of the distributions
plotlmrdia46(lmrdia46(), add=TRUE, nopoints=TRUE, inset=0.01,
             autolegend=TRUE, xleg="topleft", lwd.cex=1.5) #
## End(Not run)
```

`plotradarlmr` *Plot L-moment Radar Plot (Chart) Graphic*

Description

Plot a L-moment radar plots (charts). This graphic is somewhat experimental and of unknown application benefit as no known precedent seems available. L-moment ratio diagrams (`plotlmrdia`) are incredibly useful but have generally been restricted to the 2-D domain. The graphic supported here attempts to provide a visualization of τ_r for an arbitrary $(r-2) > 3$ number of axes in the form of a radar plot. The angle of the axes is uninformative but the order of the axes is for τ_r for $r = 3, 4, \cdots$. The radar plot is essentially a line graph but mapped to a circular space at the expense of more *ink* being used. The radar plot is primarily intended to be a mechansim in **lmomco** for which similarity between other radar plots or presence of outlier combinations of τ_r can be judged when seen amongst various samples.

Usage

```
plotradarlmr(lmom, num.axis=4, plot=TRUE, points=FALSE, poly=TRUE, tag=NA,
             title="L-moment Ratio Radar Plot", make.zero.axis=FALSE,
             minrat=NULL, maxrat=NULL, theomins=TRUE, rot=0,
             labadj=1.2, lengthadj=1.75, offsetadj=0.25, scaleadj=2.2,
     axis.control  = list(col=1, lty=2, lwd=0.5, axis.cex=0.75, lab.cex=0.95),
     point.control = list(col=8, lwd=0.5, pch=16),
     poly.control  = list(col=rgb(0,0,0,.1), border=1, lty=1, lwd=1), ...)
```

Arguments

`lmom`	L-moment object such as from `lmoms`.
`num.axis`	The number of axes. Some error trapping in axis count relative to the length of the τ_r in `lmom` is made.
`plot`	A logical controlling whether R function `plot` will be called.
`points`	A logical controlling whether the points of defined by the τ_r in `lmom`.
`poly`	A logical controlling whether the polygon of defined by the τ_r in `lmom`.
`tag`	A text tag plotted at the center of the plot. An `NA` will result in nothing being plotted.
`title`	The title of the plot. An `NA` will result in nothing being plotted.
`make.zero.axis`	A logical controlling whether polygon will be "faked in" like as if τ_r having all zeros are provided. This feature is to act as a mechanism to overlay only the zero axis such as might be needed when a lot of other material has been already been drawn on the plot.
`minrat`	A vector of the minimum values for the τ_r axes in case the user desired to have some zoomability. The default is all -1 values, and a scalar for `minrat` will be repeated for the `num.axis`.
`maxrat`	A vector of the maximum values for the τ_r axes in case the user desired to have some zoomability. The default is all $+1$ values, and a scalar for `maxrat` will be repeated for the `num.axis`.
`theomins`	The are some basic and fundamental lower limits other than -1 that if used provide for a better relative scaling of the axes on the plot. If `TRUE`, then some select overwritting of potential user-provided `minrat` is provided.
`rot`	The basic rotational offset for the angle of the first (τ_3) axis.
`labadj`	An adjustment multiplier to help positions of the axis titles.
`lengthadj`	An adjustment multiplier characterize axis length.
`offsetadj`	An adjustment to help set the empty space in the middle of the plot for the `tag`.
`scaleadj`	An adjustment multiplier to help set the parent domain of the underlying (but hidden) x-y plot called by the R function `plot`.
`axis.control`	A specially built and not error trapped R `list` to hold the control elements of the axes.
`point.control`	A specially built and not error trapped R `list` to hold the control elements for plotting of the points if `points=TRUE`.

poly.control
: A specially built and not error trapped R `list` to hold the control elements for plotting of the polygon if `poly=TRUE`.

...
: Additional arguments passed on to the R function `text` function for the `title` and `tag`. This argument is largely not intended for general use, unlike most idioms of `...` in R, but is provided at the release of this function to help developers and avoid future backwards compatibility problems.

Note

This function has many implicit flexible features. The example below attempts to be reasonably comprehensive. Note that in the example that it is required to continue "knowing" what `minrat` and `maxrat` where used with `plot=TRUE`.

Author(s)

W.H. Asquith

See Also

plotlmrdia

Examples

```
## Not run:
plotradarlmr(NULL, minrat=-0.6, maxrat=0.6, tag="2 GEVs") # create the plot base
gev  <- vec2par(c(1230,123,-.24), type="gev") # set first parent distribution
poly <- list(col=NA, border=rgb(0,0,1,.1))    # set up polygon handling (blue)
for(i in 1:100) { # perform 100 simulations of the GEV with a sample of size 36
   plotradarlmr(lmoms(rlmomco(36,gev), nmom=6), plot=FALSE,
                poly.control=poly, minrat=-0.6, maxrat=0.6)
}
poly <- list(col=NA, border=4, lwd=3) # set up parent polygon
plotradarlmr(theoLmoms(gev, nmom=6), plot=FALSE,
             poly.control=poly, minrat=-0.6, maxrat=0.6) # draw the parent
 gev <- vec2par(c(450,1323,.5), type="gev") # set second parent distribution
poly <- list(col=NA, border=rgb(0,1,0,.1))  # set up polygon handling (green)
for(i in 1:100) { # perform 100 simulations of the GEV with a sample of size 36
   plotradarlmr(lmoms(rlmomco(36,gev), nmom=6), plot=FALSE,
                poly.control=poly, minrat=-0.6, maxrat=0.6) # draw the parent
}
poly <- list(col=NA, border=3, lwd=3) # set up parent polygon
plotradarlmr(theoLmoms(gev, nmom=6), plot=FALSE,
             poly.control=poly, minrat=-0.6, maxrat=0.6)
poly <- list(col=NA, border=6, lty=1, lwd=2) # make the zeros purple to standout.
plotradarlmr(NULL, make.zero.axis=TRUE, plot=FALSE,
             poly.control=poly, minrat=-0.6, maxrat=0.6) #
## End(Not run)
```

pmfben *Probability Density Function of the Benford Distribution*

Description

This function computes the probability mass function of the Benford distribution (Benford's Law) given parameters defining the number of first M-significant digits and the numeric base. The mass function has the simple expression

$$P(d) = \log_b\left(1 + \frac{1}{d}\right).$$

for any base $b \geq 2$ and digits d. The first significant digits in decimal are $d \in 1, \cdots, 9$, the first two-significant digits similarly are $d \in 10, \cdots, 99$, and the first three-significant digits similarly are $d \in 100, \cdots, 999$.

Usage

```
pmfben(d, para=list(para=c(1, 10)), ...)
```

Arguments

`d`	A integer value vector of M-significant digits.
`para`	The number of first M-significant digits followed by the numerical base (only base10 supported) and the list structure mimics similar uses of the **lmomco** list structure. Default are the first significant digits and hence the digits 1 through 9.
`...`	Additional arguments to pass (not likely to be needed but changes in base handling might need this).

Value

Probability density (f) for x.

Author(s)

W.H. Asquith

References

Benford, F., 1938, The law of anomalous numbers: Proceedings of the American Philosophical Society, v. 78, no. 4, pp. 551–572, https://www.jstor.org/stable/984802.

Goodman, W., 2016, The promises and pitfalls of Benford's law: Significance (Magazine), June 2015, pp. 38–41, doi:10.1111/j.17409713.2016.00919.x.

See Also

cdfben, quaben

Examples

```
# probability masses matching values in authoritative texts
pmfben(1:9, para=list(para=c(1, 10)))
# [1] 0.30103000 0.17609126 0.12493874 0.09691001
# [5] 0.07918125 0.06694679 0.05799195 0.05115252
# [9] 0.04575749
cumsum( pmfben(1:9, para=list(para=c(1, 10))) ) # should end in unity
# [1] 0.3010300 0.4771213 0.6020600 0.6989700 0.7781513
# [6] 0.8450980 0.9030900 0.9542425 1.0000000
```

---

pmoms — *The Sample Product Moments: Mean, Standard Deviation, Skew, and Excess Kurtosis*

Description

Compute the first four sample product moments. Both classical (theoretical and biased) versions and unbiased (nearly) versions are produced. Readers are directed to the References and the source code for implementation details.

Usage

```
pmoms(x)
```

Arguments

`x`	A real value vector.

Value

An R `list` is returned.

`moments`	Vector of the product moments: first element is the mean (`mean` in R), second is standard deviation, and the higher values typically are not used as these are not unbiased moments, but the ratios[3] and ratios[4] are nearly unbiased.
`ratios`	Vector of the product moment ratios. Second element is the coefficient of variation, ratios[3] is skew, and ratios[4] is kurtosis.
`sd`	Nearly unbiased standard deviation [well at least unbiased variance (`unbiased.sd^2`)] computed by R function `sd`.
`umvu.sd`	Uniformly-minimum variance unbiased estimator of standard deviation.
`skew`	Nearly unbiased skew, same as ratios[3].
`kurt`	Nearly unbiased kurtosis, same as ratios[4].
`excesskurt`	Excess kurtosis from the Normal distribution: `kurt - 3`.
`classic.sd`	Classical (theoretical) definition of standard deviation.
`classic.skew`	Classical (theoretical) definition of skew.

`classic.kurt` Classical (theoretical) definition of kurtosis

`classic.excesskurt` Excess classical (theoretical) kurtosis from Normal distribution: `classic.kurt - 3`.

`message` The product moments are confusing in terms of definition because they are not naturally unbiased. This characteristic is different from the L-moments. The author thinks that it is informative to show the biased versions within the "classic" designations. Therefore, this `message` includes several clarifications of the output.

`source` An attribute identifying the computational source (the function name) of the product moments: "pmoms".

Note

This function is primarily available for gamesmanship with the Pearson Type III distribution as its parameterization in **lmomco** returns the product moments as the very parameters of that distribution. This of course is like the Normal distribution in which the first two parameters are the first two product moments; the Pearson Type III just adds skew. See the example below. Another reason for having this function in **lmomco** is that it demonstrates application of unbiased product moments and permits comparisons to the L-moments (see Asquith, 2011; figs. 12.13–12.16).

The `umvu.sd` is computed by

$$\hat{\sigma}' = \frac{\Gamma[(n-1)/2]}{\Gamma(n/2)\sqrt{2}}\sqrt{\sum_{i=1}^{n}(x_i - \hat{\mu})^2},$$

where $\hat{\sigma}'$ is the estimate of standard deviation for the sample x of size n, $\Gamma(\cdots)$ is the complete gamma function, and $\hat{\mu}$ is the arthimetic mean.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

Joanes, D.N., Gill, C.A., 1998, Comparing measures of sample skewness and kurtosis: The Statistician, v. 47, no. 1, pp. 183–189.

See Also

`lmoms`

Examples

```
# A simple example
PM <- pmoms(rnorm(1000)) # n standard normal values as a fake data set.
cat(c(PM$moments[1],PM$moments[2],PM$ratios[3],PM$ratios[4],"\n"))
# As sample size gets very large the four values returned should be
# 0,1,0,0 by definition of the standard normal distribution.

# A more complex example
para <- vec2par(c(100,500,3),type='pe3') # mean=100, sd=500, skew=3
# The Pearson type III distribution is implemented here such that
# the "parameters" are equal to the mean, standard deviation, and skew.
simDATA <- rlmomco(100,para) # simulate 100 observations
PM <- pmoms(simDATA) # compute the product moments

p.tmp <- c(PM$moments[1],PM$moments[2],PM$ratios[3])
cat(c("Sample P-moments:",p.tmp,"\n"))
# This distribution has considerable variation and large skew. Stability
# of the sample product moments requires LARGE sample sizes (too large
# for a builtin example)

# Continue the example through the L-moments
lmr <- lmoms(simDATA) # compute the L-moments
epara <- parpe3(lmr) # estimate the Pearson III parameters. This is a
# hack to back into comparative estimates of the product moments. This
# can only be done because we know that the parent distribution is a
# Pearson Type III

l.tmp <- c(epara$para[1],epara$para[2],epara$para[3])
cat(c("PearsonIII by L-moments:",l.tmp,"\n"))
# The first values are the means and will be identical and close to 100.
# The second values are the standard deviations and the L-moment to
#   PearsonIII will be closer to 500 than the product moment (this
#   shows the raw power of L-moment based analysis---they work).
# The third values are the skew. Almost certainly the L-moment estimate
#   of skew will be closer to 3 than the product moment.
```

---

pp *Plotting-Position Formula*

Description

The plotting positions of a data vector (x) are returned in ascending order. The plotting-position formula is

$$pp_i = \frac{i-a}{n+1-2a},$$

where pp_i is the nonexceedance probability F of the ith ascending data value. The parameter a specifies the plotting-position type, and n is the sample size (`length(x)`). Alternatively, the plotting positions can be computed by

$$pp_i = \frac{i+A}{n+B},$$

where A and B can obviously be expressed in terms of a for $B > A > -1$ (Hosking and Wallis, 1997, sec. 2.8).

Usage

```
pp(x, A=NULL, B=NULL, a=0, sort=TRUE, ties.method="first", ...)
```

Arguments

`x`	A vector of data values. The vector is used to get sample size through `length`.
`A`	A value for the plotting-position coefficient A.
`B`	A value for the plotting-position coefficient B.
`a`	A value for the plotting-position formula from which A and B are computed, default is `a=0`, which returns the Weibull plotting positions.
`sort`	A logical whether the ranks of the data are sorted prior to F computation. It was a design mistake years ago to default this function to a sort, but it is now far too late to risk changing the logic now. The function originally lacked the `sort` argument for many years.
`ties.method`	This is the argument of the same name passed to `rank`.
`...`	Additional arguments to pass.

Value

An R `vector` is returned.

Note

Various plotting positions have been suggested in the literature. Stedinger and others (1992, p.18.25) comment that "all plotting positions give crude estimates of the unknown [non]exceedance probabilities associated with the largest (and smallest) events." The various plotting positions are summarized in the follow table.

Weibull $a = 0$, Unbiased exceedance probability for all distributions (see discussion in `pp.f`).

Median $a = 0.3175$, Median exceedance probabilities for all distributions (if so, see `pp.median`).

APL ≈ 0.35, Often used with probability-weighted moments.

Blom $a = 0.375$, Nearly unbiased quantiles for normal distribution.

Cunnane $a = 0.40$, Approximately quantile unbiased.

Gringorten $a = 0.44$, Optimized for Gumbel distribution.

Hazen $a = 0.50$, A traditional choice.

The function uses the R `rank` function, which has specific settings to handle tied data. For implementation here, the `ties.method="first"` method to `rank` is used. The user has flexibility in changing this to their own custom purposes.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

Stedinger, J.R., Vogel, R.M., and Foufoula-Georgiou, E., 1992, Frequency analysis of extreme events, in Handbook of Hydrology, chapter 18, editor-in-chief D. A. Maidment: McGraw-Hill, New York.

See Also

nonexceeds, pwm.pp, pp.f, pp.median, headrick.sheng.lalpha

Examples

```
Q  <- rnorm(20)
PP <- pp(Q)
plot(PP, sort(Q))

Q <- rweibull(30, 1.4, scale=400)
WEI <- parwei(lmoms(Q))
PP <- pp(Q)
plot( PP, sort(Q))
lines(PP, quawei(PP, WEI))

# This plot looks similar, but when connecting lines are added
# the nature of the sorting is obvious.
plot( pp(Q, sort=FALSE), Q)
lines(pp(Q, sort=FALSE), Q, col=2)
```

---

pp.f *Quantile Function of the Ranks of Plotting Positions*

Description

There are two major forms (outside of the general plotting-position formula pp) for estimation of the p_rth probability of the rth order statistic for a sample of size n: the mean is $pp'_r = r/(n+1)$ (Weibull plotting position) and the Beta quantile function is $pp_r(F) = IIB(F, r, n+1-r)$, where F represents the nonexceedance probability of the plotting position. IIB is the "inverse of the incomplete beta function" or the quantile function of the Beta distribution as provided in R by `qbeta(f, a, b)`. If $F = 0.5$, then the median is returned but that is conveniently implemented in pp.median. Readers might consult Gilchrist (2011, chapter 12) and Karian and Dudewicz (2011, p. 510).

Usage

```
pp.f(f, x)
```

Arguments

`f`	A nonexceedance probability.
`x`	A vector of data. The ranks and the length of the vector are computed within the function.

Value

An R `vector` is returned.

Note

The function uses the R function `rank`, which has specific settings to handle tied data. For implementation here, the `ties.method="first"` method to `rank` is used.

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Karian, Z.A., and Dudewicz, E.J., 2011, Handbook of fitting statistical distributions with R: Boca Raton, FL, CRC Press.

See Also

`pp`, `pp.median`

Examples

```
X <- sort(rexp(10))
PPlo <- pp.f(0.25, X)
PPhi <- pp.f(0.75, X)
plot(c(PPlo,NA,PPhi), c(X,NA,X))
points(pp(X), X) # Weibull i/(n+1)
```

`pp.median` *Quantile Function of the Ranks of Plotting Positions*

Description

The median of a plotting position. The median is $pp^\star_r = IIB(0.5, r, n+1-r)$. IIB is the "inverse of the incomplete beta function" or the quantile function of the Beta distribution as provided in R by `qbeta(f, a, b)`. Readers might consult Gilchrist (2011, chapter 12) and Karian and Dudewicz (2011, p. 510). The pp'_r are known in some fields as "mean rankit" and $pp^\star_r$ as "median rankit."

Usage

```
pp.median(x)
```

Arguments

x	A real value vector. The ranks and the length of the vector are computed within the function.

Value

An R `vector` is returned.

Note

The function internally calls `pp.f` (see **Note** in for that function).

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Karian, Z.A., and Dudewicz, E.J., 2011, Handbook of fitting statistical distributions with R: Boca Raton, FL, CRC Press.

See Also

`pp`, `pp.f`

Examples

```
## Not run:
X <- rexp(10)*rexp(10)
means  <- pp(X, sort=FALSE)
median <- pp.median(X)
supposed.median <- pp(X, a=0.3175, sort=FALSE)
lmr <- lmoms(X)
par <- parwak(lmr)
FF  <- nonexceeds()
plot(FF, qlmomco(FF, par), type="l", log="y")
points(means,  X)
points(median, X, col=2)
points(supposed.median, X, pch=16, col=2, cex=0.5)
# The plot shows that the median and supposed.median by the plotting-position
# formula are effectively equivalent. Thus, the partial application it seems
# that a=0.3175 would be good enough in lieu of the complexity of the
# quantile function of the Beta distribution.

## End(Not run)
```

prettydist *A Pretty List of Distribution Names*

Description

Return a full name of one or more distributions from the abbreviation for the distribution. The official list of abbreviations for the **lmomco** package is available under `dist.list`.

Usage

```
prettydist(x)
```

Arguments

`x`	A vector of **lmomco** distribution abbreviations.

Value

A vector of distribution identifiers.

Author(s)

W.H. Asquith

See Also

`dist.list`

Examples

```
the.lst <- dist.list() # the authoritative list of abbreviations
prettydist(the.lst)
```

prob2grv *Convert a Vector of Annual Nonexceedance Probabilities to Gumbel Reduced Variates*

Description

This function converts a vector of annual nonexceedance probabilities F to Gumbel reduced variates (GRV, grv; Hosking and Wallis [1997, p. 92])

$$grv = -\log(-\log(F)),$$

where $0 \leq F \leq 1$. The Gumbel distribution (`quagum`), which is a special case of the Generalized Extreme Value (`quagev`), will plot as a straightline when the horizontal axis is GRV transformed.

Usage

```
prob2grv(f)
```

Arguments

`f`	A vector of annual nonexceedance probabilities.

Value

A vector of Gumbel reduced variates.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`grv2prob`, `prob2T`

Examples

```
F <- nonexceeds()
grv <- prob2grv(F)
```

`prob2lrv` *Convert a Vector of Annual Nonexceedance Probabilities to Logistic Reduced Variates*

Description

This function converts a vector of annual nonexceedance probabilities F to logistic reduced variates (LRV, lrv)

$$lrv = 1/(\exp(-lrv) + 1),$$

where $0 \le F \le 1$. The logistic distribution, which is generalized by the Generalized Logistic (`quaglo`) with $\kappa = 0$, will plot as a straightline when the horizontal axis is LRV transformed.

Usage

```
prob2lrv(f)
```

Arguments

`f`	A vector of annual nonexceedance probabilities.

Value

A vector of logistic reduced variates.

Author(s)

W.H. Asquith

References

Bradford, R.B., 2002, Volume-duration growth curves for flood estimation in permeable catchments: Hydrology and Earth System Sciences, v. 6, no. 5, pp. 939–947.

See Also

lrv2prob, prob2T

Examples

```
F <- nonexceeds()
lrv <- prob2lrv(F)
## Not run:
X <- rlmomco(10040, vec2par(c(0,1,0), type="glo"))
plot(prob2lrv(pp(X, a=0.4)), sort(X)); abline(0,1)

## End(Not run)
```

---

prob2T *Convert a Vector of Annual Nonexceedance Probabilities to T-year Return Periods*

Description

This function converts a vector of annual nonexceedance probabilities F to T-year return periods

$$T = \frac{1}{1 - F},$$

where $0 \leq F \leq 1$.

Usage

```
prob2T(f)
```

Arguments

f	A vector of annual nonexceedance probabilities.

Value

A vector of T-year return periods.

Author(s)

W.H. Asquith

See Also

T2prob, nonexceeds, add.lmomco.axis, prob2grv, prob2lrv

Examples

```
F <- nonexceeds()
T <- prob2T(F)
```

---

pwm *Unbiased Sample Probability-Weighted Moments*

Description

Unbiased sample probability-weighted moments (PWMs) are computed from a sample. The β_r's are computed using

$$\beta_r = n^{-1} \sum_{j=1}^{n} \binom{j-1}{r} x_{j:n}.$$

Usage

```
pwm(x, nmom=5, sort=TRUE)
```

Arguments

`x`	A vector of data values.
`nmom`	Number of PWMs to return ($r =$ `nmom - 1`).
`sort`	Do the data need sorting? The computations require sorted data. This option is provided to optimize processing speed if presorted data already exists.

Value

An R `list` is returned.

`betas`	The PWMs. Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
`source`	Source of the PWMs: "pwm".

Author(s)

W.H. Asquith

References

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

lmoms, pwm2lmom, pwm

Examples

```
# Data listed in Hosking (1995, table 29.2, p. 551)
H <- c(3,4,5,6,6,7,8,8,9,9,9,10,10,11,11,11,13,13,13,13,13,
       17,19,19,25,29,33,42,42,51.9999,52,52,52)
# 51.9999 was really 52, but a real non censored data point.
z <-  pwmRC(H,52,checkbetas=TRUE)
str(z)
# Hosking(1995) reports that A-type L-moments for this sample are
# lamA1=15.7 and lamAL-CV=.389, and lamAL-skew=.393
pwm2lmom(z$Abetas)
# WHA gets 15.666, 0.3959, and 0.4030

# See p. 553 of Hosking (1995)
# Data listed in Hosking (1995, table 29.3, p. 553)
D <- c(-2.982, -2.849, -2.546, -2.350, -1.983, -1.492, -1.443,
       -1.394, -1.386, -1.269, -1.195, -1.174, -0.854, -0.620,
       -0.576, -0.548, -0.247, -0.195, -0.056, -0.013,  0.006,
        0.033,  0.037,  0.046,  0.084,  0.221, 0.245, 0.296)
D <- c(D,rep(.2960001,40-28)) # 28 values, but Hosking mentions
                              # 40 values in total
z <-  pwmRC(D,.2960001)
# Hosking reports B-type L-moments for this sample are
# lamB1 = -.516 and lamB2 = 0.523
pwm2lmom(z$Bbetas)
# WHA gets -.5162 and 0.5218
```

---

pwm.beta2alpha — *Conversion of Beta to Alpha Probability-Weighted Moments (PWMs) or Alpha to Beta PWMs*

Description

Conversion of "beta" (the well known ones) to "alpha" probability-weighted moments (PWMs) by pwm.beta2alpha or alpha to beta PWMs by pwm.alpha2beta. The relations between the α and β PWMs are

$$\alpha_r = \sum_{k=0}^{r} (-1)^k \binom{r}{k} \beta_k,$$

and

$$\beta_r = \sum_{k=0}^{r} (-1)^k \binom{r}{k} \alpha_k.$$

Lastly, note that the β are almost exclusively used in the literature. Because each is a linear combination of the other, they are equivalent in meaning but not numerically.

Usage

```
pwm.beta2alpha(pwm)

pwm.alpha2beta(pwm)
```

Arguments

`pwm`	A vector of alpha or beta probability-weighted moments depending on which related function is called.

Value

If $\beta_r \rightarrow \alpha_r$ (`pwm.beta2alpha`), a vector of the α_r. Note that convention is the have a α_0, but this is placed in the first index `i=1` vector. Alternatively, if $\alpha_r \rightarrow \beta_r$ (`pwm.alpha2beta`), a vector of the β_r.

Author(s)

W.H. Asquith

References

NEED

See Also

`pwm`, `pwm2lmom`

Examples

```
X <- rnorm(100)
pwm(X)$betas
pwm.beta2alpha(pwm(X)$betas)
pwm.alpha2beta(pwm.beta2alpha(pwm(X)$betas))
```

pwm.gev	*Generalized Extreme Value Plotting-Position Probability-Weighted Moments*

Description

Generalized Extreme Value plotting-position probability-weighted moments (PWMs) are computed from a sample. The first five β_r's are computed by default. The plotting-position formula for the Generalized Extreme Value distribution is

$$pp_i = \frac{i - 0.35}{n},$$

where pp_i is the nonexceedance probability F of the ith ascending values of the sample of size n. The PWMs are computed by

$$\beta_r = n^{-1} \sum_{i=1}^{n} pp_i^r \times x_{j:n},$$

where $x_{j:n}$ is the jth order statistic $x_{1:n} \leq x_{2:n} \leq x_{j:n} \cdots \leq x_{n:n}$ of random variable X, and r is $0, 1, 2, \ldots$. Finally, `pwm.gev` dispatches to `pwm.pp(data,A=-0.35,B=0)` and does not have its own logic.

Usage

```
pwm.gev(x, nmom=5, sort=TRUE)
```

Arguments

`x`	A vector of data values.
`nmom`	Number of PWMs to return.
`sort`	Do the data need sorting? The computations require sorted data. This option is provided to optimize processing speed if presorted data already exists.

Value

An R `list` is returned.

`betas`	The PWMs. Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
`source`	Source of the PWMs: “pwm.gev”.

Author(s)

W.H. Asquith

References

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

pwm.ub, pwm.pp, pwm2lmom

Examples

```
pwm <- pwm.gev(rnorm(20))
```

---

pwm.pp *Plotting-Position Sample Probability-Weighted Moments*

Description

The sample probability-weighted moments (PWMs) are computed from the plotting positions of the data. The first five β_r's are computed by default. The plotting-position formula for a sample size of n is

$$pp_i = \frac{i + A}{n + B},$$

where pp_i is the nonexceedance probability F of the ith ascending data values. An alternative form of the plotting position equation is

$$pp_i = \frac{i + a}{n + 1 - 2a},$$

where a is a single plotting position coefficient. Having a provides A and B, therefore the parameters A and B together specify the plotting-position type. The PWMs are computed by

$$\beta_r = n^{-1} \sum_{i=1}^{n} pp_i^r \times x_{j:n},$$

where $x_{j:n}$ is the jth order statistic $x_{1:n} \leq x_{2:n} \leq x_{j:n} \cdots \leq x_{n:n}$ of random variable X, and r is $0, 1, 2, \dots$ for the PWM order.

Usage

```
pwm.pp(x, pp=NULL, A=NULL, B=NULL, a=0, nmom=5, sort=TRUE)
```

Arguments

x	A vector of data values.
pp	An optional vector of nonexceedance probabilities. If present then A and B or a are ignored.
A	A value for the plotting-position formula. If A and B are both zero then the unbiased PWMs are computed through pwm.ub.
B	Another value for the plotting-position formula. If A and B are both zero then the unbiased PWMs are computed through pwm.ub.
a	A single plotting position coefficient from which, if not NULL, A and B will be internally computed;
nmom	Number of PWMs to return.
sort	Do the data need sorting? The computations require sorted data. This option is provided to optimize processing speed if presorted data already exists.

Value

An R list is returned.

betas	The PWMs. Note that convention is the have a β_0, but this is placed in the first index i=1 of the betas vector.
source	Source of the PWMs: "pwm.pp".

Author(s)

W.H. Asquith

References

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

pwm.ub, pwm.gev, pwm2lmom

Examples

```
pwm <- pwm.pp(rnorm(20), A=-0.35, B=0)

X <- rnorm(20)
pwm <- pwm.pp(X, pp=pp(X)) # weibull plotting positions
```

pwm.ub *Unbiased Sample Probability-Weighted Moments*

Description

Unbiased sample probability-weighted moments (PWMs) are computed from a sample. The β_r's are computed using

$$\beta_r = n^{-1} \binom{n-1}{r}^{-1} \sum_{j=1}^{n} \binom{j-1}{r} x_{j:n}.$$

Usage

```
pwm.ub(x, nmom=5, sort=TRUE)
```

Arguments

x	A vector of data values.
nmom	Number of PWMs to return ($r =$ nmom - 1).
sort	Do the data need sorting? The computations require sorted data. This option is provided to optimize processing speed if presorted data already exists.

Value

An R list is returned.

betas	The PWMs. Note that convention is the have a β_0, but this is placed in the first index i=1 of the betas vector.
source	Source of the PWMs: "pwm.ub".

Note

Through a user inquiry, it came to the author's attention in May 2014 that some unrelated studies using PWMs in the earth-system sciences have published erroneous sample PWMs formula. Because **lmomco** is intended to be an authoritative source, here are some computations to further prove correctness with provenance:

```
"pwm.handbookhydrology" <- function(x, nmom=5) {
   x <- sort(x, decreasing = TRUE); n <- length(x); betas <- rep(NA, nmom)
   for(r in 0:(nmom-1)) {
      tmp <- sum(sapply(1:(n-r),
          function(j) { choose(n - j, r) * x[j] / choose(n - 1, r) }))
      betas[(r+1)] <- tmp/n
   }
   return(betas)
}
```

and a demonstration with alternative algebra in Stedinger and others (1993)

```
set.seed(1)
glo <- vec2par(c(123,1123,-.5), type="glo"); X <- rlmomco(100, glo)
lmom2pwm(lmoms(X, nmom=5))$betas # unbiased L-moments flipped to PWMs
[1]  998.7932 1134.0658 1046.4906  955.8872  879.3349
pwm.ub(X, nmom=5)$betas  # Hosking and Wallis (1997) and Asquith (2011)
[1]  998.7932 1134.0658 1046.4906  955.8872  879.3349
pwm.handbookhydrology(X) # ** alert reverse sort, opposite usually seen**
[1]  998.7932 1134.0658 1046.4906  955.8872  879.3349
```

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Stedinger, J.R., Vogel, R.M., Foufoula-Georgiou, E., 1993, Frequency analysis of extreme events: *in* Handbook of Hydrology, ed. Maidment, D.R., McGraw-Hill, Section 18.6 Partial duration series, mixtures, and censored data, pp. 18.37–18.39.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

pwm.pp, pwm.gev, pwm2lmom

Examples

```
pwm <- pwm.ub(rnorm(20))
```

---

pwm2lmom — *Probability-Weighted Moments to L-moments*

Description

Converts the probability-weighted moments (PWM) to the L-moments. The conversion is linear so procedures based on PWMs are identical to those based on L-moments through a system of linear equations

$$\lambda_1 = \beta_0,$$

$$\lambda_2 = 2\beta_1 - \beta_0,$$

$$\lambda_3 = 6\beta_2 - 6\beta_1 + \beta_0,$$
$$\lambda_4 = 20\beta_3 - 30\beta_2 + 12\beta_1 - \beta_0,$$
$$\lambda_5 = 70\beta_4 - 140\beta_3 + 90\beta_2 - 20\beta_1 + \beta_0,$$
$$\tau = \lambda_2/\lambda_1,$$
$$\tau_3 = \lambda_3/\lambda_2,$$
$$\tau_4 = \lambda_4/\lambda_2, \text{ and}$$
$$\tau_5 = \lambda_5/\lambda_2.$$

The general expression and the expression used for computation if the argument is a vector of PWMs is

$$\lambda_{r+1} = \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} \binom{r+k}{k} \beta_{k+1}.$$

Usage

```
pwm2lmom(pwm)
```

Arguments

pwm	A PWM object created by pwm.ub or similar.

Details

The probability-weighted moments (PWMs) are linear combinations of the L-moments and therefore contain the same statistical information of the data as the L-moments. However, the PWMs are harder to interpret as measures of probability distributions. The linearity between L-moments and PWMs means that procedures base on one are equivalent to the other.

The function can take a variety of PWM argument types in pwm. The function checks whether the argument is an R list and if so attempts to extract the β_r's from list names such as BETA0, BETA1, and so on. If the extraction is successful, then a list of L-moments similar to lmom.ub is returned. If the extraction was not successful, then an R list name betas is checked; if betas is found, then this vector of PWMs is used to compute the L-moments. If pwm is a list but can not be routed in the function, a warning is made and NULL is returned. If the pwm argument is a vector, then this vector of PWMs is used. to compute the L-moments are returned.

Value

One of two R lists are returned. Version I is

L1	Arithmetic mean.
L2	L-scale—analogous to standard deviation.
LCV	coefficient of L-variation—analogous to coe. of variation.
TAU3	The third L-moment ratio or L-skew—analogous to skew.
TAU4	The fourth L-moment ratio or L-kurtosis—analogous to kurtosis.
TAU5	The fifth L-moment ratio.

L3	The third L-moment.
L4	The fourth L-moment.
L5	The fifth L-moment.

Version II is

lambdas	The L-moments.
ratios	The L-moment ratios.
source	Source of the L-moments "pwm2lmom".

Author(s)

W.H. Asquith

References

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Hosking, J.R.M., 1990, L-moments–Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

lmom.ub, pwm.ub, pwm, lmom2pwm

Examples

```
D <- c(123,34,4,654,37,78)
pwm2lmom(pwm.ub(D))
pwm2lmom(pwm(D))
pwm2lmom(pwm(rnorm(100)))
```

---

pwm2vec *Convert Probability-Weighted Moment object to a Vector*

Description

This function converts a probability-weighted moment object in the structure used by **lmomco** into a simple vector of β_0, β_1, β_2, β_3, β_4, $\ldots$, β_{r-1}.

Usage

```
pwm2vec(pwm, ...)
```

Arguments

pwm	Probability-weighted moment object such as from pwm and vec2pwm.
...	Not presently used.

Value

A vector of the first five probability-weighted moments if available. The $betas field of the pwm argument is simply returned by this function.

Author(s)

W.H. Asquith

See Also

pwm, vec2pwm, lmom2vec

Examples

```
pmr <- pwm(rnorm(40));            pwm2vec(pmr)
pmr <- vec2pwm(c(140,150,45,21));  pwm2vec(pmr)
```

---

pwmLC *Sample Probability-Weighted Moments for Left-Tail Censoring*

Description

Compute the sample probability-weighted moments (PWMs) for left-tail censored data set—that is a data set censored from below. The censoring threshold is denoted as T.

Usage

```
pwmLC(x, threshold=NULL, nmom=5, sort=TRUE)
```

Arguments

x	A vector of data values.
threshold	The left-tail censoring (lower) threshold.
nmom	Number of PWMs to return.
sort	Do the data need sorting? Note that convention is the have a β'_0, but this is placed in the first index i=1 of the betas vector.

Details

There is some ambiguity if the threshold also numerically equals valid data in the data set. In the data for the examples below, which are taken from elsewhere, there are real observations at the censoring level. One can see how a hack is made to marginally decrease or increase the data or the threshold for the computations. This is needed because the code uses

```
sapply(x, function(v) { if(v >= T) return(T); return(v) } )
```

to reset the data vector x. By operating on the data in this fashion one can toy with various levels of the threshold for experimental purposes; this seemed a more natural way for general implementation. The code sets n = length(x) and m = n - length(x[x == T]), which also seems natural. The β_r^A are computed by dispatching to pwm.

Value

An R list is returned.

Aprimebetas	The A'-type PWMs. These should be same as pwm() returns if there is no censoring. Note that convention is the have a β_0, but this is placed in the first index i=1 of the betas vector.
Bprimebetas	The B'-type PWMs. These should be NA if there is no censoring. Note that convention is the have a β_0, but this is placed in the first index i = 1 of the betas vector.
source	Source of the PWMs: "pwmLC".
threshold	The upper censoring threshold.
zeta	The left censoring fraction: numbelowthreshold/samplesize.
numbelowthreshold	Number of data points equal to or above the threshold.
observedsize	Number of real data points in the sample (above the threshold).
samplesize	Number of actual sample values.

Author(s)

W.H. Asquith

References

Zafirakou-Koulouris, A., Vogel, R.M., Craig, S.M., and Habermeier, J., 1998, L-moment diagrams for censored observations: Water Resources Research, v. 34, no. 5, pp. 1241–1249.

See Also

lmoms, pwm2lmom, pwm, pwmRC

Examples

```
#
```

pwmRC *Sample Probability-Weighted Moments for Right-Tail Censoring*

Description

Compute the sample Probability-Weighted Moments (PWMs) for right-tail censored data set—that is a data set censored from above. The censoring threshold is denoted as T. The data possess m values that are observed (noncensored, $< T$) out of a total of n samples. The ratio of m to n is defined as $\zeta = m/n$, which will play an important role in parameter estimation. The ζ is interpreted as the probability $\mathrm{Pr}\{\}$ that x is less than the quantile at ζ nonexceedance probability: ($\mathrm{Pr}\{x < X(\zeta)\}$). Two types of PWMs are computed for right-tail censored situations. The "A"-type PWMs and "B"-type PWMs. The A-type PWMs are defined by

$$\beta_r^A = m^{-1} \sum_{j=1}^{m} {j-1 \choose r} x_{[j:n]},$$

which are the PWMs of the uncensored sample of m observed values. The B-type PWMs are computed from the "complete" sample, in which the $n - m$ censored values are replaced by the censoring threshold T. The B-type PWMs are defined by

$$\beta_r^B = n^{-1} \left(\sum_{j=1}^{m} {j-1 \choose r} x_{[j:n]} + \sum_{j=m+1}^{n} {j-1 \choose r} T \right).$$

The two previous expressions are used in the function. These PWMs are readily converted to L-moments by the usual methods (`pwm2lmom`). When there are more than a few censored values, the PWMs are readily computed by computing β_r^A and using the expression

$$\beta_r^B = Z\beta_r^A + \frac{1-Z}{r+1} T,$$

where

$$Z = \frac{m}{n} \frac{{m-1 \choose r}}{{n-1 \choose r}}.$$

The two expressions above are consulted when the `checkbetas=TRUE` argument is present. Both sequences of B-type are `cated` to the terminal. This provides a check on the implementation of the algorithm. The functions `Apwm2BpwmRC` and `Bpwm2ApwmRC` can be used to switch back and forth between the two PWM types given fitted parameters for a distribution in the **lmomco** package that supports right-tail censoring. Finally, the `RC` in the function name is to denote `R`ight-tail `C`ensoring.

Usage

```
pwmRC(x, threshold=NULL, nmom=5, sort=TRUE, checkbetas=FALSE)
```

Arguments

`x`	A vector of data values.
`threshold`	The right-tail censoring (upper) threshold.
`nmom`	Number of PWMs to return.
`sort`	Do the data need sorting? Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
`checkbetas`	A cross relation between β_r^A and β_r^B exists—display the results of the secondary computation of the β_r^B. The two displayed vectors should be numerically equal.

Details

There is some ambiguity if the threshold also numerically equals valid data in the data set. In the data for the examples below, which are taken from elsewhere, there are real observations at the censoring level. One can see how a hack is made to marginally decrease or increase the data or the threshold for the computations. This is needed because the code uses

```
sapply(x, function(v) { if(v >= T) return(T); return(v) } )
```

to reset the data vector `x`. By operating on the data in this fashion one can toy with various levels of the threshold for experimental purposes; this seemed a more natural way for general implementation. The code sets n = `length(x)` and m = `n - length(x[x == T])`, which also seems natural. The β_r^A are computed by dispatching to `pwm`.

Value

An R `list` is returned.

`Abetas`	The A-type PWMs. These should be same as `pwm()` returns if there is no censoring. Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
`Bbetas`	The B-type PWMs. These should be `NA` if there is no censoring. Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
`source`	Source of the PWMs: “pwmRC”.
`threshold`	The upper censoring threshold.
`zeta`	The right censoring fraction: `numabovethreshold/samplesize`.
`numabovethreshold`	Number of data points equal to or above the threshold.
`observedsize`	Number of real data points in the sample (below the threshold).
`samplesize`	Number of actual sample values.

Author(s)

W.H. Asquith

References

Greenwood, J.A., Landwehr, J.M., Matalas, N.C., and Wallis, J.R., 1979, Probability weighted moments—Definition and relation to parameters of several distributions expressable in inverse form: Water Resources Research, v. 15, pp. 1,049–1,054.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

lmoms, pwm2lmom, pwm, pwmLC

Examples

```
# Data listed in Hosking (1995, table 29.2, p. 551)
H <- c(3,4,5,6,6,7,8,8,9,9,9,10,10,11,11,11,13,13,13,13,13,
       17,19,19,25,29,33,42,42,51.9999,52,52,52)
# 51.9999 was really 52, a real (noncensored) data point.
z <-  pwmRC(H,threshold=52,checkbetas=TRUE)
str(z)
# Hosking(1995) reports that A-type L-moments for this sample are
# lamA1=15.7 and lamAL-CV=.389, and lamAL-skew=.393
pwm2lmom(z$Abetas)
# My version of R reports 15.666, 0.3959, and 0.4030


# See p. 553 of Hosking (1995)
# Data listed in Hosking (1995, table 29.3, p. 553)
D <- c(-2.982, -2.849, -2.546, -2.350, -1.983, -1.492, -1.443,
       -1.394, -1.386, -1.269, -1.195, -1.174, -0.854, -0.620,
       -0.576, -0.548, -0.247, -0.195, -0.056, -0.013,  0.006,
        0.033,  0.037,  0.046,  0.084,  0.221,  0.245,  0.296)
D <- c(D,rep(.2960001,40-28)) # 28 values, but Hosking mentions
                              # 40 values in total
z <-  pwmRC(D,.2960001)
# Hosking reports B-type L-moments for this sample are
# lamB1 = -.516 and lamB2 = 0.523
pwm2lmom(z$Bbetas)
# My version of R reports -.5162 and 0.5218
```

---

qlmomco *Quantile Function of the Distributions*

Description

This function acts as an alternative front end to par2qua. The nomenclature of the qlmomco function is to mimic that of built-in R functions that interface with distributions.

Usage

```
qlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from lmom2par or similar.

Value

Quantile value for F for the specified parameters.

Author(s)

W.H. Asquith

See Also

dlmomco, plmomco, rlmomco, slmomco, add.lmomco.axis, supdist

Examples

```
para <- vec2par(c(0,1),type='nor') # standard normal parameters
p75  <- qlmomco(.75,para) # 75th percentile of one standard deviation
```

---

qua.ostat *Compute the Quantiles of the Distribution of an Order Statistic*

Description

This function computes a specified quantile by nonexceedance probability F for the jth-order statistic of a sample of size n for a given distribution. Let the quantile function (inverse distribution) of the Beta distribution be

$$\mathrm{B}^{(-1)}(F, j, n - j + 1),$$

and let $x(F, \Theta)$ represent the quantile function of the given distribution and Θ represents a vector of distribution parameters. The quantile function of the distribution of the jth-order statistic is

$$x\big(\mathrm{B}^{(-1)}(F, j, n - j + 1), \Theta\big).$$

Usage

```
qua.ostat(f, j, n, para=NULL)
```

Arguments

f	The nonexceedance probability F for the quantile.
j	The jth-order statistic $x_{1:n} \leq x_{2:n} \leq \ldots \leq x_{j:n} \leq x_{n:n}$.
n	The sample size.
para	A distribution parameter list from a function such as `lmom2par` or `vec2par`.

Value

The quantile of the distribution of the jth-order statistic is returned.

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton, Fla.

See Also

`lmom2par`, `vec2par`

Examples

```
gpa <- vec2par(c(100, 500, 0.5), type="gpa")
n <- 20   # the sample size
j <- 15   # the 15th order statistic
F <- 0.99 # the 99th percentile
theoOstat <- qua.ostat(F, j, n, gpa)

## Not run:
# Let us test this value against a brute force estimate.
Jth <- vector(mode="numeric")
for(i in seq_len(50000)) {
  Q <- sort( rlmomco(n, gpa) )
  Jth[i] <- Q[j]
}
bruteOstat <- quantile(Jth, F) # estimate by built-in function
theoOstat  <- signif( theoOstat, digits=5)
bruteOstat <- signif(bruteOstat, digits=5)
cat(c("Theoretical=", theoOstat, "  Simulated=", bruteOstat, "\n")) #
## End(Not run)
```

qua2ci.cov	*Estimate a Confidence Interval for Quantiles of a Parent Distribution using Sample Variance-Covariances of L-moments*

Description

This function estimates the lower and upper limits of a specified confidence interval for aribitrary quantile values for a sample x and a specified distribution form. The estimation is based on the sample variance-covariance structure of the L-moments (lmoms.cov) through a Monte Carlo approach. The quantile values, actually the nonexceedance probabilities (F for $0 \le F \le 1$), are specified by the user. The user provides type of parent distribution distribution and this form which will be fitted internal to the function.

Usage

```
qua2ci.cov(x,f, type=NULL, nsim=1000,
                interval=c("confidence", "none"), level=0.90, tol=1E-6,
                asnorm=FALSE, altlmoms=NULL, flip=NULL, dimless=TRUE,
                usefastlcov=TRUE, nmom=5, getsimlmom=FALSE, verbose=FALSE, ...)
```

Arguments

x	A real value vector.
f	Nonexceedance probabilities ($0 \le F \le 1$) of the quantiles for which the confidence interval is needed.
type	Three character distribution type (for example, type='gev').
nsim	The number of simulations to perform. Large numbers produce more refined confidence limit estimates at the cost of CPU time. The default is anticipated to be large enough to semi-quantitatively interpret results without too much computational delay. Larger simulation numbers are recommended.
interval	The type of interval to compute. If "none", then the simulated quantiles are returned at which point *only* the first value in f or f[1] will be considered but a warning will be issued to remind the user. This option is nice for making boxplots of the quantile distribution.
level	The confidence interval ($0 \le$ level < 1). The interval is specified as the size of the interval for which the default is 0.90 or the 90th percentile. The function will return the 5th $[(1-0.90)/2]$ and 95th $[(1-(1-0.90)/2)]$ percentile cumulative probability of the simulated quantile distribution as specified by the nonexceedance probability argument.
tol	The tolerance argument of same name and default to feed to MASS::mvrnorm() and try increasing this tolerance if the error "'Sigma' is not positive definite" occurs (see **Note** for more discussion).
asnorm	Use the mean and standard deviation of the simulated quantiles as parameters of the Normal distribution to estimate the confidence interval. Otherwise, a Bernstein polynomial approximation (dat2bernqua) to the empirical distribution of the simulated quantile distribution is used.

altlmoms	Alternative L-moments to rescale the simulated L-moments from the variance-covariance structure of the sample L-moments in x. These L-moments need to be an **lmomco** package L-moment object (e.g. lmoms). The presence of alternative L-moments will result in dimless=TRUE.
flip	A flipping or reflection value denoted as η. The values in x are flipped by this value ($y = \eta - x$) and analysis proceeds with flipped information, and then results are flipped back just prior to returning values with the exception that if getsimlmom=TRUE then the simultated L-moments are in "flipped space."
dimless	Perform the simulations in dimensionless space meaning that values in x are converted by $y = (x - \lambda_1)/\lambda_2$ and simulation based on y and scale is returned on output according to the L-moments of x or the alternative L-moments in altlmoms. Scale is returned to the simulated L-moments, if returned by getsimlmom=TRUE, which is not fully parallel with the returned behavior when flipping is involved.
usefastlcov	A logical to use the function Lmomcov() from the **Lmoments** package to compute the sample variance-covariance matrices and not the much slower function lmoms.cov in the **lmomco** package.
nmom	The number of L-moments involved. This argument needs to be high enough to permit parameterization of the distribution in type but computational effort increases as nmom gets large. This option is provided in conjunction with getsimlmom=TRUE to be able to get a "wider set" of simulated L-moments returned than precisely required by the distribution. Also, some distributions might as part of their specific fitting algorithms, require inspection of higher L-moments than seemingly required than their numer of parameters suggests.
getsimlmom	A logical controlling whether the simulated L-moment matrix having nsim rows and nmom columns is returned instead of confidence limits.
verbose	The verbosity of the operation of the function.
...	Additional arguments to pass such as to lmom2par.

Value

An R data.frame is returned.

lwr	The lower value of the confidence interval having nonexceedance probability equal to $(1-\texttt{level})/2$.
fit	The fit of the quantile based on the L-moments of x and possibly by reflection controlled by flip or based on the alternative L-moments in altlmoms and again by the reflection controlled by flip.
upr	The upper value of the confidence interval having nonexceedance probability equal to $1 - (1-\texttt{level})/2$.
qua_med	The median of the simulated quantiles.
qua_mean	The mean of the simulated quantiles for which the median and mean should be very close if the simulation size is large enough and the quantile distribution is symmetrical.
qua_var	The variance ($\sigma^2(F)$) of the simulated quantiles.
qua_lam2	The L-scale ($\lambda_2(F)$) of the simulated quantiles for which $\sigma^2(F) \approx \pi \times \lambda_2^2(F)$.

Note

These particular data set needs further evaluation as these particular sample can produce non-positive definite matrix being fed to `MASS:mvrnorm()`. It is noted that there are no ties in this data set.

```
test_dat <- c(0.048151736, 0.036753258, 0.034895847, 0.082792447, 0.096984927,
              0.213977450, 0.020264292, 0.269585438, 0.304746113, 0.066339093,
              0.015651114, 0.025122412, 0.184095698, 0.047167958, 0.049824752,
              0.043390768, 0.055228680, 0.009325696, 0.042145010, 0.008113992,
              0.118901521, 0.050399301, 0.049646181, 0.032299402, 0.015229284,
              0.013684668, 0.049371734, 0.068426211, 0.207159600, 0.087228473,
              0.306276783, 0.024870356, 0.016946801, 0.051553444, 0.017654117)
qua2ci.cov(test_dat, 0.5, type="pe3", tol=1E-6, nmom=5) # fails

lams <- lmoms(    test_dat)$lambdas
lamc <- lmoms.cov(test_dat)
n <- 100
set.seed(1)
MV1 <- mvtnorm::rmvnorm(n, mean=lams, sigma=lamc, method="eigen")
MV1 <- mvtnorm::rmvnorm(n, mean=lams, sigma=lamc, method="chol")
MV1 <- mvtnorm::rmvnorm(n, mean=lams, sigma=lamc, method="svd")
colnames(MV1) <- paste0(rep("lam",5),1:5)
set.seed(1)
MV2 <- MASS::mvrnorm(n, lams, lamc, tol=5E-2)
set.seed(1)
MV3 <- MASS::mvrnorm(n, lams, lamc, tol=Inf)

summary(MV2-MV3)
summary(MV1)
summary(MV2)
plotlmrdia(lmrdia(), xlim=c(0.3,0.7), ylim=c(0,.6))
points(MV1[,3]/MV1[,2], MV1[,4]/MV1[,2], col="red",  cex=0.5)
points(MV2[,3]/MV2[,2], MV2[,4]/MV2[,2], col="blue", cex=0.5)
```

Next we, try focusing on the upper left corner of the matrix, after all we do not need beyond the 3rd moment because the Pearson III is being used.

```
qua2ci.cov(test_dat, 0.5, type="pe3", tol=1E-6, nmom=3) # fails
```

Now try increasing the tolerance setting on the matrix postive definite test in the `MASS::mvrnorm()` function.

```
qua2ci.cov(test_dat, 0.5, type="pe3", tol=1E-4, nmom=5) # fails
```

Now try again just focusing on the upper left corner that we really need.

```
set.seed(1)
qua2ci.cov(test_dat, 0.5, type="pe3", tol=1E-4, nmom=3) # IT WORKS
# nonexceed     lwr      fit      upr  qua_med qua_mean   qua_var qua_lam2
#       0.5 0.02762 0.044426 0.061189 0.044322 0.044319 0.0001019 0.005672
```

Let us now try a hack of smoothing the data through the Bernstein polynomial. Perhaps subtle issues in the data can be "fixed" by this and the seed has been set to have the `MASS::mvrnorm()` see the same seed although the variance-covariance matrix is slightly changing. Notice that the tolerance now returns to the default and that we are requesting up through the 5th L-moment.

```
set.seed(1)
n <- length(test_dat)
smth_dat <- dat2bernqua((1:n)/(n+1), test_dat)
qua2ci.cov(smth_dat, 0.5, type="pe3", tol=1E-6, nmom=5) # IT WORKS
# nonexceed     lwr      fit     upr  qua_med qua_mean   qua_var  qua_lam2
#       0.5 0.02864 0.048288 0.06778 0.048406 0.048201 0.0001405 0.0066678
```

A quick look at the smoothing. The author is not advocating for this but this trick might be useful in data-mining scale work where for some samples, we need something back. The user might then consider using the differences `upr−fit` and `fit−lwr` to reconstruct the interval from a fit based on the original sample.

```
plot( (1:n)/(n+1), sort(test_dat))
lines((1:n)/(n+1), smth_dat, col=2)
```

Author(s)

W.H. Asquith

See Also

`lmoms`, `lmoms.cov`, `qua2ci.simple`

Examples

```
## Not run:
samsize <- 128; nsim <- 2000; f <- 0.999
wei <- parwei(vec2lmom(c(100,75,-.3)))
set.seed(1734); X <- rlmomco(samsize, wei); set.seed(1734)
tmp <- qua2ci.cov(X, f, type="wei", nsim=nsim)
print(tmp) # show results of one 2000 replicated Monte Carlo
# nonexceed     lwr    fit    upr  qua_med  qua_mean  qua_var  qua_lam2
#     0.999   310.4  333.2  360.2    333.6     334.3    227.3    8.4988
set.seed(1734)
qf <- qua2ci.cov(X, f, type="wei", nsim=nsim, interval="none") # another
boxplot(qf)
message(" quantile variance: ", round(tmp$qua_var,  digits=2),
        " compared to ", round(var(qf, na.rm=TRUE), digits=2))
set.seed(1734)
genci.simple(wei, n=samsize, f=f)
# nonexceed     lwr    fit    upr  qua_med  qua_mean  qua_var  qua_lam2
#     0.999   289.7  312.0  337.7    313.5     313.6    213.5    8.2330

#-----------------------------------------
# Using X from above example, demonstrate that using dimensionless
# simulation that the results are the same.
```

```
set.seed(145); qua2ci.cov(X, 0.1, type="wei") # both outputs same
set.seed(145); qua2ci.cov(X, 0.1, type="wei", dimless=TRUE)
# nonexceed     lwr    fit    upr  qua_med  qua_mean  qua_var  qua_lam2
#       0.1  -78.62 -46.01 -11.39   -43.58    -44.38   416.04     11.54

#----------------------------------------
# Using X again, demonstration application of the flip and notice that just
# simple reversal is occurring and that the Weibull is a reversed GEV.
eta <- 0
set.seed(145); qua2ci.cov(X, 0.9, type="wei", nsim=nsim)
# nonexceed     lwr    fit    upr  qua_med  qua_mean  qua_var  qua_lam2
#       0.9   232.2  244.2  255.9    244.3     244.1    51.91    4.0635
set.seed(145); qua2ci.cov(X, 0.9, type="gev", nsim=nsim, flip=eta)
# nonexceed     lwr    fit    upr  qua_med  qua_mean  qua_var  qua_lam2
#       0.9   232.2  244.2  256.2    244.2     244.3    53.02    4.1088
# The values are slightly different, which likely represents a combination
# of numerics of the variance-covariance matrix because the Monte Carlo
# is seeded the same.

#----------------------------------------
# Using X again, removed dimension and have the function add it back.
lmr <- lmoms(X); Y <- (X - lmr$lambdas[1])/lmr$lambdas[2]
set.seed(145); qua2ci.cov(Y, 0.9, type="wei", altlmoms=lmr, nsim=nsim)
# nonexceed     lwr    fit    upr  qua_med  qua_mean  qua_var  qua_lam2
#       0.9   232.2  244.2  255.9    244.3     244.1    51.91   4.0635
## End(Not run)
```

`qua2ci.simple` *Estimate a Confidence Interval for a Single Quantile of a Parent Distribution by a Simple Algorithm*

Description

This function estimates the lower and upper limits of a specified confidence interval for an aribitrary quantile value of a specified parent distribution [quantile function $Q(F, \theta)$ with parameters θ] using Monte Carlo simulation. The quantile value, actually the nonexceedance probability (F for $0 \le F \le 1$) of the value, is specified by the user. The user also provides the parameters of the parent distribution (see `lmom2par`). This function does consider an estimate of the variance-covariance structure of the sample data (for that see `qua2ci.cov`). The `qua2ci.simple` is the original implementation and dates close to the initial releases of **lmomco** and was originally named `qua2ci`. That name is now deprecated but retained as an alias, which will be removed at some later release.

For `nsim` simulation runs (ideally a large number), samples of size n are drawn from $Q(F, \theta)$. The L-moments of each simulated sample are computed using `lmoms` and a distribution of the same type is fit. The F-quantile of the just-fitted distribution is computed and placed into a vector. The process of simulating the sample, computing the L-moments, computing the parameters, and solving for the F-quantile is repeated for the specified number of simulation runs.

To estimate the confidence interval, the L-moments of the vector simulated quantiles are computed. Subsequently, the parameters of a user-specified distribution "error" distribution (`edist`) are computed. The two quantiles of this error distribution for the specified confidence interval are computed.

These two quantiles represent the estimated lower and upper limits for the confidence interval of the parent distribution for samples of size n. The error distribution defaults to the Generalized Normal (see pargno) because this distribution has the Normal as a special case but extends the fit to the 3rd L-moment (τ_3) for exotic situations in which some asymmetry in the quantile distribution might exist.

Finally, it is often useful to have vectors of lower and upper limits for confidence intervals for a vector of F values. The function genci.simple does just that and uses qua2ci.simple as the computational underpinning.

Usage

```
qua2ci.simple(f,para,n, level=0.90, edist="gno", nsim=1000, showpar=FALSE,
                        empdist=TRUE, verbose=FALSE, maxlogdiff=6, ...)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$) of the quantile for which the confidence interval is needed. This function is not vectorized and therefore only the first value will be used. This is in contrast to the vectorization of F in the conceptually similar function qua2ci.cov.
para	The parameters from lmom2par or vec2par—these parameters represent the "true" parent.
n	The sample size for each Monte Carlo simulation will use.
level	The confidence interval ($0 \leq$ `level` < 1). The interval is specified as the size of the interval. The default is 0.90 or the 90th percentile. The function will return the 5th $[(1-0.90)/2]$ and 95th $[(1-(1-0.90)/2)]$ percentile cumulative probability of the simulated quantile distribution as specified by the nonexceedance probability argument. The arguments `level` and `f` therefore are separate features.
edist	The model for the error distribution. Although the Normal (the default) commonly is assumed in error analyses, it need not be, as support for other distributions supported by **lmomco** is available. The default is the Generalized Normal so the not only is the Normal possible but asymmetry is also accomodated (lmomgno). For example, if the L-skew (τ_4) or L-kurtosis (τ_4) values depart considerably from those of the Normal ($\tau_3 = 0$ and $\tau_4 = 0.122602$), then the Generalized Normal or some alternative distribution would likely provide more reliable confidence interval estimation.
nsim	The number of simulations (replications) for the sample size `n` to perform. Large numbers produce more refined confidence limit estimates at the cost of CPU time. The default is anticipated to be large enough for evaluative-useage without too much computational delay. Larger simulation numbers are recommended.
showpar	The parameters of the `edist` for each simulation are printed.
empdist	If `TRUE`, then an R `environment` is appended onto the element `empdist` in the returned list, otherwise `empdist` is `NA`.
verbose	The verbosity of the operation of the function.

maxlogdiff — The maximum permitted difference in log10 space between a simulated quantile and the true value. It is possible that a well fit simulated sample to the parent distribution type provides crazy quantile estimates in the far reaches of either tail. The default value of 6 was chosen based on experience with the Kappa distribution fit to a typical heavy-right tail flood magnitude data set. The concern motivating this feature is that as the number of parameters increases, it seems progressively there is more chance for a distribution tail to swing wildy into regions for which an analyst would not be comfortable with given discipline-specific knowledge. The choice of 6-log cycles is *ad hoc* at best, and users are encouraged to do their own exploration. If verbose=TRUE then a message will be printed when the maxlogdiff condition is tripped.

... — Additional arguments to pass such as to lmom2par.

Value

An R list is returned. The lwr and upr match the nomenclature of qua2ci.cov but because qua2ci.simple is provided the parent, the true value is returned, whereas qua2ci.cov returns the fit.

lwr — The lower value of the confidence interval having nonexceedance probability equal to $(1-\texttt{level})/2$.

true — The value returned by par2qua(f,para).

upr — The upper value of the confidence interval having nonexceedance probability equal to $1-(1-\texttt{level})/2$.

elmoms — The L-moments from lmoms of the distribution of simulated of quantiles.

epara — The parameters of the error distribution fit using the elmoms.

empdist — An R environment (see below).

ifail — A diagnostic value. A value of zero means that successful exit was made.

ifailtext — A descriptive message related to the ifail value.

nsim — An echoing of the nsim argument for the function.

sim.attempts — The number of executions of the while loop (see Note below).

The empdist element in the returned list is an R environment that contains:

simquas — A nsim-long vector of the simulated quantiles for f.

empir.dist.lwr — The *lower* limit derived from the R quantile function for type=6, which uses $i/(n+1)$.

empir.dist.upr — The *upper* limit derived from the R quantile function for type=6, which uses $i/(n+1)$.

bern.smooth.lwr — The *lower* limit estimated by the Bernstein smoother in dat2bernqua for poly.type = "Bernstein" and bound.type = "none".

bern.smooth.upr — The *upper* limit estimated by the Bernstein smoother in dat2bernqua for poly.type = "Bernstein" and bound.type = "none".

epmoms — The product moments of the simulated quantiles from pmoms.

Note

This function relies on a `while` loop that runs until `nsim` have successfully completed. Some reasons for an early `next` in the loop include invalid L-moments by `are.lmom.valid` of the simluated data or invalid fitted parameters by `are.par.valid` to simulated L-moments. See the source code for more details.

Author(s)

W.H. Asquith

See Also

`lmoms`, `pmoms`, `par2qua`, `genci.simple`, `qua2ci.cov`

Examples

```
## Not run:
# It is well known that standard deviation (sigma) of the
# sample mean is equal to sigma/sample_size. Let is look at the
# quantile distribution of the median (f=0.5)
mean   <- 0; sigma <- 100
parent <- vec2par(c(mean,sigma), type='nor')
CI     <- qua2ci.simple(0.5, parent, n=10, nsim=20)
# Theoretrical sample mean sigma = 100/10 = 10
# L-moment theory: L-scale * sqrt(pi) = sigma
# Thus, it follows that the quantity
CI$elmoms$lambdas[2]/sqrt(pi)
# approaches 10 as nsim --> Inf.
## End(Not run)

# Another example.
D   <- c(123, 34, 4, 654, 37, 78, 93, 95, 120) # fake sample
lmr <- lmoms(D)    # compute the L-moments of the data
WEI <- parwei(lmr) # estimate Weibull distribution parameters
CI  <- qua2ci.simple(0.75,WEI,20, nsim=20, level=0.95)
# CI contains the estimate 95-percent confidence interval for the
# 75th-percentile of the parent Weibull distribution for 20 sample size 20.
## Not run:
pdf("Substantial_qua2ci_example.pdf")
level <- 0.90; cilo <- (1-level)/2; cihi <- 1 - cilo
para <- lmom2par(vec2lmom(c(180,50,0.75)), type="gev")
A <- qua2ci.simple(0.98, para, 30, edist="gno", level=level, nsim=3000)
Apara <- A$epara; Aenv <- A$empdist
Bpara <- lmom2par(A$elmoms, type="aep4")

lo <- log10(A$lwr); hi <- log10(A$upr)
xs <- 10^(seq(lo-0.2, hi+0.2, by=0.005))
lo <- A$lwr; hi <- A$upr; xm <- A$true; sbar <- mean(Aenv$simquas)
dd <- density(Aenv$simquas, adjust=0.5)
pk <- max(dd$y, dlmomco(xs, Apara), dlmomco(xs, Bpara))
dx <- dd$x[dd$x >= Aenv$empir.dist.lower & dd$x <= Aenv$empir.dist.upper]
dy <- dd$y[dd$x >= Aenv$empir.dist.lower & dd$x <= Aenv$empir.dist.upper]
```

```
dx <- c(dx[1], dx, dx[length(dx)]); dy <- c(0, dy, 0)

plot(c(0), c(0), type="n", xlim=range(xs), ylim=c(0,pk),
                 xlab="X VALUE", ylab="PROBABILITY DENSITY")
polygon(dx, dy, col=8)
lines(xs, dlmomco(xs, Apara)); lines(xs, dlmomco(xs, Bpara), col=2, lwd=2)
lines(dd, lty=2, lwd=2, col=8)
lines(xs, dlmomco(xs, para), col=6); lines(c(xm,xm), c(0,pk), lty=4, lwd=3)
lines(c(lo,lo,NA,hi,hi), c(0,pk,NA,0,pk), lty=2)

xlo <- qlmomco(cilo, Apara); xhi <- qlmomco(cihi, Apara)
points(c(xlo, xhi), c(dlmomco(xlo, Apara), dlmomco(xhi, Apara)), pch=16)
xlo <- qlmomco(cilo, Bpara); xhi <- qlmomco(cihi, Bpara)
points(c(xlo, xhi), c(dlmomco(xlo, Bpara), dlmomco(xhi, Bpara)), pch=16, col=2)
lines(rep(Aenv$empir.dist.lwr, 2), c(0,pk), lty=3, lwd=2, col=3)
lines(rep(Aenv$empir.dist.upr, 2), c(0,pk), lty=3, lwd=2, col=3)
lines(rep(Aenv$bern.smooth.lwr,2), c(0,pk), lty=3, lwd=2, col=4)
lines(rep(Aenv$bern.smooth.upr,2), c(0,pk), lty=3, lwd=2, col=4)
cat(c(  "F(true) = ",             round(plmomco(xm,   Apara), digits=2),
      "; F(mean(sim), edist) = ", round(plmomco(sbar, Apara), digits=2), "\n"), sep="")
dev.off()
## End(Not run)
## Not run:
ty <- "nor" # try running with "glo" (to get the L-skew "fit", see below)
para <- lmom2par(vec2lmom(c(-180,70,-.5)), type=ty)
f <- 0.99; n <- 41; ns <- 1000; Qtrue <- qlmomco(f, para)
Qsim1 <- replicate(ns, qlmomco(f, lmom2par(lmoms(rlmomco(n, para)), type=ty)))
Qsim2 <- qua2ci.simple(f, para, n, nsim=ns, edist="gno")
Qbar1 <- mean(Qsim1); Qbar2 <- mean(Qsim2$empdist$simquas)
epara <- Qsim2$epara; FT <- plmomco(Qtrue, epara)
F1 <- plmomco(Qbar1, epara); F2 <- plmomco(Qbar2, epara)
cat(c(  "F(true) = ",      round(FT, digits=2),
      "; F(via sim.) = ",  round(F1, digits=2),
      "; F(via edist) = ", round(F2, digits=2), "\n"), sep="")
# The given L-moments are highly skewed, but a Normal distribution is fit so
# L-skew is ignored. The game is deep tail (f=0.99) estimation. The true value of the
# quantile has a percentile on the error distribution 0.48 that is almost exactly 0.5
# (median = mean = symmetrical error distribution).  A test run shows nice behavior:
# F(true) =  0.48; F(via sim.) =  0.49; F(via edist) =  0.5
# But another run with ty <- "glo" (see how 0.36 << [0.52, 0.54]) has
# F(true) =  0.36; F(via sim.) =  0.54; F(via edist) =  0.52
# So as the asymmetry becomes extreme, the error distribution becomes asymmetrical too.
## End(Not run)
```

---

quaaep4 *Quantile Function of the 4-Parameter Asymmetric Exponential Power Distribution*

Description

This function computes the quantiles of the 4-parameter Asymmetric Exponential Power distribution given parameters (ξ, α, κ, and h) of the distribution computed by paraep4. The quantile function of the distribution given the cumulative distribution function $F(x)$ for $F < F(\xi)$ is

$$x(F) = \xi - \alpha\kappa\left[\gamma^{(-1)}\big((1+\kappa^2)F/\kappa^2,\ 1/h\big)\right]^{1/h},$$

and for $F \geq F(\xi)$ is

$$x(F) = \xi + \frac{\alpha}{\kappa}\left[\gamma^{(-1)}\big((1+\kappa^2)(1-F),\ 1/h\big)\right]^{1/h},$$

where $x(F)$ is the quantile x for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, κ is a shape parameter, h is another shape parameter, $\gamma^{(-1)}(Z, shape)$ is the inverse of the upper tail of the incomplete gamma function. The range of the distribution is $-\infty < x < \infty$. The inverse upper tail of the incomplete gamma function is `qgamma(Z, shape, lower.tail=FALSE)` in R. The mathematical definition of the upper tail of the incomplete gamma function shown in documentation for `cdfaep4`. If the τ_3 of the distribution is zero (symmetrical), then the distribution is known as the Exponential Power (see `lmrdia46`).

Usage

```
quaaep4(f, para, paracheck=TRUE)
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$).
`para`	The parameters from `paraep4` or similar.
`paracheck`	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

Delicado, P., and Goria, M.N., 2008, A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution: Computational Statistics and Data Analysis, v. 52, no. 3, pp. 1661–1673.

See Also

`cdfaep4`, `pdfaep4`, `lmomaep4`, `paraep4`

Examples

```
para <- vec2par(c(0,1, 0.5, 2), type="aep4");
IQR <- quaaep4(0.75,para) - quaaep4(0.25,para);
cat("Interquartile Range=",IQR,"\n")

## Not run:
F <- c(0.00001, 0.0001, 0.001, seq(0.01, 0.99, by=0.01),
       0.999, 0.9999, 0.99999);
delx <- 0.1;
x <- seq(-10,10, by=delx);
K <- .67

PAR <- list(para=c(0,1, K, 0.5), type="aep4");
plot(x,cdfaep4(x, PAR), type="n",
     ylab="NONEXCEEDANCE PROBABILITY",
     ylim=c(0,1), xlim=c(-20,20));
lines(x,cdfaep4(x,PAR), lwd=3);
lines(quaaep4(F, PAR), F, col=4);

PAR <- list(para=c(0,1, K, 1), type="aep4");
lines(x,cdfaep4(x, PAR), lty=2, lwd=3);
lines(quaaep4(F, PAR), F, col=4, lty=2);

PAR <- list(para=c(0,1, K, 2), type="aep4");
lines(x,cdfaep4(x, PAR), lty=3, lwd=3);
lines(quaaep4(F, PAR), F, col=4, lty=3);

PAR <- list(para=c(0,1, K, 4), type="aep4");
lines(x,cdfaep4(x, PAR), lty=4, lwd=3);
lines(quaaep4(F, PAR), F, col=4, lty=4);

## End(Not run)
```

---

quaaep4kapmix — *Quantile Function Mixture Between the 4-Parameter Asymmetric Exponential Power and Kappa Distributions*

Description

This function computes the quantiles of a mixture as needed between the 4-parameter Asymmetric Exponential Power (AEP4) and Kappa distributions given L-moments (lmoms). The quantile function of a two-distribution mixture is supported by par2qua2 and is

$$x(F) = (1 - w) \times A(F) + w \times K(F),$$

where $x(F)$ is the mixture for nonexceedance probability F, $A(F)$ is the AEP4 quantile function (quaaep4), $K(F)$ is the Kappa quantile function (quakap), and w is a weight factor.

Now, the above mixture is only applied if the τ_4 for the given τ_3 is within the overlapping region of the AEP4 and Kappa distributions. For this condition, the w is computed by proration between the upper Kappa distribution bound (same as the τ_3 and τ_4 of the Generalized Logistic distribution, see lmrdia) and the lower bounds of the AEP4. For τ_4 above the Kappa, then the AEP4 is exclusive and conversely, for τ_4 below the AEP4, then the Kappa is exclusive.

The w therefore is the proration

$$w = [\tau_4^K(\hat{\tau}_3) - \hat{\tau}_4]/[\tau_4^K(\hat{\tau}_3) - \tau_4^A(\hat{\tau}_3)],$$

where $\hat{\tau}_4$ is the sample L-kurtosis, τ_4^K is the upper bounds of the Kappa and τ_4^A is the lower bounds of the AEP4 for the sample L-skew ($\hat{\tau}_3$).

The parameter estimation for the AEP4 by paraep4 can fall back to pure Kappa if argument `kapapproved=TRUE` is set. Such a fall back is unrelated to the mixture described here.

Usage

```
quaaep4kapmix(f, lmom, checklmom=TRUE)
```

Arguments

`f`	Nonexceedance probability ($0 \le F \le 1$).
`lmom`	A L-moment object created by lmoms or similar.
`checklmom`	Should the `lmom` be checked for validity using the are.lmom.valid function. Normally this should be left as the default and it is very unlikely that the L-moments will not be viable (particularly in the τ_4 and τ_3 inequality). However, for some circumstances or large simulation exercises then one might want to bypass this check.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2014, Parameter estimation for the 4-parameter asymmetric exponential power distribution by the method of L-moments using R: Computational Statistics and Data Analysis, v. 71, pp. 955–970.

See Also

par2qua2, quaaep4, quakap, paraep4, parkap

Examples

```
## Not run:
FF <- c(0.0001, 0.0005, 0.001, seq(0.01,0.99, by=0.01), 0.999,
       0.9995, 0.9999); Z <- qnorm(FF)
t3s <- seq(0, 0.5, by=0.1); T4step <- 0.02
pdf("mixture_test.pdf")
for(t3 in t3s) {
   T4low <- (5*t3^2 - 1)/4; T4kapup <- (5*t3^2 + 1)/6
   t4s <- seq(T4low+T4step, T4kapup+2*T4step, by=T4step)
   for(t4 in t4s) {
      lmr <- vec2lmom(c(0,1,t3,t4)) # make L-moments for lmomco
      if(! are.lmom.valid(lmr)) next # for general protection
      kap  <- parkap(lmr)
      if(kap$ifail == 5) next # avoid further work if numeric problems
      aep4 <- paraep4(lmr, method="A")
      X <- quaaep4kapmix(FF, lmr)
      if(is.null(X)) next # one last protection
      plot(Z, X, type="l", lwd=5, col=1, ylim=c(-15,15),
           xlab="STANDARD NORMAL VARIATE",
           ylab="VARIABLE VALUE")
      mtext(paste("L-skew =",lmr$ratios[3],
                  "  L-kurtosis = ",lmr$ratios[4]))
      # Now add two more quantile functions for reference and review
      # of the mixture. These of course would not be done in practice
      # only quaaep4kapmix() would suffice.
      if(! as.logical(aep4$ifail)) {
         lines(Z, qlmomco(F,aep4), lwd=2, col=2)
      }
      if(! as.logical(kap$ifail)) {
         lines(Z, qlmomco(F,kap),  lwd=2, col=3)
      }
      message("t3=",t3,"  t4=",t4) # stout for a log file
  }
}
dev.off()

## End(Not run)
```

---

quaben *Quantile Function of the Benford Distribution*

Description

This function computes the quantiles of the Benford distribution (Benford's Law) given parameter defining the number of first M-significant figures and the numeric base. The quantile function has no analytical form and summation of the probability mass function (to form the cumulative distribution function, see also cdfben) is used with clever use of the `cut()` function.

Usage

```
quaben(f, para=list(para=c(1, 10)), ...)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The number of the first M-significant digits followed by the numerical base (only base10 supported) and the list structure mimics similar uses of the **lmomco** list structure. Default are the first significant digits and hence the digits 1 through 9.
...	Additional arguments to pass (not likely to be needed but changes in base handling might need this).

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Benford, F., 1938, The law of anomalous numbers: Proceedings of the American Philosophical Society, v. 78, no. 4, pp. 551–572, https://www.jstor.org/stable/984802.

Goodman, W., 2016, The promises and pitfalls of Benford's law: Significance (Magazine), June 2015, pp. 38–41, doi:10.1111/j.17409713.2016.00919.x.

See Also

cdfben, pmfben

Examples

```
para <- list(para=c(1, 10))
quaben(    cdfben(  5, para=para) , para=para) # 5
quaben(sum(pmfben(1:5, para=para)), para=para) # 5
```

---

quacau — *Quantile Function of the Cauchy Distribution*

Description

This function computes the quantiles of the Cauchy distribution given parameters (ξ and α) of the distribution provided by parcau. The quantile function of the distribution is

$$x(F) = \xi + \alpha \times \tan\big(\pi(F - 0.5)\big),$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter and α is a scale parameter. The quantile function of the Cauchy distribution is supported by R function qcauchy. This function does not use qcauchy because qcauchy does not return Inf for $F = 1$ although it returns -Inf for $F = 0$.

Usage

```
quacau(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parcau or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the distribution quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

Gilchirst, W.G., 2000, Statistical modeling with quantile functions: Chapman and Hall/CRC, Boca Raton, FL.

See Also

cdfcau, pdfcau, lmomcau, parcau

Examples

```
para <- c(12,12)
quacau(.5,vec2par(para,type='cau'))
```

---

quaemu *Quantile Function of the Eta-Mu Distribution*

Description

This function computes the quantiles of the Eta-Mu ($\eta : \mu$) distribution given η and μ) computed by paremu. The quantile function is complex and numerical rooting of the cumulative distribution function (cdfemu) is used.

Usage

```
quaemu(f, para, paracheck=TRUE, yacoubsintegral=TRUE, eps=1e-7)
```

Arguments

f
: Nonexceedance probability ($0 \le F \le 1$).

para
: The parameters from paremu or vec2par.

paracheck
: A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

yacoubsintegral
: A logical controlling whether the integral by Yacoub (2007) is used for the cumulative distribution function instead of numerical integration of pdfemu.

eps
: A close-enough error term for the recursion process.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

cdfemu, pdfemu, lmomemu, paremu

Examples

```
## Not run:
quaemu(0.75,vec2par(c(0.9, 1.5), type="emu")) #
## End(Not run)
```

---

quaexp *Quantile Function of the Exponential Distribution*

Description

This function computes the quantiles of the Exponential distribution given parameters (ξ and α) computed by parexp. The quantile function is

$$x(F) = \xi - \alpha \log(1 - F),$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, and α is a scale parameter.

Usage

```
quaexp(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parexp or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfexp, pdfexp, lmomexp, parexp

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quaexp(0.5,parexp(lmr))
```

quagam *Quantile Function of the Gamma Distribution*

Description

This function computes the quantiles of the Gamma distribution given parameters (α and β) computed by pargam. The quantile function has no explicit form. See the qgamma function of R and cdfgam. The parameters have the following interpretations: α is a shape parameter and β is a scale parameter in the R syntax of the qgamma() function.

Alternatively, a three-parameter version is available following the parameterization of the Generalized Gamma distribution used in the **gamlss.dist** package and for **lmomco** is documented under pdfgam. The three parameter version is automatically triggered if the length of the para element is three and not two.

Usage

```
quagam(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargam or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgam, pdfgam, lmomgam, pargam

Examples

```
  lmr <- lmoms(c(123,34,4,654,37,78))
  g <- pargam(lmr)
  quagam(0.5,g)
## Not run:
  # generate 50 random samples from this fitted parent
  Qsim <- rlmomco(5000,g)
  # compute the apparent gamma parameter for this parent
  gsim <- pargam(lmoms(Qsim))

## End(Not run)

## Not run:
# 3-p Generalized Gamma Distribution and gamlss.dist package parameterization
gg <- vec2par(c(2, 4, 3), type="gam")
X <- gamlss.dist::rGG(1000, mu=2, sigma=4, nu=3); FF <- nonexceeds(sig6=TRUE)
plot(qnorm(lmomco::pp(X)), sort(X), pch=16, col=8) # lets compare the two quantiles
lines(qnorm(FF), gamlss.dist::qGG(FF, mu=2, sigma=4, nu=3), lwd=6, col=3)
lines(qnorm(FF), quagam(FF, gg), col=2, lwd=2) #
## End(Not run)

## Not run:
# 3-p Generalized Gamma Distribution and gamlss.dist package parameterization
gg <- vec2par(c(7.4, 0.2, -3), type="gam")
X <- gamlss.dist::rGG(1000, mu=7.4, sigma=0.2, nu=-3); FF <- nonexceeds(sig6=TRUE)
plot(qnorm(lmomco::pp(X)), sort(X), pch=16, col=8) # lets compare the two quantiles
lines(qnorm(FF), gamlss.dist::qGG(FF, mu=7.4, sigma=0.2, nu=-3), lwd=6, col=3)
lines(qnorm(FF), quagam(FF, gg), col=2, lwd=2) #
## End(Not run)
```

---

quagdd *Quantile Function of the Gamma Difference Distribution*

Description

This function computes the quantiles of the Gamma Difference distribution (Klar, 2015) given parameters ($\alpha_1 > 0$, $\beta_1 > 0$, $\alpha_2 > 0$, $\beta_2 > 0$) computed by pargdd. The quantile function requires numerical rooting of the cumulative distribution function cdfgdd.

Usage

```
quagdd(f, para, paracheck=TRUE, silent=TRUE, ...)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargdd or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity.
silent	The argument of silent for the try() operation wrapped on integrate().
...	Additional arguments to pass.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Klar, B., 2015, A note on gamma difference distributions: Journal of Statistical Computation and Simulation v. 85, no. 18, pp. 1–8, doi:10.1080/00949655.2014.996566.

See Also

cdfgdd, pdfgdd, lmomgdd, pargdd

Examples

```
## Not run:
  para <- list(para=c(3, 0.1, 0.1, 4), type="gdd")
  quagdd(0.5, para) # [1] 26.71568
## End(Not run)

## Not run:
  p <- c(3, 1, 0.2, 2)
  NEP  <- seq(0.001, 0.999, by=0.001)
  para <- list(para=p, type="gdd")
  F1 <- runif(10000); F2 <- runif(10000)
  XX  <- sort(qgamma(F1, p[1], p[2]) - qgamma(F2, p[3], p[4])); FF  <- pp(XX)
  plot(NEP, quagdd(NEP, para), type="l", col=grey(0.8), lwd=6,
       xlab="Nonexceedance probability", ylab="Gamma difference quantile")
  lines(FF, XX, col="red") #
## End(Not run)

## Not run:
  para <- list(para=c(3, 2, 0.2, 2), type="gdd")
  nsam <- 50; nsim <- 10
  F1 <- runif(1000); F2 <- runif(1000)
  plot(c(1,4), c(0, 2), type="n", xlim=c(0.5, 4.5),
       xlab="Lmoment order", ylab="Lambda value")
  for(i in seq_len(nsim)) {
    X <- quagdd(runif(nsam), para)
    afunc <- function(par, lmr=NA) { p <- exp(par)
      tlmr <- pwm2lmom(pwm(qgamma(F1, p[1], p[2]) -
                           qgamma(F2, p[3], p[4])))
      sum((lmr$lambdas[1:4] - tlmr$lambdas[1:4])^2)
    }
    slmr <- lmoms(X, nmom=4); init.para <- c(0, 0, 0, 0)
    sara <- NULL
    try( sara <- optim( init.para, afunc, lmr=slmr ) )
    if(is.null(sara)) next
    sara$para <- exp(sara$par); sara$type <- "gdd"
```

```
    mara <- pargdd(slmr)
    points(1:4+0.1, lmomgdd(mara)$lambdas[1:4], col="red" )
    points(1:4-0.1, lmomgdd(sara)$lambdas[1:4], col="blue")
    print(lmomgdd(mara)$lambdas[1:4])
    print(lmomgdd(sara)$lambdas[1:4])
  }
  lines( 1:4, lmomgdd(para)$lambdas[1:4], col="darkgreen")
  points(1:4, lmomgdd(para)$lambdas[1:4], pch=16, col="darkgreen") #
## End(Not run)
```

---

quagep *Quantile Function of the Generalized Exponential Poisson Distribution*

Description

This function computes the quantiles of the Generalized Exponential Poisson distribution given parameters (β, κ, and h) of the distribution computed by pargep. The quantile function of the distribution is

$$x(F) = \eta^{-1} \log[1 + h^{-1} \log(1 - F^{1/\kappa}[1 - \exp(-h)])],$$

where $F(x)$ is the nonexceedance probability for quantile $x > 0$, $\eta = 1/\beta$, $\beta > 0$ is a scale parameter, $\kappa > 0$ is a shape parameter, and $h > 0$ is another shape parameter.

Usage

```
quagep(f, para, paracheck=TRUE)
```

Arguments

- `f` — Nonexceedance probability ($0 \le F \le 1$).
- `para` — The parameters from pargep or vec2par.
- `paracheck` — A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Details

If `f = 1` or is so close to unity that `NaN` in the computations of the quantile function, then the function enters into an infinite loop for which an "order of magnitude decrement" on the value of `.Machine$double.eps` is made until a numeric hit is encountered. Let η be this machine value, then $F = 1 - \eta^{1/j}$ where j is the iteration in the infinite loop. Eventually F becomes small enough that a finite value will result. This result is an estimate of the maximum numerical value the function can produce on the current running platform. This feature assists in the numerical integration of the quantile function for L-moment estimation (see expect.max.ostat). The expect.max.ostat was zealous on reporting errors related to lack of finite integration. However with the "order magnitude decrementing," then the errors in expect.max.ostat become fewer and are either

```
Error in integrate(fnb, lower, upper, subdivisions = 200L) :
  extremely bad integrand behaviour
```

or

```
Error in integrate(fnb, lower, upper, subdivisions = 200L) :
  maximum number of subdivisions reached
```

and are shown here to aid in research into Generalized Exponential Power implementation.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Barreto-Souza, W., and Cribari-Neto, F., 2009, A generalization of the exponential-Poisson distribution: Statistics and Probability, 79, pp. 2493–2500.

See Also

cdfgep, pdfgep, lmomgep, pargep

Examples

```
gep <- list(para=c(2, 1.5, 3), type="gep")
quagep(0.5, gep)
## Not run:
  pdf("gep.pdf")
  F <- nonexceeds(f01=TRUE)
  K <- seq(-1,2,by=.2); H <- seq(-1,2,by=.2)
  K <- 10^(K); H <- 10^(H)
  for(i in 1:length(K)) {
    for(j in 1:length(H)) {
      gep <- vec2par(c(2,K[i],H[j]), type="gep")
      message("(K,H): ",K[i]," ",H[j])
      plot(F, quagep(F, gep), lty=i, col=j, type="l", ylim=c(0,4),
           xlab="NONEXCEEDANCE PROBABILITY", ylab="X(F)")
      mtext(paste("(K,H): ",K[i]," ",H[j]))
    }
  }
  dev.off()

## End(Not run)
```

quagev — *Quantile Function of the Generalized Extreme Value Distribution*

Description

This function computes the quantiles of the Generalized Extreme Value distribution given parameters (ξ, α, and κ) of the distribution computed by pargev. The quantile function of the distribution is

$$x(F) = \xi + \frac{\alpha}{\kappa}\left(1 - (-\log(F))^{\kappa}\right),$$

for $\kappa \neq 0$, and

$$x(F) = \xi - \alpha \log(-\log(F)),$$

for $\kappa = 0$, where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of x is $-\infty < x \leq \xi + \alpha/\kappa$ if $k > 0$; $\xi + \alpha/\kappa \leq x < \infty$ if $\kappa \leq 0$. Note that the shape parameter κ parameterization of the distribution herein follows that in tradition by the greater L-moment community and others use a sign reversal on κ. (The **evd** package is one example.)

Usage

```
quagev(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargev or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgev, pdfgev, lmomgev, pargev

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
quagev(0.5, pargev(lmr))
```

---

quagld *Quantile Function of the Generalized Lambda Distribution*

Description

This function computes the quantiles of the Generalized Lambda distribution given parameters (ξ, α, κ, and h) of the distribution computed by pargld. The quantile function is

$$x(F) = \xi + \alpha(F^{\kappa} - (1 - F)^{h}),$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, and κ, and h are shape parameters. Note that in this parameterization, the scale term is shown in the numerator and not the denominator. This is done for **lmomco** as part of the parallel nature between distributions whose various scale parameters are shown having the same units as the location parameter.

Usage

```
quagld(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargld or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2007, L-moments and TL-moments of the generalized lambda distribution: Computational Statistics and Data Analysis, v. 51, no. 9, pp. 4484–4496.

Karian, Z.A., and Dudewicz, E.J., 2000, Fitting statistical distributions—The generalized lambda distribution and generalized bootstrap methods: CRC Press, Boca Raton, FL, 438 p.

See Also

cdfgld, pargld, lmomgld, lmomTLgld, pargld, parTLgld

Examples

```
## Not run:
  para <- vec2par(c(123,34,4,3),type="gld")
  quagld(0.5,para, paracheck=FALSE)

## End(Not run)
```

---

quaglo *Quantile Function of the Generalized Logistic Distribution*

Description

This function computes the quantiles of the Generalized Logistic distribution given parameters (ξ, α, and κ) computed by parglo. The quantile function is

$$x(F) = \xi + \frac{\alpha}{\kappa}\left(1 - \left(\frac{1-F}{F}\right)^{\kappa}\right),$$

for $\kappa \neq 0$, and

$$x(F) = \xi - \alpha \log\left(\frac{1-F}{F}\right),$$

for $\kappa = 0$, where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
quaglo(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parglo or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfglo, pdfglo, lmomglo, parglo

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quaglo(0.5,parglo(lmr))
```

---

quagno *Quantile Function of the Generalized Normal Distribution*

Description

This function computes the quantiles of the Generalized Normal (Log-Normal3) distribution given parameters (ξ, α, and κ) computed by pargno. The quantile function has no explicit form. The parameters have the following interpretations: ξ is a location parameter, α is a scale parameter, and κ is a shape parameter.

Usage

```
quagno(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargno or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgno, pdfgno, lmomgno, pargno, qualn3

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quagno(0.5,pargno(lmr))
```

---

quagov *Quantile Function of the Govindarajulu Distribution*

Description

This function computes the quantiles of the Govindarajulu distribution given parameters (ξ, α, and β) computed by pargov. The quantile function is

$$x(F) = \xi + \alpha[(\beta+1)F^{\beta} - \beta F^{\beta+1}],$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is location parameter, α is a scale parameter, and β is a shape parameter.

Usage

```
quagov(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from pargov or similar.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Chapman and Hall/CRC, Boca Raton.

Nair, N.U., Sankaran, P.G., Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

Nair, N.U., Sankaran, P.G., and Vineshkumar, B., 2012, The Govindarajulu distribution—Some Properties and applications: Communications in Statistics, Theory and Methods, 41(24), 4391–4406.

See Also

cdfgov, pdfgov, lmomgov, pargov

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quagov(0.5,pargov(lmr))
## Not run:
lmr <- lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2));
par <- pargov(lmr)# LMRQ said to have a linear mean residual quantile function.
# Let us have a look.
F <- c(0,nonexceeds(),1)
plot(F, qlmomco(F,par), type="l", lwd=3, xlab="NONEXCEEDANCE PROBABILITY",
     ylab="LIFE TIME, RESIDUAL LIFE, OR REVERSED RESIDUAL LIFE")
lines(F, rmlmomco(F,par),  col=2, lwd=4)  # heavy red line (residual life)
lines(F, rrmlmomco(F,par), col=2, lty=2)  # dashed red (reversed res. life)
lines(F, cmlmomco(F,par),  col=4)         # conditional mean (blue)
# Notice how the conditional mean attaches to the parent at F=1, but it does not
# attached at F=0 because of the none zero origin.
cmlmomco(0,par)           # 1.307143 # expected life given birth only
lmomgov(par)$lambdas[1]   # 1.307143 # expected life of the parent distribution
rmlmomco(0, par)          # 1.288989 # residual life given birth only
qlmomco(0, par)           # 0.018153 # instantaneous life given birth
# Note: qlmomco(0,par) + rmlmomco(0,par) is the E[lifetime], but rmlmomco()
# is the RESIDUAL MEAN LIFE.

## End(Not run)
```

quagpa — *Quantile Function of the Generalized Pareto Distribution*

Description

This function computes the quantiles of the Generalized Pareto distribution given parameters (ξ, α, and κ) computed by pargpa. The quantile function is

$$x(F) = \xi + \frac{\alpha}{\kappa}\left(1 - (1-F)^{\kappa}\right),$$

for $\kappa \neq 0$, and

$$x(F) = \xi - \alpha \log(1-F),$$

for $\kappa = 0$, where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of x is $\xi \leq x \leq \xi + \alpha/\kappa$ if $k > 0$; $\xi \leq x < \infty$ if $\kappa \leq 0$. Note that the shape parameter κ parameterization of the distribution herein follows that in tradition by the greater L-moment community and others use a sign reversal on κ. (The **evd** package is one example.)

Usage

```
quagpa(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargpa or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124, doi:10.1111/j.25176161.1990.tb01775.x.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgpa, pdfgpa, lmomgpa, pargpa

Examples

```
  lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
  quagpa(0.5,pargpa(lmr))

## Not run:
  # Let us compare L-moments, parameters, and 90th percentile for a simulated
  # GPA distibution of sample size 100 having the following parameters between
  # lmomco and lmom packages in R. The answers are the same.
  gpa.par <- lmomco::vec2par(c(1.02787, 4.54603, 0.07234), type="gpa")
  X <- lmomco::rlmomco(100, gpa.par)
   lmom::samlmu(X)
  lmomco::lmoms(X)
    lmom::pelgpa( lmom::samlmu(X))
  lmomco::pargpa(lmomco::lmoms(X))
    lmom::quagpa(0.90,   lmom::pelgpa(  lmom::samlmu(X)))
  lmomco::quagpa(0.90, lmomco::pargpa(lmomco::lmoms( X))) #
## End(Not run)
```

---

quagum *Quantile Function of the Gumbel Distribution*

Description

This function computes the quantiles of the Gumbel distribution given parameters (ξ and α) computed by pargum. The quantile function is

$$x(F) = \xi - \alpha \log(-\log(F)),$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, and α is a scale parameter.

Usage

```
quagum(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from pargum or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, p. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfgum, pdfgum, lmomgum, pargum

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quagum(0.5,pargum(lmr))
```

---

quakap — *Quantile Function of the Kappa Distribution*

Description

This function computes the quantiles of the Kappa distribution given parameters (ξ, α, κ, and h) computed by parkap. The quantile function is

$$x(F) = \xi + \frac{\alpha}{\kappa}\left(1 - \left(\frac{1-F^h}{h}\right)^{\kappa}\right),$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, κ is a shape parameter, and h is another shape parameter.

Usage

```
quakap(f, para, paracheck=TRUE)
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$).
`para`	The parameters from `parkap` or `vec2par`.
`paracheck`	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1994, The four-parameter kappa distribution: IBM Journal of Reserach and Development, v. 38, no. 3, pp. 251–258.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`cdfkap`, `pdfkap`, `lmomkap`, `parkap`

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78,21,32,231,23))
quakap(0.5,parkap(lmr))
```

---

quakmu — *Quantile Function of the Kappa-Mu Distribution*

Description

This function computes the quantiles of the Kappa-Mu ($\kappa : \mu$) distribution given parameters (κ and α) computed by `parkmu`. The quantile function is complex and numerical rooting of the cumulative distribution function (`cdfkmu`) is used.

Usage

```
quakmu(f, para, paracheck=TRUE, getmed=FALSE, qualo=NA, quahi=NA, verbose=FALSE,
                 marcumQ=TRUE, marcumQmethod=c("chisq", "delta", "integral"))
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$).
`para`	The parameters from `parkmu` or `vec2par`.
`paracheck`	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.
`getmed`	Same argument for `cdfkmu`. Because of nesting a `quakmu` call in `cdfkmu`, this argument and the next two are shown here are to avoid confusion in use of `...` instead. This argument should not overrided by the user.
`qualo`	A lower limit of the range of x to look for a `uniroot` of $F(x)$.
`quahi`	An upper limit of the range of x to look for a `uniroot` of $F(x)$.
`verbose`	Should alert messages be shown by `message()`?
`marcumQ`	Same argument for `cdfkmu`, which the user can set change.
`marcumQmethod`	Same argument for `cdfkmu`, which the user can set change.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Yacoub, M.D., 2007, The kappa-mu distribution and the eta-mu distribution: IEEE Antennas and Propagation Magazine, v. 49, no. 1, pp. 68–81

See Also

`cdfkmu`, `pdfkmu`, `lmomkmu`, `parkmu`

Examples

```
quakmu(0.75,vec2par(c(0.9, 1.5), type="kmu"))
```

quakur *Quantile Function of the Kumaraswamy Distribution*

Description

This function computes the quantiles $0 < x < 1$ of the Kumaraswamy distribution given parameters (α and β) computed by parkur. The quantile function is

$$x(F) = (1 - (1 - F)^{1/\beta})^{1/\alpha},$$

where $x(F)$ is the quantile for nonexceedance probability F, α is a shape parameter, and β is a shape parameter.

Usage

```
quakur(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parkur or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Jones, M.C., 2009, Kumaraswamy's distribution—A beta-type distribution with some tractability advantages: Statistical Methodology, v. 6, pp. 70–81.

See Also

cdfkur, pdfkur, lmomkur, parkur

Examples

```
lmr <- lmoms(c(0.25, 0.4, 0.6, 0.65, 0.67, 0.9))
quakur(0.5,parkur(lmr))
```

qualap *Quantile Function of the Laplace Distribution*

Description

This function computes the quantiles of the Laplace distribution given parameters (ξ and α) computed by parlap. The quantile function is

$$x(F) = \xi + \alpha \times \log(2F),$$

for $F \leq 0.5$, and

$$x(F) = \xi - \alpha \times \log(2(1 - F)),$$

for $F > 0.5$, where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, and α is a scale parameter.

Usage

```
qualap(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parlap or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: IBM Research Report RC12210, T.J. Watson Research Center, Yorktown Heights, New York.

See Also

cdflap, pdflap, lmomlap, parlap

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
qualap(0.5,parlap(lmr))
```

qualmrq — *Quantile Function of the Linear Mean Residual Quantile Function Distribution*

Description

This function computes the quantiles of the Linear Mean Residual Quantile Function distribution given parameters (μ and α) computed by parlmrq. The quantile function is

$$x(F) = -(\alpha + \mu) \times \log(1 - F) - 2\alpha \times F,$$

where $x(F)$ is the quantile for nonexceedance probability F, μ is a location parameter, and α is a scale parameter. The parameters must satisfy $\mu > 0$ and $-\mu \leq \alpha < \mu$.

Usage

```
qualmrq(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parlmrq or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

See Also

cdflmrq, pdflmrq, lmomlmrq, parlmrq

Examples

```
lmr <- lmoms(c(3, 0.05, 1.6, 1.37, 0.57, 0.36, 2.2));
par <- parlmrq(lmr)
qualmrq(0.75,par)
## Not run:
# The distribution is said to have a linear mean residual quantile function.
# Let us have a look.
F <- nonexceeds(); par <- vec2par(c(101,21), type="lmrq")
plot(F, qlmomco(F,par), type="l", lwd=3, xlab="NONEXCEEDANCE PROBABILITY",
     ylab="LIFE TIME, RESIDUAL LIFE, OR REVERSED RESIDUAL LIFE")
lines(F, rmlmomco(F,par),  col=2, lwd=4) # heavy red line (residual life)
lines(F, rrmlmomco(F,par), col=2, lty=2) # dashed red (reversed res. life)
lines(F, cmlmomco(F,par),  col=4)        # conditional mean (blue)
# Notice that the rmlmomco() is a straight line as the name of the parent
# distribution: Linear Mean Residual Quantile Distribution suggests.
# Curiously, the reversed mean residual is not linear.

## End(Not run)
```

---

qualn3 *Quantile Function of the 3-Parameter Log-Normal Distribution*

Description

This function computes the quantiles of the Log-Normal3 distribution given parameters (ζ, lower bounds; μ_{log}, location; and σ_{log}, scale) of the distribution computed by parln3. The quantile function (same as Generalized Normal distribution, quagno) is

$$x = \Phi^{(-1)}(Y),$$

where $\Phi^{(-1)}$ is the quantile function of the Standard Normal distribution and Y is

$$Y = \frac{\log(x - \zeta) - \mu_{\mathrm{log}}}{\sigma_{\mathrm{log}}},$$

where ζ is the lower bounds (real space) for which $\zeta < \lambda_1 - \lambda_2$ (checked in are.parln3.valid), μ_{log} be the mean in natural logarithmic space, and σ_{log} be the standard deviation in natural logarithm space for which $\sigma_{\mathrm{log}} > 0$ (checked in are.parln3.valid) is obvious because this parameter has an analogy to the second product moment. Letting $\eta = \exp(\mu_{\mathrm{log}})$, the parameters of the Generalized Normal are $\zeta + \eta$, $\alpha = \eta\sigma_{\mathrm{log}}$, and $\kappa = -\sigma_{\mathrm{log}}$. At this point, the algorithms (quagno) for the Generalized Normal provide the functional core.

Usage

```
qualn3(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from parln3 or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the distribution quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Note

The parameterization of the Log-Normal3 results in ready support for either a known or unknown lower bounds. More information regarding the parameter fitting and control of the ζ parameter can be seen in the Details section under parln3.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

cdfln3, pdfln3, lmomln3, parln3, quagno

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
qualn3(0.5,parln3(lmr))
```

---

quanor *Quantile Function of the Normal Distribution*

Description

This function computes the quantiles of the Normal distribution given parameters (μ and σ) computed by parnor. The quantile function has no explicit form (see cdfnor and qnorm). The parameters have the following interpretations: μ is the arithmetic mean and σ is the standard deviation. The R function qnorm is used.

Usage

```
quanor(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parnor or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfnor, pdfnor, lmomnor, parnor

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quanor(0.5,parnor(lmr))
```

quapdq3 *Quantile Function of the Polynomial Density-Quantile3 Distribution*

Description

This function computes the quantiles of the Polynomial Density-Quantile3 distribution (PDQ3) given parameters (ξ, α, and κ) computed by parpdq3. The quantile function is

$$x(F) = \xi + \alpha\left[\log\left(\frac{F}{1-F}\right) + \kappa\log\left(\frac{\left[1-\kappa(2F-1)\right]^2}{4F(1-F)}\right)\right],$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of the distribution is $-\infty < x < \infty$. This formulation of logistic distribution generalization is unique in the literature.

Usage

```
quapdq3(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from parpdq3 or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Details

The PDQ3 was proposed by Hosking (2007) with the core justification of maximizing entropy and that “maximizing entropy subject to a set of constraints can be regarded as deriving a distribution that is consistent with the information specified in the constraints while making minimal assumptions about the form of the distribution other than those embodied in the constraints.” The PDQ3 is that family constrained to the λ_1, λ_2, and τ_3 values of the L-moments. (See also the Polynomial Density-Quantile4 function for constraint on λ_1, λ_2, and τ_4 values of the L-moments, quapdq4.)

The PDQ3 has maximum entropy conditional on having specified values for the L-moments of λ_1, λ_2, and $\lambda_3 = \tau_3\lambda_2$. The tails of the PDQ3 are exponentially decreasing and the distribution could be useful in distributional analysis with data showing similar tail characteristics. The attainable L-kurtosis range is $\tau_4 = (5\tau_3/\kappa) - 1$.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

cdfpdq3, pdfpdq3, lmompdq3, parpdq3, quapdq4

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
quapdq3(0.5, parpdq3(lmr)) # [1] 51.22802

## Not run:
  FF <- seq(0.002475, 1 - 0.002475, by=0.001)
  para <- list(para=c(0.6933, 1.5495, 0.5488), type="pdq3")
  plot(log(FF/(1-FF)), quapdq3(FF, para), type="l", col=grey(0.8), lwd=4,
       xlab="Logistic variate, log(f/(1-f))", ylab="Quantile, Q(f)")
  lines(log(FF/(1-FF)), log(qf(FF, df1=7, df2=1)), lty=2)
  legend("topleft", c("log F(7,1) distribution with same L-moments",
                      "PDQ3 distribution with same L-moments as the log F(7,1)"),
         lwd=c(1, 4), lty=c(2, 1), col=c(1, grey(0.8)), cex=0.8)
  mtext("Mimic Hosking (2007, fig. 2 [right])") #
## End(Not run)
```

---

quapdq4 *Quantile Function of the Polynomial Density-Quantile4 Distribution*

Description

This function computes the quantiles of the Polynomial Density-Quantile4 distribution (PDQ4) given parameters (ξ, α, and κ) computed by parpdq4. The quantile function for $0 < \kappa < 1$ is

$$x(F) = \xi + \alpha\left[\log\left(\frac{F}{1-F}\right) - 2\kappa\ \mathrm{atanh}\big(\kappa[2F-1]\big)\right] \text{ and}$$

for $-\infty < \kappa < 0$ is

$$x(F) = \xi + \alpha\left[\log\left(\frac{F}{1-F}\right) + 2\kappa\ \mathrm{atan}\big(\kappa[2F-1]\big)\right],$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α is a scale parameter, and κ is a shape parameter. The range of the distribution is $-\infty < x < \infty$.

Usage

```
quapdq4(f, para, paracheck=TRUE)
```

Arguments

f
: Nonexceedance probability ($0 \leq F \leq 1$).

para
: The parameters from parpdq4 or vec2par.

paracheck
: A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Details

The PDQ4 was proposed by Hosking (2007) with the core justification of maximizing entropy and that "maximizing entropy subject to a set of constraints can be regarded as deriving a distribution that is consistent with the information specified in the constraints while making minimal assumptions about the form of the distribution other than those embodied in the constraints." The PDQ4 is that family constrained to the λ_1, λ_2, and τ_4 values of the L-moments. (See also the Polynomial Density-Quantile3 function for constraint on λ_1, λ_2, and τ_3 values of the L-moments, quapdq3.)

The PDQ4 is a symmetrical distribution ($\tau_3 = 0$ everywhere) that has maximum entropy conditional on having specified values for the L-moments of λ_1, λ_2, and $\lambda_4 = \tau_4\lambda_2$ with $\lambda_3 = \tau_3 = 0$. The tails of the PDQ4 are exponentially decreasing and the distribution could be useful in distributional analysis with data showing similar tail characteristics. The attainable L-kurtosis range is $-1/4 < \tau_4 < 1$ with the sign change from negative to positive of κ occurring at $\tau_4 = 1/6$. Finally, PDQ4 generalizes the logistic distribution, which is the special case $\kappa \rightarrow 0$, and contains distributions both lighter-tailed ($\kappa < 0$) and heavier-tailed ($\kappa > 0$) than the logistic.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 2007, Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics: Journal of Statistical Planning and Inference, v. 137, no. 9, pp. 2,870–2891, doi:10.1016/j.jspi.2006.10.010.

See Also

cdfpdq4, pdfpdq4, lmompdq4, parpdq4, quapdq3

Examples

```
  lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
  quapdq4(0.5, parpdq4(lmr)) # [1] 155

## Not run:
  FF <- seq(0.0001, 0.9999, by=0.001)
  para <- list(para=c(0, 0.4332, -0.7029), type="pdq4")
```

```
  plot( qnorm(FF, sd=1), quapdq4(FF, para), type="l", col=grey(0.8), lwd=4,
       xlab="Standard normal variate", ylab="Quantiles, Q(f)")
  lines(qnorm(FF, sd=1),   qnorm(FF, sd=1), lty=2)
  legend("topleft", c("Standard normal distribution",
                     "PDQ4 distribution with same L-moments as the standard normal"),
        lwd=c(1, 4), lty=c(2, 1), col=c(1, grey(0.8)), cex=0.8)
  mtext("Mimic Hosking (2007, fig. 3 [right])") #
## End(Not run)

## Not run:
  # A quick recipe to look at the shapes of quantile functions.
  FF <- seq(0.001, 0.999, by=0.001)
  plot(qnorm(FF), qnorm(FF), type="n", ylim=c(-7, 7),
       xlab="Standard normal variate", ylab="PDQ4 variate")
  abline(h=0, lty=2, lwd=0.9); abline(v=0, lty=2, lwd=0.9)

  lscale   <- 1 / sqrt(pi)
  tau4s    <- seq(-1/4, 0.7, by=.05)
  tau4s[1] <- tau4s[1] + 0.001
  for(i in 1:length(tau4s)) {
    lmr <- vec2lmom(c(0, lscale, 0, tau4s[i]))
    if(! are.lmom.valid(lmr)) next
    pdq4 <- parpdq4(lmr, snapt4uplimit=FALSE)
    lines(qnorm(FF), qlmomco(FF, pdq4), col=rgb(abs(tau4s[i]), 0, 1))
  }
  abline(0,1, col="darkgreen", lwd=3)
  txt <- "Standard normal distribution (Tau4=0.122602)"
  txt <- c(txt, paste0("PDQ4 distribution for varying Tau4 values",
                      " (color varies for accenting)"))
  legend("topleft", txt, col=c("darkgreen", rgb(0.2, 0, 1)),
                      cex=0.9, bty="n", lwd=c(3,1)) #
## End(Not run)
```

---

quape3 — *Quantile Function of the Pearson Type III Distribution*

Description

This function computes the quantiles of the Pearson Type III distribution given parameters (μ, σ, and γ) computed by `parpe3`. The quantile function has no explicit form (see `cdfpe3`).

For the implementation in the **lmomco** package, the three parameters are μ, σ, and γ for the mean, standard deviation, and skew, respectively. Therefore, the Pearson Type III distribution is of considerable theoretical interest to this package because the parameters, which are estimated via the L-moments, are in fact the product moments, although, the values fitted by the method of L-moments will not be numerically equal to the sample product moments. Further details are provided in the Examples section under `pmoms`.

Usage

```
quape3(f, para, paracheck=TRUE)
```

Arguments

f
: Nonexceedance probability ($0 \leq F \leq 1$).

para
: The parameters from `parpe3` or `vec2par`.

paracheck
: A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

`cdfpe3`, `pdfpe3`, `lmompe3`, `parpe3`

Examples

```
  lmr <- lmoms(c(123,34,4,654,37,78))
  quape3(0.5,parpe3(lmr))

## Not run:
  # Let us run an experiment on the reflection symmetric PE3.
  # Pick some parameters suitable for hydrologic applications in log.
  para_neg <- vec2par(c(3,.3,-1), type="pe3") # Notice only the
  para_pos <- vec2par(c(3,.3,+1), type="pe3") # sign change of skew.

  nsim <- 1000 # Number of simulations
  nsam <- 70   # Reasonable sample size in hydrology
  neg <- pos <- rep(NA, nsim)
  for(i in 1:nsim) {
    ff <- runif(nsam) # Ensure that each qlmomco()-->quape3() has same probs.
    neg[i] <- lmoms.cov(qlmomco(ff, para_neg), nmom=3, se="lmrse")[3]
    pos[i] <- lmoms.cov(qlmomco(ff, para_pos), nmom=3, se="lmrse")[3]
    # We have extracted the sample standard error of L-skew from the sample
    # This is not the same as the standard error of so computed PE3
    # parameters, but for the illustration here, it does not matter much.
```

```
  }
  zz <- data.frame(setau3=c(neg,pos), # preserve to make grouping boxplot
                   sign=c(rep("negskew", nsim), rep("posskew", nsim)))
  boxplot(zz$setau3~zz$sign, xlab="Sign of a '1' PE3 skew",
                              ylab="Standard error of L-skew")
  mtext("Standard Errors of 1,000 PE3 Parents (3,0.3,+/-1) (n=70)")
  # Notice that the distribution of the standard errors of L-skew are
  # basically the same whether or no the sign of the skew is reversed.
  # Finally, we make a scatter plot as a check that for any given sample
  # derived from same probabilities that the standard errors are indeed,
  # that is, remain sample specific.
  plot(neg, pos, xlab="Standard error of -1 skew simulation",
                 ylab="Standard error of +1 skew simulation")
  mtext("Standard Errors of 1,000 PE3 Parents (3,0.3,+/-1) (n=70)") #
## End(Not run)
```

---

quaray *Quantile Function of the Rayleigh Distribution*

Description

This function computes the quantiles of the Rayleigh distribution given parameters (ξ and α) computed by parray. The quantile function is

$$x(F) = \xi + \sqrt{-2\alpha^2 \log(1-F)},$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, and α is a scale parameter.

Usage

```
quaray(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parray or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1986, The theory of probability weighted moments: Research Report RC12210, IBM Research Division, Yorkton Heights, N.Y.

See Also

cdfray, pdfray, lmomray, parray

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quaray(0.5,parray(lmr))
```

---

quarevgum *Quantile Function of the Reverse Gumbel Distribution*

Description

This function computes the quantiles of the Reverse Gumbel distribution given parameters (ξ and α) computed by parrevgum. The quantile function is

$$x(F) = \xi + \alpha \log(-\log(1-F)),$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, and α is a scale parameter.

Usage

```
quarevgum(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parrevgum or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1995, The use of L-moments in the analysis of censored data, in Recent Advances in Life-Testing and Reliability, edited by N. Balakrishnan, chapter 29, CRC Press, Boca Raton, Fla., pp. 546–560.

See Also

cdfrevgum, pdfrevgum, lmomrevgum, parrevgum

Examples

```
# See p. 553 of Hosking (1995)
# Data listed in Hosking (1995, table 29.3, p. 553)
D <- c(-2.982, -2.849, -2.546, -2.350, -1.983, -1.492, -1.443,
       -1.394, -1.386, -1.269, -1.195, -1.174, -0.854, -0.620,
       -0.576, -0.548, -0.247, -0.195, -0.056, -0.013,  0.006,
        0.033,  0.037,  0.046,  0.084,  0.221,  0.245,  0.296)
D <- c(D,rep(.2960001,40-28)) # 28 values, but Hosking mentions
                              # 40 values in total
z <-  pwmRC(D,threshold=.2960001)
str(z)
# Hosking reports B-type L-moments for this sample are
# lamB1 = -.516 and lamB2 = 0.523
btypelmoms <- pwm2lmom(z$Bbetas)
# My version of R reports lamB1 = -0.5162 and lamB2 = 0.5218
str(btypelmoms)
rg.pars <- parrevgum(btypelmoms,z$zeta)
str(rg.pars)
# Hosking reports xi = 0.1636 and alpha = 0.9252 for the sample
# My version of R reports xi = 0.1635 and alpha = 0.9254
F  <- nonexceeds()
PP <- pp(D) # plotting positions of the data
plot(PP,sort(D),ylim=range(quarevgum(F,rg.pars)))
lines(F,quarevgum(F,rg.pars))
# In the plot notice how the data flat lines at the censoring level,
# but the distribution continues on.  Neat.
```

---

quarice *Quantile Function of the Rice Distribution*

Description

This function computes the quantiles of the Rice distribution given parameters (ν and α) computed by parrice. The quantile function is complex and numerical rooting of the cumulative distribution function cdfrice is used.

Usage

```
quarice(f, para, xmax=NULL, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parrice or vec2par.
xmax	The maximum x value used for integeration.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

cdfrice, pdfrice, lmomrice, parrice

Examples

```
lmr <- vec2lmom(c(125,0.20), lscale=FALSE)
quarice(0.75,parrice(lmr))
# The quantile function of the Rice as implemented in lmomco
# is slow because of rooting the CDF, which is created by
# integration of the PDF. Rician random variates are easily created.
# Thus, in speed applications the rlmomco() with a Rice parameter
# object could be bypassed by the following function, rrice().
## Not run:
"rrice" = function(n, nu, alpha) { # from the VGAM package
    theta = 1 # any number
    X = rnorm(n, mean=nu * cos(theta), sd=alpha)
    Y = rnorm(n, mean=nu * sin(theta), sd=alpha)
    return(sqrt(X^2 + Y^2))
}
n <- 5000; # suggest making it about 10,000
nu <- 100; alpha <- 10
set.seed(501); lmoms(rrice(n, nu, alpha))
set.seed(501); lmoms(rlmomco(n, vec2par(c(nu,alpha), type='rice')))
# There are slight numerical differences between the two?

## End(Not run)
```

quasla *Quantile Function of the Slash Distribution*

Description

This function computes the quantiles of the Slash distribution given parameters (ξ and α) provided by parsla. The quantile function $x(F; \xi, \alpha)$ for nonexceedance probability F and where ξ is a location parameter and α is a scale parameter is complex and requires numerical optimization of the cumulative distribution function (cdfsla).

Usage

```
quasla(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parsla or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Rogers, W.H., and Tukey, J.W., 1972, Understanding some long-tailed symmetrical distributions: Statistica Neerlandica, v. 26, no. 3, pp. 211–226.

See Also

cdfsla, pdfsla, lmomsla, parsla

Examples

```
para <- c(12,1.2)
quasla(0.55,vec2par(para,type='sla'))
```

quasmd *Quantile Function of the Singh–Maddala Distribution*

Description

This function computes the quantiles of the Singh–Maddala (Burr Type XII) distribution given parameters (ξ, a, b, and q) computed by parsmd. The quantile function is

$$x(F) = \xi + a\biggl((1-F)^{-1/q} - 1\biggr)^{1/b},$$

where $x(F)$ with $0 \le x \le \infty$ is the quantile for nonexceedance probability F, ξ is a location parameter, a is a scale parameter ($a > 0$), b is a shape parameter ($b > 0$), and q is another shape parameter ($q > 0$).

Usage

```
quasmd(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from parsmd or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Kumar, D., 2017, The Singh–Maddala distribution—Properties and estimation: International Journal of System Assurance Engineering and Management, v. 8, no. S2, 15 p., doi:10.1007/s13198-01706001.

Shahzad, M.N., and Zahid, A., 2013, Parameter estimation of Singh Maddala distribution by moments: International Journal of Advanced Statistics and Probability, v. 1, no. 3, pp. 121–131, doi:10.14419/ijasp.v1i3.1206.

See Also

cdfsmd, pdfsmd, lmomsmd, parsmd

Examples

```
quasmd(0.99, parsmd(vec2lmom(c(155, 118.6, 0.6, 0.45)))) # 1547.337 99th percentile
```

---

quast3 *Quantile Function of the 3-Parameter Student t Distribution*

Description

This function computes the quantiles of the 3-parameter Student t distribution given parameters (ξ, α, ν) computed by `parst3`. There is no explicit solution for the quantile function for nonexceedance probability `F` but built-in R functions can be used. The implementation is `U` = ξ and `A` = α for $1.001 \le \nu \le 10^5.5$, one can use `U + A*qt(F, N)` where `qt` is the 1-parameter Student t quantile function. The numerically accessible range of implementation here and consistency to the τ_4 and τ_6 is $10.001 \le \nu \le 10^5.5$. The limits for ν stem from study of ability for theoretical integration of the quantile function to produce viable τ_4 and τ_6 (see `inst/doc/t4t6/studyST3.R`).

Usage

```
quast3(f, para, paracheck=TRUE)
```

Arguments

`f`	Nonexceedance probability ($0 \le F \le 1$).
`para`	The parameters from `parst3` or `vec2par`.
`paracheck`	A logical on whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

`cdfst3`, `pdfst3`, `lmomst3`, `parst3`

Examples

```
lmr <- lmoms(c(123, 34, 4, 654, 37, 78))
quast3(0.75, parst3(lmr))
```

quatexp *Quantile Function of the Truncated Exponential Distribution*

Description

This function computes the quantiles of the Truncated Exponential distribution given parameters (ψ and α) computed by partexp. The parameter ψ is the right truncation, and α is a scale parameter. The quantile function, letting $\beta = 1/\alpha$ to match nomenclature of Vogel and others (2008), is

$$x(F) = -\frac{1}{\beta}\log(1 - F[1 - \exp(-\beta\psi)]),$$

where $x(F)$ is the quantile $0 \le x \le \psi$ for nonexceedance probability F and $\psi > 0$ and $\alpha > 0$. This distribution represents a nonstationary Poisson process.

The distribution is restricted to a narrow range of L-CV ($\tau_2 = \lambda_2/\lambda_1$). If $\tau_2 = 1/3$, the process represented is a stationary Poisson for which the quantile function is simply the uniform distribution and $x(F) = \psi\, F$. If $\tau_2 = 1/2$, then the distribution is represented as the usual exponential distribution with a location parameter of zero and a scale parameter $1/\beta$. Both of these limiting conditions are supported.

Usage

```
quatexp(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from partexp or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Vogel, R.M., Hosking, J.R.M., Elphick, C.S., Roberts, D.L., and Reed, J.M., 2008, Goodness of fit of probability distributions for sightings as species approach extinction: Bulletin of Mathematical Biology, DOI 10.1007/s11538-008-9377-3, 19 p.

See Also

cdftexp, pdftexp, lmomtexp, partexp

Examples

```
lmr <- vec2lmom(c(40,0.38), lscale=FALSE)
quatexp(0.5,partexp(lmr))
## Not run:
F <- seq(0,1,by=0.001)
A <- partexp(vec2lmom(c(100, 1/2), lscale=FALSE))
plot(qnorm(F), quatexp(F, A), pch=16, type='l')
by <- 0.01; lcvs <- c(1/3, seq(1/3+by, 1/2-by, by=by), 1/2)
reds <- (lcvs - 1/3)/max(lcvs - 1/3)
for(lcv in lcvs) {
    A <- partexp(vec2lmom(c(100, lcv), lscale=FALSE))
    lines(qnorm(F), quatexp(F, A), col=rgb(reds[lcvs == lcv],0,0))
}

## End(Not run)
```

---

quatri *Quantile Function of the Asymmetric Triangular Distribution*

Description

This function computes the quantiles of the Asymmetric Triangular distribution given parameters (ν, ω, and ψ) of the distribution computed by partri. The quantile function of the distribution is

$$x(F) = \nu + \sqrt{(\psi - \nu)(\omega - \nu)F},$$

for $F < P$,

$$x(F) = \psi - \sqrt{(\psi - \nu)(\psi - \omega)(1 - F)},$$

for $F > P$, and

$$x(F) = \omega,$$

for $F = P$ where $x(F)$ is the quantile for nonexceedance probability F, ν is the minimum, ψ is the maximum, and ω is the mode of the distribution and

$$P = \frac{(\omega - \nu)}{(\psi - \nu)}.$$

Usage

```
quatri(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from partri or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

See Also

cdftri, pdftri, lmomtri, partri

Examples

```
lmr <- lmoms(c(46, 70, 59, 36, 71, 48, 46, 63, 35, 52))
quatri(0.5,partri(lmr))
```

---

quawak *Quantile Function of the Wakeby Distribution*

Description

This function computes the quantiles of the Wakeby distribution given parameters (ξ, α, β, γ, and δ) computed by parwak. The quantile function is

$$x(F) = \xi + \frac{\alpha}{\beta}(1 - (1 - F)^{\beta}) - \frac{\gamma}{\delta}(1 - (1 - F))^{-\delta},$$

where $x(F)$ is the quantile for nonexceedance probability F, ξ is a location parameter, α and β are scale parameters, and γ and δ are shape parameters. The five returned parameters from parwak in order are ξ, α, β, γ, and δ.

Usage

```
quawak(f, wakpara, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
wakpara	The parameters from parwak or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

Hosking, J.R.M., 1996, FORTRAN routines for use with the method of L-moments: Version 3, IBM Research Report RC20525, T.J. Watson Research Center, Yorktown Heights, New York.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfwak, pdfwak, lmomwak, parwak

Examples

```
lmr <- lmoms(c(123,34,4,654,37,78))
quawak(0.5,parwak(lmr))
```

---

quawei *Quantile Function of the Weibull Distribution*

Description

This function computes the quantiles of the Weibull distribution given parameters (ζ, β, and δ) computed by parwei. The quantile function is

$$x(F) = \beta[-\log(1-F)]^{1/\delta} - \zeta,$$

where $x(F)$ is the quantile for nonexceedance probability F, ζ is a location parameter, β is a scale parameter, and δ is a shape parameter.

The Weibull distribution is a reverse Generalized Extreme Value distribution. As result, the Generalized Extreme Value algorithms are used for implementation of the Weibull in **lmomco**. The relations between the Generalized Extreme Value distribution parameters (ξ, α, κ) are κ) is $\kappa = 1/\delta$, $\alpha = \beta/\delta$, and $\xi = \zeta - \beta$. These relations are taken from Hosking and Wallis (1997).

In R, the quantile function of the Weibull distribution is `qweibull`. Given a Weibull parameter object `p`, the R syntax is `qweibull(f, p$para[3], scale=p$para[2]) - p$para[1]`. For the current implementation for this package, the reversed Generalized Extreme Value distribution quagev is used and the implementation is `-quagev((1-f),para)`.

Usage

```
quawei(f, para, paracheck=TRUE)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from parwei or vec2par.
paracheck	A logical controlling whether the parameters are checked for validity. Overriding of this check might be extremely important and needed for use of the quantile function in the context of TL-moments with nonzero trimming.

Value

Quantile value for nonexceedance probability F.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge University Press.

See Also

cdfwei, pdfwei, lmomwei, parwei

Examples

```
# Evaluate Weibull deployed here and within R (qweibull)
lmr <- lmoms(c(123,34,4,654,37,78))
WEI <- parwei(lmr)
Q1  <- quawei(0.5,WEI)
Q2  <- qweibull(0.5,shape=WEI$para[3],scale=WEI$para[2])-WEI$para[1]
if(Q1 == Q2) EQUAL <- TRUE

# The Weibull is a reversed generalized extreme value
Q <- sort(rlmomco(34,WEI)) # generate Weibull sample
lm1 <- lmoms(Q)    # regular L-moments
lm2 <- lmoms(-Q)   # L-moment of negated (reversed) data
WEI <- parwei(lm1) # parameters of Weibull
GEV <- pargev(lm2) # parameters of GEV
F <- nonexceeds()  # Get a vector of nonexceedance probs
plot(pp(Q),Q)
lines(F,quawei(F,WEI))
lines(F,-quagev(1-F,GEV),col=2) # line over laps previous
```

ralmomco *Alpha-Percentile Residual Quantile Function of the Distributions*

Description

This function computes the α-Percentile Residual Quantile Function for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair and Vineshkumar (2011, p. 85) and Nair et al. (2013, p. 56) as

$$P_\alpha(u) = x(1 - [1-\alpha][1-u]) - x(u),$$

where $P_\alpha(u)$ is the α-percentile residual quantile for nonexceedance probability u and percentile α and $x(u)$ is a constant for $x(F = u)$. The reversed α-percentile residual quantile is available under rralmomco.

Usage

```
ralmomco(f, para, alpha=0)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.
alpha	The α percentile, which is divided by 100 inside the function ahead of calling the quantile function of the distribution.

Value

α-percentile residual quantile value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., and Vineshkumar, B., 2011, Reversed percentile residual life and related concepts: Journal of the Korean Statistical Society, v. 40, no. 1, pp. 85–92.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rmlmomco, rralmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
maximum.lifetime <- quagov(1,A) # 2649 days
ralmomco(0,A,alpha=0)   #    0 days
ralmomco(0,A,alpha=100) # 2649 days
ralmomco(1,A,alpha=0)   #    0 days (death certain)
ralmomco(1,A,alpha=100) #    0 days (death certain)
## Not run:
F <- nonexceeds(f01=TRUE)
plot(F, qlmomco(F,A), type="l",
     xlab="NONEXCEEDANCE PROBABILITY", ylab="LIFETIME, IN DAYS")
lines(F, rmlmomco(F, A), col=4, lwd=4) # thick blue, residual mean life
lines(F, ralmomco(F, A, alpha=50), col=2) # solid red, median residual life
lines(F, ralmomco(F, A, alpha=10), col=2, lty=2) # lower dashed line,
                                              # the 10th percentile of residual life
lines(F, ralmomco(F, A, alpha=90), col=2, lty=2) # upper dashed line,
                                              # 10th percentile of residual life
## End(Not run)
```

`reslife.lmoms` *L-moments of Residual Life*

Description

This function computes the L-moments of residual life for a quantile function $x(F)$ for an exceedance threshold in probabiliy of u. The L-moments of residual life are thoroughly described by Nair et al. (2013, p. 202). These L-moments are define as

$$\lambda(u)_r = \sum_{k=0}^{r-1} (-1)^k \binom{r-1}{k}^2 \int_u^1 \left(\frac{p-u}{1-u}\right)^{r-k-1} \left(\frac{1-p}{1-u}\right)^k \frac{x(p)}{1-u}\,\mathrm{d}p,$$

where $\lambda(u)_r$ is the rth L-moment at residual life probability u. The L-moment ratios $\tau(u)_r$ have the usual definitions. The implementation here exclusively uses the quantile function of the distribution. If $u = 0$, then the usual L-moments of the quantile function are returned because the integration domain is the entire potential lifetime range. If $u = 1$, then $\lambda(1)_1 = x(1)$ is returned, which is the maximum lifetime of the distribution (the value for the upper support of the distribution), and the remaining $\lambda(1)_r$ for $r \geq 2$ are set to NA. Lastly, the notation (u) is neither super or subscripted to avoid confusion with L-moment order r or the TL-moments that indicate trimming level as a superscript (see `TLmoms`).

Usage

```
reslife.lmoms(f, para, nmom=5)
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$).
`para`	The parameters from `lmom2par` or `vec2par`.
`nmom`	The number of moments to compute. Default is 5.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.
`life.exceeds`	The value for $x(F)$ for $F =$ `f`.
`life.percentile`	The value $100\times$`f`.
`trim`	Level of symmetrical trimming used in the computation, which is `NULL` because no trimming theory for L-moments of residual life have been developed or researched.
`leftrim`	Level of left-tail trimming used in the computation, which is `NULL` because no trimming theory for L-moments of residual life have been developed or researched.
`rightrim`	Level of right-tail trimming used in the computation, which is `NULL` because no trimming theory for L-moments of residual life have been developed or researched.
`source`	An attribute identifying the computational source of the L-moments: "reslife.lmoms".

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

`rmlmomco`, `rreslife.lmoms`

Examples

```
A <- vec2par(c(230, 2649, 3), type="gov") # Set lower bounds = 230 hours
F <- nonexceeds(f01=TRUE)
plot(F, rmlmomco(F,A), type="l", ylim=c(0,3000), # mean residual life [black]
     xlab="NONEXCEEDANCE PROBABILITY",
     ylab="LIFE, RESIDUAL LIFE (RL), RL_L-SCALE, RL_L-skew (rescaled)")
L1 <- L2 <- T3 <- vector(mode="numeric", length=length(F))
for(i in 1:length(F)) {
  lmr <- reslife.lmoms(F[i], A, nmom=3)
  L1[i] <- lmr$lambdas[1]; L2[i] <- lmr$lambdas[2]; T3[i] <- lmr$ratios[3]
}
lines(c(0,1), c(1500,1500),  lty=2) # Origin line (to highlight T3 crossing "zero")
lines(F, L1,          col=2, lwd=3) # Mean life (not residual, that is M(u)) [red]
```

```
lines(F, L2,          col=3, lwd=3) # L-scale of residual life [green]
lines(F, 5E3*T3+1500, col=4, lwd=3) # L-skew of residual life (re-scaled) [blue]
## Not run:
# Nair et al. (2013, p. 203), test shows L2(u=0.37) = 771.2815
A <- vec2par(c(230, 2649, 0.3), type="gpa"); F <- 0.37
"afunc" <- function(p) { return((1-p)*rmlmomco(p,A)) }
L2u1 <- (1-F)^(-2)*integrate(afunc,F,1)$value
L2u2 <- reslife.lmoms(F,A)$lambdas[2]

## End(Not run)
```

---

riglmomco *Income Gap Ratio Quantile Function for the Distributions*

Description

This function computes the Income Gap Ratio for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair et al. (2013, p. 230) as

$$G(u) = 1 - \frac{{}_{\mathrm{r}}\lambda_1(u)}{x(u)},$$

where $G(u)$ is the income gap quantile for nonexceedance probability u, $x(u)$ is a constant for $x(F = u)$ is the quantile for u, and ${}_{\mathrm{r}}\lambda_1(u)$ is the 1st reversed residual life L-moment (rreslife.lmoms).

Usage

```
riglmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from lmom2par or vec2par.

Value

Income gap ratio quantile value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rreslife.lmoms

Examples

```
# Let us parametize some "income" distribution.
A <- vec2par(c(123, 264, 2.11), type="gov")
riglmomco(0.5, A)
## Not run:
F <- nonexceeds(f01=TRUE)
plot(F, riglmomco(F,A), type="l",
     xlab="NONEXCEEDANCE PROBABILITY", ylab="INCOME GAP RATIO")
## End(Not run)
```

---

rlmomco *Random Variates of a Distribution*

Description

This function generates random variates for the specified distribution in the parameter object argument. See documentation about the parameter object is seen in `lmom2par` or `vec2par`. The prepended `r` in the function name is to parallel the built-in distribution syntax of R but of course reflects the **lmomco** name in the function. An assumption is made that the user knows that they are providing appropirate (that is valid) distribution parameters. This is evident by the

`paracheck = FALSE`

argument passed to the `par2qua` function.

Usage

```
rlmomco(n, para)
```

Arguments

`n`	Number of samples to generate
`para`	The parameters from `lmom2par` or similar.

Value

Vector of quantile values.

Note

The action of this function in R idiom is `par2qua(runif(n), para)` for the distribution parameters `para`, the R function `runif` is the Uniform distribution, and `n` being the simulation size.

Author(s)

W.H. Asquith

See Also

dlmomco, plmomco, qlmomco, slmomco

Examples

```
lmr      <- lmoms(rnorm(20)) # generate 20 standard normal variates
para     <- parnor(lmr) # estimate parameters of the normal
simulate <- rlmomco(20,para) # simulate 20 samples using lmomco package

lmr  <- vec2lmom(c(1000,500,.3)) # first three lmoments are known
para <- lmom2par(lmr,type="gev") # est. parameters of GEV distribution
Q    <- rlmomco(45,para) # simulate 45 samples
PP   <- pp(Q)            # compute the plotting positions
plot(PP,sort(Q))         # plot the data up
```

---

rmlmomco *Mean Residual Quantile Function of the Distributions*

Description

This function computes the Mean Residual Quantile Function for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair et al. (2013, p. 51) as

$$M(u) = \frac{1}{1-u}\int_u^1 [x(p) - x(u)]\; \mathrm{d}p,$$

where $M(u)$ is the mean residual quantile for nonexceedance probability u and $x(u)$ is a constant for $x(F = u)$. The variance of $M(u)$ is provided in rmvarlmomco.

The integration instead of from $0 \rightarrow 1$ for the usual quantile function is $u \rightarrow 1$. Note that $x(u)$ is a constant, so

$$M(u) = \frac{1}{1-u}\int_u^1 x(p)\; \mathrm{d}p - x(u),$$

is equivalent and the basis for the implementation in rmlmomco. Assuming that $x(F)$ is a life distribution, the $M(u)$ is interpreted (see Nair et al. [2013, p. 51]) as the average remaining life beyond the $100(1-F)\%$ of the distribution. Alternatively, $M(u)$ is the mean residual life conditioned that survival to lifetime $x(F)$ has occurred.

If $u = 0$, then $M(0)$ is the expectation of the life distribution or in otherwords $M(0) = \lambda_1$ of the parent quantile function. If $u = 1$, then $M(u) = 0$ (death has occurred)—there is zero residual life remaining. The implementation intercepts an intermediate ∞ and returns 0 for $u = 1$.

The $M(u)$ is referred to as a quantile function but this quantity is not to be interpreted as a type of probability distribution. The second example produces a $M(u)$ that is not monotonic increasing with u and therefore it is immediately apparent that $M(u)$ is not the quantile function of some probability distribution by itself. Nair et al. (2013) provide extensive details on quantile-based reliability analysis.

Usage

```
rmlmomco(f, para)
```

Arguments

`f`	Nonexceedance probability ($0 \le F \le 1$).
`para`	The parameters from `lmom2par` or `vec2par`.

Value

Mean residual value for F.

Note

The Mean Residual Quantile Function is the first of many other functions and "curves" associated with lifetime/reliability analysis operations that at their root use the quantile function (QF, $x(F)$) of a distribution. Nair et al. (2013) (NSB) is the authoritative text on which the following functions in **lmomco** were based

Residual mean QF	$M(u)$	`rmlmomco`	NSB[p.51]
Variance residual QF	$V(u)$	`rmvarlmomco`	NSB[p.54]
α-percentile residual QF	$P_\alpha(u)$	`ralmomco`	NSB[p.56]
Reversed α-percentile residual QF	$R_\alpha(u)$	`rralmomco`	NSB[p.69–70]
Reversed residual mean QF	$R(u)$	`rrmlmomco`	NSB[p.57]
Reversed variance residual QF	$D(u)$	`rrmvarlmomco`	NSB[p.58]
Conditional mean QF	$\mu(u)$	`cmlmomco`	NSB[p.68]
Vitality function (see conditional mean)			
Total time on test transform QF	$T(u)$	`tttlmomco`	NSB[p.171–172, 176]
Scaled total time on test transform QF	$\phi(u)$	`sttlmomco`	NSB[p.173]
Lorenz curve	$L(u)$	`lrzlmomco`	NSB[p.174]
Bonferroni curve	$B(u)$	`bfrlmomco`	NSB[p.179]
Leimkuhler curve	$K(u)$	`lkhlmomco`	NSB[p.181]
Income gap ratio curve	$G(u)$	`riglmomco`	NSB[p.230]
Mean life: $\mu \equiv \mu(0) \equiv \lambda_1(u=0) \equiv \lambda_1$			
L-moments of residual life	$\lambda_r(u)$	`reslife.lmoms`	NSB[p.202]
L-moments of reversed residual life	${}_{\mathrm{r}}\lambda_r(u)$	`rreslife.lmoms`	NSB[p.211]

Author(s)

W.H. Asquith

References

Kupka, J., and Loo, S., 1989, The hazard and vitality measures of ageing: Journal of Applied Probability, v. 26, pp. 532–542.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, cmlmomco, rmvarlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
qlmomco(0.5, A)  # The median lifetime = 1261 days
rmlmomco(0.5, A) # The average remaining life given survival to the median = 861 days

# 2nd example with discussion points
F <- nonexceeds(f01=TRUE)
plot(F, qlmomco(F, A), type="l", # usual quantile plot as seen throughout lmomco
     xlab="NONEXCEEDANCE PROBABILITY", ylab="LIFETIME, IN DAYS")
lines(F, rmlmomco(F, A), col=2, lwd=3)           # mean residual life
L1 <- lmomgov(A)$lambdas[1]                      # mean lifetime at start/birth
lines(c(0,1), c(L1,L1), lty=2)                   # line "ML" (mean life)
# Notice how ML intersects M(F|F=0) and again later in "time" (about F = 1/4)  showing
# that this Govindarajulu has a peak mean residual life that is **greater** than the
# expected lifetime at start. The M(F) then tapers off to zero at infinity time (F=1).
# M(F) is non-monotonic for this example---not a proper probability distribution.
```

---

rmvarlmomco *Variance Residual Quantile Function of the Distributions*

Description

This function computes the Variance Residual Quantile Function for a quantile function $x(F)$ (par2qua, qlmomco). The variance is defined by Nair et al. (2013, p. 55) as

$$V(u) = \frac{1}{1-u}\int_u^1 M(u)^2 \,\mathrm{d}p,$$

where $V(u)$ is the variance of $M(u)$ (the residual mean quantile function, rmlmomco) for nonexceedance probability u.

Usage

```
rmvarlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.

Value

Residual variance value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rmlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
qlmomco(0.5, A)  # The median lifetime = 1261 days
rmlmomco(0.5, A) # The average remaining life given survival to the median = 861 days
rmvarlmomco(0.5, A) # and the variance of that value.
## Not run:
A <- lmom2par(vec2lmom(c(2000, 450, 0.14, 0.1)), type="kap")
F <- nonexceeds(f01=TRUE)
plot(F, qlmomco(F,A), type="l", ylim=c(100,6000),
     xlab="NONEXCEEDANCE PROBABILITY", ylab="LIFETIME OR SQRT(VAR LIFE), IN DAYS")
lines(F, sqrt( rmvarlmomco(F, A)), col=4, lwd=4) # thick blue, residual mean life
lines(F, sqrt(rrmvarlmomco(F, A)), col=2, lwd=4) # thick red, reversed resid. mean life
lines(F,   rmlmomco(F,A), col=4, lty=2); lines(F, rrmlmomco(F,A), col=2, lty=2)
lines(F,  tttlmomco(F,A), col=3, lty=2); lines(F,  cmlmomco(F,A), col=3)

## End(Not run)
```

---

rralmomco *Reversed Alpha-Percentile Residual Quantile Function of the Distributions*

Description

This function computes the Reversed α-Percentile Residual Quantile Function for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair and Vineshkumar (2011, p. 87) and Midhu et al. (2013, p. 13) as

$$R_\alpha(u) = x(u) - x(u[1-\alpha]),$$

where $R_\alpha(u)$ is the reversed α-percentile residual quantile for nonexceedance probability u and percentile α and $x(u[1-\alpha])$ is a constant for $x(F = u[1-\alpha])$. The nonreversed α-percentile residual quantile is available under ralmomco.

Usage

```
rralmomco(f, para, alpha=0)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.
alpha	The α percentile, which is divided by 100 inside the function ahead of calling the quantile function of the distribution.

Value

Reversed α-percentile residual quantile value for F.

Note

Technically it seems that Nair et al. (2013) do not explictly define the reversed α-percentile residual quantile but their index points to pp. 69–70 for a derivation involving the Generalized Lambda distribution (GLD) but that derivation (top of p. 70) has incorrect algebra. A possibilty is that Nair et al. (2013) forgot to include $R_\alpha(u)$ as an explicit definition in juxtaposition to $P_\alpha(u)$ (ralmomco) and then apparently made an easy-to-see algebra error in trying to collect terms for the GLD.

Author(s)

W.H. Asquith

References

Nair, N.U., and Vineshkumar, B., 2011, Reversed percentile residual life and related concepts: Journal of the Korean Statistical Society, v. 40, no. 1, pp. 85–92.

Midhu, N.N., Sankaran, P.G., and Nair, N.U., 2013, A class of distributions with linear mean residual quantile function and it's generalizations: Statistical Methodology, v. 15, pp. 1–24.

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, ralmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(145, 2649, 2.11), type="gov") # so set lower bounds = 0.0
rralmomco(0.78, A, alpha=50)
## Not run:
F <- nonexceeds(f01=TRUE); r <- range(rralmomco(F,A, alpha=50), ralmomco(F,A, alpha=50))
plot(F, rralmomco(F,A, alpha=50), type="l", xlab="NONEXCEEDANCE PROBABILITY",
                  ylim=r, ylab="MEDIAN RESIDUAL OR REVERSED LIFETIME, IN DAYS")
lines(F, ralmomco(F, A, alpha=50), col=2) # notice the lack of symmetry

## End(Not run)
```

`rreslife.lmoms` *L-moments of Reversed Residual Life*

Description

This function computes the L-moments of reversed residual life for a quantile function $x(F)$ for an exceedance threshold in probabiliy of u. The L-moments of residual life are thoroughly described by Nair et al. (2013, p. 211). These L-moments are define as

$$ {}_{\mathrm{r}}\lambda(u)_r = \sum_{k=0}^{r-1} (-1)^k \binom{r-1}{k}^2 \int_0^u \left(\frac{p}{u}\right)^{r-k-1} \left(1-\frac{p}{u}\right)^k \frac{x(p)}{u}\,\mathrm{d}p, $$

where ${}_{\mathrm{r}}\lambda(u)_r$ is the rth L-moment at residual life probability u. The L-moment ratios ${}_{\mathrm{r}}\tau(u)_r$ have the usual definitions. The implementation here exclusively uses the quantile function of the distribution. If $u = 0$, then the usual L-moments of the quantile function are returned because the integration domain is the entire potential lifetime range. If $u = 0$, then ${}_{\mathrm{r}}\lambda(1)_1 = x(0)$ is returned, which is the minimum lifetime of the distribution (the value for the lower support of the distribution), and the remaining ${}_{\mathrm{r}}\lambda(1)_r$ for $r \geq 2$ are set to `NA`. The reversal aspect is denoted by the prepended romanscript r to the λ's and τ's. Lastly, the notation (u) is neither super or subscripted to avoid confusion with L-moment order r or the TL-moments that indicate trimming level as a superscript (see `TLmoms`).

Usage

```
rreslife.lmoms(f, para, nmom=5)
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$).
`para`	The parameters from `lmom2par` or `vec2par`.
`nmom`	The number of moments to compute. Default is 5.

Value

An R `list` is returned.

`lambdas`	Vector of the L-moments. First element is ${}_{\mathrm{r}}\lambda_1$, second element is ${}_{\mathrm{r}}\lambda_2$, and so on.
`ratios`	Vector of the L-moment ratios. Second element is ${}_{\mathrm{r}}\tau$, third element is ${}_{\mathrm{r}}\tau_3$ and so on.
`life.notexceeds`	The value for $x(F)$ for $F =$ `f`.
`life.percentile`	The value $100\times$`f`.

trim	Level of symmetrical trimming used in the computation, which is NULL because no trimming theory for L-moments of residual life have been developed or researched.
leftrim	Level of left-tail trimming used in the computation, which is NULL because no trimming theory for L-moments of residual life have been developed or researched.
rightrim	Level of right-tail trimming used in the computation, which is NULL because no trimming theory for L-moments of residual life have been developed or researched.
source	An attribute identifying the computational source of the L-moments: "rreslife.lmoms".

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

rmlmomco, reslife.lmoms

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
"afunc" <- function(p)        { return(par2qua(p,A,paracheck=FALSE)) }
"bfunc" <- function(p,u=NULL) { return((2*p - u)*par2qua(p,A,paracheck=FALSE)) }
f <- 0.35
rL1a <- integrate(afunc, lower=0, upper=f)$value     / f   # Nair et al. (2013, eq. 6.18)
rL2a <- integrate(bfunc, lower=0, upper=f, u=f)$value / f^2 # Nair et al. (2013, eq. 6.19)
rL <- rreslife.lmoms(f, A, nmom=2) # The data.frame shows equality of the two approaches.
rL1b <- rL$lambdas[1]; rL2b <- rL$lambdas[2]
print(data.frame(rL1a=rL1a, rL1b=rL1b, rL2b=rL2b, rL2b=rL2b))
## Not run:
# 2nd Example, let us look at Tau3, each of the L-skews are the same.
T3    <- par2lmom(A)$ratios[3]
T3.0  <-  reslife.lmoms(0, A)$ratios[3]
rT3.1 <- rreslife.lmoms(1, A)$ratios[3]

## End(Not run)
## Not run:
# Nair et al. (2013, p. 212), test shows rL2(u=0.77) = 12.6034
A <- vec2par(c(230, 269, 3.3), type="gpa"); F <- 0.77
"afunc" <- function(p) { return(p*rrmlmomco(p,A)) }
rL2u1 <- (F)^(-2)*integrate(afunc,0,F)$value
rL2u2 <- rreslife.lmoms(F,A)$lambdas[2]
```

```
## End(Not run)
```

---

rrmlmomco *Reversed Mean Residual Quantile Function of the Distributions*

Description

This function computes the Reversed Mean Residual Quantile Function for quantile function $x(F)$ (par2qua, qlmomco). The function is defined by Nair et al. (2013, p.57) as

$$R(u) = x(u) - \frac{1}{u}\int_0^u x(p)\;\mathrm{d}p,$$

where $R(u)$ is the reversed mean residual for nonexceedance probability u and $x(u)$ is a constant for $x(F = u)$.

Usage

```
rrmlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.

Value

Reversed mean residual value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rrmvarlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.6), type="gov") # so set lower bounds = 0.0
qlmomco(0.5, A)  # The median lifetime = 1005 days
rrmlmomco(0.5, A) # The reversed mean remaining life given median survival = 691 days

## Not run:
F <- nonexceeds(f01=TRUE)
plot(F, qlmomco(F,A), type="l", # life
     xlab="NONEXCEEDANCE PROBABILITY", ylab="LIFETIME, IN DAYS")
lines(F,  rmlmomco(F, A), col=4, lwd=4) # thick blue, mean residual life
lines(F, rrmlmomco(F, A), col=2, lwd=4) # thick red, reversed mean residual life

## End(Not run)
```

---

rrmvarlmomco *Reversed Variance Residual Quantile Function of the Distributions*

Description

This function computes the Reversed Variance Residual Quantile Function for a quantile function xF (par2qua, qlmomco). The variance is defined by Nair et al. (2013, p. 58) as

$$D(u) = \frac{1}{u}\int_0^u R(u)^2\;\mathrm{d}p,$$

where $D(u)$ is the variance of $R(u)$ (the reversed mean residual quantile function, rrmlmomco) for nonexceedance probability u. The variance of $M(u)$ is provided in rmvarlmomco.

Usage

```
rrmvarlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.

Value

Reversed residual variance value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rrmlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 264, 1.6), type="gov") # so set lower bounds = 0.0
rrmvarlmomco(0.5, A) # variance at the median reversed mean residual life
## Not run:
A <- vec2par(c(-100, 264, 1.6), type="gov")
F <- nonexceeds(f01=TRUE)
plot(F, rmvarlmomco(F,A), type="l")
lines(F, rrmvarlmomco(F,A), col=2)

## End(Not run)
```

---

sen.mean *Sen Weighted Mean Statistic*

Description

The Sen weighted mean statistic $\mathcal{S}_{n,k}$ is a robust estimator of the mean of a distribution

$$\mathcal{S}_{n,k} = \binom{n}{2k+1}^{-1} \sum_{i=1}^{n} \binom{i-1}{k} \binom{n-i}{k} x_{i:n},$$

where $x_{i:n}$ are the sample order statistics and k is a weighting or trimming parameter. If $k = 2$, then the $\mathcal{S}_{n,2}$ is the first symmetrical TL-moment (trim = 1). Note that $\mathcal{S}_{n,0} = \mu = \overline{X}_n$ or the arithmetic mean and $\mathcal{S}_{n,k}$ is the sample median if either n is even and $k = (n/2) - 1$ or n is odd and $k = (n-1)/2$.

Usage

```
sen.mean(x, k=0)
```

Arguments

x	A vector of data values that will be reduced to non-missing values.
k	A weighting or trimming parameter $0 < k < (n-1)/2$.

Value

An R list is returned.

sen	The sen mean $\mathcal{S}_{n,k}$.
source	An attribute identifying the computational source: “sen.mean”.

Author(s)

W.H. Asquith

References

Jurečková, J., and Picek, J., 2006, Robust statistical methods with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 1–58488–454–1, 197 p.

Sen, P.K., 1964, On some properties of the rank-weighted means: Journal Indian Society of Agricultural Statistics: v. 16, pp. 51–61.

See Also

TLmoms, gini.mean.diff

Examples

```
fake.dat <- c(123, 34, 4, 654, 37, 78)
sen.mean(fake.dat); mean(fake.dat) # These should be the same values

sen.mean(fake.dat, k=(length(fake.dat)/2) - 1); median(fake.dat)
# Again, same values

# Finally, the sen.mean() is like a symmetrically trimmed TL-moment
# Let us demonstrate by computed a two sample trimming for each side
# for a Normal distribution having a mean of 100.
fake.dat <- rnorm(20, mean=100)
lmr <- TLmoms(fake.dat, trim=2)
sen <- sen.mean(fake.dat, k=2)

print(abs(lmr$lambdas[1] - sen$sen)) # zero is returned
```

---

sentiv.curve *Compute the Sensitivity Curve for a Single Quantile*

Description

The *sensitivity curve* (SC) is a means to assess how sensitive a particular statistic T_{n+1} for a sample of size n is to an additional sample x to be included. For the implementation by this function, the statistic T is a specific quantile $x(F)$ of interest set by a nonexceedance probability F. The SC is

$$SC_{n+1}(x, |F) = (n+1)(T_{n+1} - T_n),$$

where T_n represent the statistic for the sample of size n. The notation here follows that of Hampel (1974, p. 384) concerning n and $n + 1$.

Usage

```
sentiv.curve(f, x, method=c("bootstrap", "polynomial", "none"),
                    data=NULL, para=NULL, ...)
```

Arguments

f	The nonexceedance probability F of the quantile for which the sensitivity of its estimation is needed. Only the first value if a vector is given is used and a warning issued.
x	The x values representing the potential *one more value* to be added to the original data.
data	A vector of mandatory sample data values. These will either be converted to (1) order statistic expectations exact analytical expressions or simulation (backup plan), (2) Bernstein (or similar) polynomials, or (3) the provided values treated as if they are the order statistic expectations.
method	A character variable determining how the statistics T are computed (see Details).
para	A distribution parameter list from a function such as vec2par or lmom2par.
...	Additional arguments to pass either to the lmoms.bootbarvar or to the dat2bernqua function.

Details

The main features of this function involve how the statistics are computed and are controlled by the method argument. Three different approaches are provided.

Bootstrap: Arguments data and para are *mandatory*. If boostrap is requested, then the distribution type set by the type attribute in para is used along with the method of L-moments for $T(F)$ estimation. The $T_n(F)$ is directly computed from the distribution in para. And for each x, the $T_{n+1}(F)$ is computed by lmoms, lmom2par, and the distribution type. The sample so fed to lmoms is denoted as c(EX, x).

Polynomial: Argument data is *mandatory* and para is *not* used. If polynomial is requested, then the Bernstein polynomial (likely) from the dat2bernqua is used. The $T_n(F)$ is computed by the data sample. And for each x, the $T_{n+1}(F)$ also is computed by dat2bernqua, but the sample so fed to dat2bernqua is denoted as c(EX, x).

None: Arguments data and para are *mandatory*. If none is requested, then the distribution type set by the type attribute in para is used along with the method of L-moments. The $T_n(F)$ is directly computed from the distribution in para. And for each x, the $T_{n+1}(F)$ is computed by lmoms, lmom2par, and the distribution type. The sample so fed to lmoms is denoted as c(EX, x).

The internal variable EX now requires discussion. If method=none, then the data are sorted and set into the internal variable EX. Conversely, if method=bootstrap or method=polynomial, then EX will contain the expectations of the order statistics from lmoms.bootbarvar.

Lastly, the Weibull plotting positions are used for the probability values for the data as provided by the pp function. Evidently, if method is either parent or polynomial then a "stylized sensitivity curve" would created (David, 1981, p. 165) because the expectations of the sample order statistics and not the sample order statistics (the sorted sample) are used.

Value

An R list is returned.

curve	The value for $SC(x) = (n + 1)(T_{n+1} - T_n)$.

curve.perchg	The percent change sensitivity curve by $SC^{(\%)}(x) = 100 \times (T_{n+1} - T_n)/T_n$.
Tnp1	The values for $T_{n+1} = T_n + SC(x)/(n+1)$.
Tn	The value (singular) for T_n which was estimated according to method.
color	The curve potentially passes through a zero depending on the values for x. The color is set to distinquish between negatives and positives so that the user could use the absolute value of curve on logarithmic scales and use the color to distinquish the original negatives.
EX	The values for the internal variable EX.
source	An attribute identifying the computational source of the sensitivity curve: “sentiv.curve”.

Author(s)

W.H. Asquith

References

David, H.A., 1981, Order statistics: John Wiley, New York.

Hampel, F.R., 1974, The influence curve and its role in robust estimation: Journal of the American Statistical Association, v. 69, no. 346, pp. 383–393.

See Also

expect.max.ostat

Examples

```
## Not run:
set.seed(50)
mean <- 12530; lscale <- 5033; lskew <- 0.4
n <- 46; type <- "gev"; lmr <- vec2lmom(c(mean,lscale,lskew))
F <- 0.90 # going to explore sensitivity on the 90th percentile
par.p <- lmom2par(lmr, type=type) # Parent distribution
TRUE.Q <- par2qua(F, par.p)
X <- sort(rlmomco(n, par.p)) # Simulate a small sample
par.s <- lmom2par(lmoms(X), type=type) # Now fit the distribution
SIM.Q <- par2qua(F, par.s); SIM.BAR <- par2lmom(par.s)$lambdas[1]
D <- log10(mean) - log10(lscale)
R <- as.integer(log10(mean)) + c(-D, D) # need some x-values to explore
Xs <- 10^(seq(R[1], R[2], by=.01)) # x-values to explore
# Sample estimate are the "parent" only to mimic a more real-world setting.
# where one "knows" the form of the parent but perhaps not the parameters.
SC1 <- sentiv.curve(F, Xs, data=X, para=par.s, method="bootstrap")
SC2 <- sentiv.curve(F, Xs, data=X, para=par.s, method="polynomial",
                              bound.type="Carv")
SC3 <- sentiv.curve(F, Xs, data=X, para=par.s, method="none")
xlim <- range(c(Xs,SC1$Tnp1,SC2$Tnp1,SC3$Tnp1))
ylim <- range(c(SC1$curve.perchg, SC2$curve.perchg, SC3$curve.perchg))
plot(xlim, c(0,0), type="l", lty=2, ylim=ylim, xaxs="i", yaxs="i",
```

```
     xlab=paste("Magnitude of next value added to sample of size",n),
     ylab=paste("Percent change fitted",F,"probability quantile"))
mtext(paste("Distribution",par.s$type,"with parameters",
      paste(round(par.s$para, digits=3), collapse=", ")))
lines(rep(TRUE.Q,  2), c(-10,10), lty=4, lwd=3)
lines(rep(SIM.BAR, 2), c(-10,10), lty=3, lwd=2)
lines(rep(SIM.Q,   2), c(-10,10), lty=2)
lines(Xs, SC1$curve.perchg, lwd=3, col=1)
lines(Xs, SC2$curve.perchg, lwd=2, col=2)
lines(Xs, SC3$curve.perchg, lwd=1, col=4)
rug(SC1$Tnp1, col=rgb(0,0,0,0.3))
rug(SC2$Tnp1, col=rgb(1,0,0,0.3))
rug(SC3$Tnp1, col=rgb(0,0,1,0.3), tcl=-.75) #
## End(Not run)
```

---

slmomco — *Reversed Cumulative Distribution Function (Survival Function) of the Distributions*

Description

This function acts as an alternative front end to par2cdf but reverses the probability to form the survival function. Conceptually, $S(F) = 1 - F(x)$ where $F(x)$ is plmomco (implemented by par2cdf). The nomenclature of the slmomco function is to mimic that of built-in R functions that interface with distributions.

Usage

```
slmomco(x, para)
```

Arguments

x	A real value.
para	The parameters from lmom2par or similar.

Value

Exceedance probability ($0 \leq S \leq 1$) for x.

Author(s)

W.H. Asquith

See Also

dlmomco, plmomco, qlmomco, rlmomco, add.lmomco.axis

Examples

```
para <- vec2par(c(0,1),type='nor') # Standard Normal parameters
exceed <- slmomco(1, para) # percentile of one standard deviation
```

---

stttlmomco *Scaled Total Time on Test Transform of Distributions*

Description

This function computes the Scaled Total Time on Test Transform Quantile Function for a quantile function $x(F)$ (par2qua, qlmomco). The TTT is defined by Nair et al. (2013, p. 173) as

$$\phi(u) = \frac{1}{\mu}\left[(1-u)x(u) + \int_0^u x(p)\;\mathrm{d}p\right],$$

where $\phi(u)$ is the scaled total time on test for nonexceedance probability u, and $x(u)$ is a constant for $x(F = u)$. The $\phi(u)$ is also expressible in terms of total time on test transform quantile function ($T(u)$, tttlmomco) as

$$\phi(u) = \frac{T(u)}{\mu},$$

where μ is the conditional mean (cmlmomco) at $u = 0$ and the later definition is the basis for implementation in **lmomco**. The integral in the first definition is closely related to the structure of the reversed residual mean quantile function ($R(u)$, rrmlmomco).

Usage

```
stttlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \le F \le 1$).
para	The parameters from lmom2par or vec2par.

Value

Scaled total time on test value for F.

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, tttlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin,
# but for this example, let us set the lower limit at 100 days.
A <- vec2par(c(100, 2649, 2.11), type="gov")
f <- 0.47  # Both computations of Phi show 0.6455061
"afunc" <- function(p) { return(par2qua(p,A,paracheck=FALSE)) }
tmpa <- 1/cmlmomco(f=0, A); tmpb <- (1-f)*par2qua(f,A,paracheck=FALSE)
Phiu1 <- tmpa * ( tmpb + integrate(afunc,0,f)$value )
Phiu2 <- stttlmomco(f, A)
## Not run:
# The TTT-plot (see Nair et al. (2013, p. 173))
n <- 30; X <- sort(rlmomco(n, A)); lmr <- lmoms(X)  # simulated lives and their L-moments
# recognize here that the "fit" is to the lifetime data themselves and not to special
# curves or projections of the data to other scales
"Phir" <- function(r, X, sort=TRUE) {
   n <- length(X); if(sort) X <- sort(X)
   if(r == 0) return(0) # can use 2:r as X_{0:n} is zero
   Tau.rOFn <- sapply(1:r, function(j) { Xlo <- ifelse((j-1) == 0, 0, X[(j-1)]);
                                        return((n-j+1)*(X[j] - Xlo)) })
   return(sum(Tau.rOFn))
}
Xbar <- mean(X); rOFn <- (1:n)/n # Nair et al. (2013) are clear r/n used in the Phi(u)
Phi <- sapply(1:n, function(r) { return(Phir(r,X, sort=FALSE)) }) / (n*Xbar)
layout(matrix(1:3, ncol=1))
plot(rOFn, Phi, type="b",
     xlab="NONEXCEEDANCE PROBABILITY", ylab="SCALED TOTAL TIME ON TEST")
lines(rOFn, stttlmomco(rOFn, A), lwd=2, col=8) # solid grey, the parent distribution
par1 <- pargov(lmr); par2 <- pargov(lmr, xi=min(X)) # notice attempt to "fit at minimum"
lines(pp(X), stttlmomco(rOFn, par1)) # now Weibull (i/(n+1)) being used for F via pp()
lines(pp(X), stttlmomco(rOFn, par2), lty=2) # perhaps better, but could miss short lives
F <- nonexceeds(f01=TRUE)
plot(pp(X), sort(X), xlab="NONEXCEEDANCE PROBABILITY", ylab="TOTAL TIME ON TEST (DAYS)")
lines(F, qlmomco(F, A), lwd=2, col=8) # the parent again
lines(F, qlmomco(F, par1), lty=1); lines(F, qlmomco(F, par2), lty=2) # two estimated fits
plot(F,  lrzlmomco(F, par2), col=2, type="l")  # Lorenz curve from L-moment fit (red)
lines(F, bfrlmomco(F, par2), col=3, lty=2) # Bonferroni curve from L-moment fit (green)
lines(F, lkhlmomco(F, par2), col=4, lty=4) # Leimkuhler curve from L-moment fit (blue)
lines(rOFn, Phi) # Scaled Total Time on Test

## End(Not run)
```

---

supdist *The Support of a Distribution based on the Parameters*

Description

This function takes a parameter object, such as that returned by `lmom2par`, and computes the support (the lower and upper bounds, $\{L, U\}$) of the distribution given by the parameters. The computation is based on two calls to `par2qua` for the parameters in argument `para` (Θ) and nonexceedance probabilities $F \in \{0, 1\}$:

```
lower <- par2qua(0, para)
upper <- par2qua(1, para)
```

The quality of $\{L, U\}$ is dependent of the handling of $F \in \{0, 1\}$ internal to each quantile function. Across the suite of distributions supported by **lmomco**, potential applications, and parameter combinations, it difficult to ensure numerical results for the respective $\{L, U\}$ are either very small, are large, or are (or should be) infinite. The distinction is sometimes difficult depending how fast the tail(s) of a distribution is (are) either approaching a limit as F respectively approaches 0^+ or 1^-.

The intent of this function is to provide a unified portal for $\{L, U\}$ estimation. Most of the time R (and **lmomco**) do the right thing anyway and the further overhead within the parameter estimation suite of functions in **lmomco** is not implemented.

The support returned by this function might be useful in extended application development involving probability density functions `pdfCCC` ($f(x, \Theta)$, see `dlmomco`) and cumulative distribution functions `cdfCCC` ($F(x, \Theta)$, see `plmomco`) functions—both of these functions use as their primary argument a value x that exists along the real number line.

Usage

```
supdist(para, trapNaN=FALSE, delexp=0.5, paracheck=TRUE, ...)
```

Arguments

`para`	The parameters of the distribution.
`trapNaN`	A logical influencing how `NaN` are handled (see Note).
`delexp`	The magnitude of the decrementing of the exponent to search down and up from. A very long-tailed but highly peaked distribution might require this to be smaller than default.
`paracheck`	A logical controlling whether the parameters are checked for validity.
`...`	Additional arguments to pass.

Value

An R `list` is returned.

`type`	Three character (minimum) distribution type (for example, `type="gev"`);
`support`	The support (or range) of the fitted distribution;
`nonexceeds`	The nonexceedance probabilities at the computed support.
`fexpons`	A vector indicating how the respective lower and upper boundaries were arrived at (see Note); and

finite	A logical on each entry of the support with a preemptive call by the is.finite function in R.
source	An attribute identifying the computational source of the distribution support: "supdist".

Note

Concerning fexpons, for the returned vectors of length 2, index 1 is for $\{L\}$ and index 2 is for $\{U\}$. If an entry in fexpons is NA, then $F = 0$ or $F = 1$ for the respective bound was possible. And even if trapNaN is TRUE, no further refinement on the bounds was attempted.

On the otherhand, if trapNaN is TRUE and if the bounds $\{L\}$ and (or) $\{U\}$ is not NA, then an attempt was made to move away from $F \in \{0, 1\}$ in incremental integer exponent from 0^+ or 1^- until a NaN was not encountered. The integer exponents are $i \in [-(\phi), -(\phi - 1), \ldots, -4]$, where ϕ = .Machine$sizeof.longdouble and -4 is a hardwired limit (1 part in 10,000). In the last example in the Examples section, the $\{U\}$ for $F = 1$ quantile is NaN but $1 - 10^i$ for which $i = -16$, which also is the .Machine$sizeof.longdouble on the author's development platform.

At first release, it seems there was justification in triggering this to TRUE if a quantile function returns a NA when asked for $F = 0$ or $F = 1$—some quantile functions partially trapped NaNs themselves. So even if trapNaN == FALSE, it is triggered to TRUE if a NA is discovered as described. *Users are encouraged to discuss adaptions or changes to the implementation of* supdist *with the author.*

Thus it should be considered a feature of supdist that should a quantile function already trap errors at either $F = 0$ or $F = 1$ and return NA, then trapNaN is internally set to TRUE regardless of being originally FALSE and the preliminary limit is reset to NaN. The Rice distribution quarice is one such example that internally already traps an $F = 1$ by returning $x(F{=}1)$ =NA.

Author(s)

W.H. Asquith

See Also

lmom2par

Examples

```
lmr <- lmoms(c(33, 37, 41, 54, 78, 91, 100, 120, 124))
supdist(lmom2par(lmr, type="gov" )) # Lower = 27.41782, Upper = 133.01470
supdist(lmom2par(lmr, type="gev" )) # Lower = -Inf,     Upper = 264.4127

supdist(lmom2par(lmr, type="wak" ))               # Lower = 16.43722, Upper = NaN
supdist(lmom2par(lmr, type="wak" ), trapNaN=TRUE) # Lower = 16.43722, Upper = 152.75126
#$support  16.43722  152.75126
#$fexpons        NA  -16
#$finite       TRUE  TRUE
## Not run:
para <- vec2par(c(0.69, 0.625), type="kmu") # very flat tails and narrow peak!
supdist(para, delexp=1   )$support # [1] 0        NaN
supdist(para, delexp=0.5 )$support # [1] 0.000000 3.030334
supdist(para, delexp=0.05)$support # [1] 0.000000 3.155655
```

```
# This distribution appears to have a limit at PI and the delexp=0.5

## End(Not run)
```

T2prob *Convert a Vector of T-year Return Periods to Annual Nonexceedance Probabilities*

Description

This function converts a vector of T-year return periods to annual nonexceedance probabilities F

$$F = 1 - \frac{1}{T},$$

where $0 \leq F \leq 1$.

Usage

```
T2prob(T)
```

Arguments

T	A vector of T-year return periods.

Value

A vector of annual nonexceedance probabilities.

Author(s)

W.H. Asquith

See Also

prob2T, nonexceeds, add.lmomco.axis

Examples

```
T <- c(1, 2, 5, 10, 25, 50, 100, 250, 500)
F <- T2prob(T)
```

`tau34sq.normtest` *The Tau34-squared Test: A Normality Test based on L-skew and L-kurtosis and an Elliptical Rejection Region on an L-moment Ratio Diagram*

Description

This function performs highly intriguing test for normality using L-skew (τ_3) and L-kurtosis (τ_4) computed from an input vector of data. The test is simultaneously focused on L-skew and L-kurtosis. Harri and Coble (2011) presented two types of normality tests based on these two L-moment ratios. Their first test is dubbed the $\tau_3\tau_4$ test. Those authors however conclude that a second test dubbed the $\tau^2_{3,4}$ test "in particular shows consistently high power against [sic] symmetric distributions and also against [sic] skewed distributions and is a powerful test that can be applied against a variety of distributions."

A sample-size transformed quantity of the sample L-skew ($\hat\tau_3$) is

$$Z(\tau_3) = \hat\tau_3 \times \frac{1}{\sqrt{0.1866/n + 0.8/n^2}},$$

which has an approximate Standard Normal distribution. A sample-sized transformation of the sample L-kurtosis ($\hat\tau_4$) is

$$Z(\tau_4)' = \hat\tau_4 \times \frac{1}{\sqrt{0.0883/n}},$$

which also has an approximate Standard Normal distribution. A superior approximation for the variate of the Standard Normal distribution however is

$$Z(\tau_4) = \hat\tau_4 \times \frac{1}{\sqrt{0.0883/n + 0.68/n^2 + 4.9/n^3}},$$

and is highly preferred for the algorithms in `tau34sq.normtest`.

The $\tau_3\tau_4$ test (not implemented in `tau34sq.normtest`) by Harri and Coble (2011) can be constructed from the $Z(\tau_3)$ and $Z(\tau_4)$ statistics as shown, and a square rejection region constructed on an L-moment ratio diagram of L-skew versus L-kurtosis. However, the preferred method is the "Tau34-squared" test $\tau^2_{3,4}$ that can be developed by expressing an ellipse on the L-moment ratio diagram of L-skew versus L-kurtosis. The $\tau^2_{3,4}$ test statistic is defined as

$$\tau^2_{3,4} = Z(\tau_3)^2 + Z(\tau_4)^2,$$

which is approximately distributed as a χ^2 distribution with two degrees of freedom. The $\tau^2_{3,4}$ also is the expression of the ellipical region on the L-moment ratio diagram of L-skew versus L-kurtosis.

Usage

```
tau34sq.normtest(x, alpha=0.05, pvalue.only=FALSE, getlist=TRUE,
                    useHoskingZt4=TRUE, verbose=FALSE, digits=4)
```

Arguments

`x`	A vector of values.
`alpha`	The α significance level.
`pvalue.only`	Only return the p-value of the test and superceeds the `getlist` argument.
`getlist`	Return a list of salient parts of the computations.
`useHoskingZt4`	J.R.M. Hosking provided a better approximation $Z(\tau_4)$ in personal correspondance to Harri and Coble (2011) than the one $Z(\tau_4)'$ they first presented in their paper. This argument is a logical on whether this approximation should be used. It is highly recommended that `useHoskingZt4` be left at the default setting.
`verbose`	Print a nice summary of the test.
`digits`	How many digits to report in the summary.

Value

An R `list` is returned if `getlist` argument is true. The list contents are

`SampleTau3`	The sample L-skew.
`SampleTau4`	The sample L-kurtosis.
`Ztau3`	The Z-value of τ_3.
`Ztau4`	The Z-value of τ_4.
`Tau34sq`	The $\tau^2_{3,4}$ value.
`ChiSq.2df`	The Chi-squared distribution nonexceedance probability.
`pvalue`	The p-value of the test (original notation for package).
`p.value`	The p-value of the test (updated to align with many other hypothesis test styles).
`isSig`	A logical on whether the p-value is “statistically significant” based on the α value.
`source`	The source of the parameters: “tau34sq.normtest”.

Author(s)

W.H. Asquith

References

Harri, A., and Coble, K.H., 2011, Normality testing—Two new tests using L-moments: Journal of Applied Statistics, v. 38, no. 7, pp. 1369–1379.

See Also

`pdfnor`, `plotlmrdia`

Examples

```
HarriCoble <- tau34sq.normtest(rnorm(20), verbose=TRUE)
## Not run:
# If this basic algorithm is run repeatedly with different arguments,
# then the first three rows of table 1 in Harri and Coble (2011) can
# basically be repeated. Testing by WHA indicates that even better
# empirical alphas will be computed compared to those reported in that table 1.
# R --vanilla --silent --args n 20 s 100 < t34.R
# Below is file t34.R
library(batch) # for command line argument parsing
a <- 0.05; n <- 50; s <- 5E5 # defaults
parseCommandArgs() # it will echo out those arguments on command line
sims <- sapply(1:s, function(i) {
          return(tau34sq.normtest(rnorm(n),
                 pvalue.only=TRUE)) })
p <- length(sims[sims <= a])
print("RESULTS(Alpha, SampleSize, EmpiricalAlpha)")
print(c(a, n, p/s))

## End(Not run)
```

---

theoLmoms — *The Theoretical L-moments and L-moment Ratios using Integration of the Quantile Function*

Description

Compute the theoretical L-moments for a vector. A theoretical L-moment in integral form is

$$\lambda_r = \frac{1}{r}\sum_{k=0}^{r-1}(-1)^k \binom{r-1}{k} \frac{r!\,I_r}{(r-k-1)!\,k!},$$

in which

$$I_r = \int_0^1 x(F) \times F^{r-k-1}(1-F)^k \,\mathrm{d}F,$$

where $x(F)$ is the quantile function of the random variable X for nonexceedance probability F, and r represents the order of the L-moments. This function actually dispatches to `theoTLmoms` with `trim=0` argument.

Usage

```
theoLmoms(para, nmom=5, minF=0, maxF=1, quafunc=NULL,
                nsim=50000, fold=5,
                silent=TRUE, verbose=FALSE, ...)
```

Arguments

`para`	A distribution parameter object such as from `vec2par`.
`nmom`	The number of moments to compute. Default is 5.
`minF`	The end point of nonexceedance probability in which to perform the integration. Try setting to non-zero (but very small) if the integral is divergent.
`maxF`	The end point of nonexceedance probability in which to perform the integration. Try setting to non-unity (but still very close [perhaps `1 - minF`]) if the integral is divergent.
`quafunc`	An optional and arbitrary quantile function that simply needs to except a nonexceedance probability and the parameter object in `para`. This is a feature that permits computation of the L-moments of a quantile function that does not have to be implemented in the greater overhead hassles of the **lmomco** style. This feature might be useful for estimation of quantile function mixtures or those distributions not otherwise implemented in this package.
`nsim`	Simulation size for Monte Carlo integration is such is internally deemed necessary (see `silent` argument).
`fold`	The number of fractions or number of folds of `nsim`, which in other words, means that `nsim` is divided by `folds` and a loop creating `folds` integrations of `nsim/folds` is used from which the mean and mean absolute error of the integrand are computed. This is to try to recover similar output as `integrate()`.
`silent`	The argument of `silent` for the `try()` operation wrapped on `integrate()`. If set true and the integral is probability divergent, Monte Carlo integration is triggered using `nsim` and `folds`. The user would have to set `verbose=TRUE` to then acquire the returned table in `integration_table` of the integration passes including those are or are not Monte Carlo.
`verbose`	Toggle verbose output. Because the R function `integrate` is used to perform the numerical integration, it might be useful to see selected messages regarding the numerical integration.
`...`	Additional arguments to pass.

Value

An R `list` is returned.

`lambdas`	Vector of the TL-moments. First element is λ_1, second element is λ_2, and so on.
`ratios`	Vector of the L-moment ratios. Second element is τ_2, third element is τ_3 and so on.
`trim`	Level of symmetrical trimming used in the computation, which will equal zero (the ordinary L-moments) because this function dispatches to `theoTLmoms`.
`source`	An attribute identifying the computational source of the L-moments: "theoLmoms".

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, pp. 105–124.

See Also

theoTLmoms

Examples

```
para <- vec2par(c(0,1), type='nor') # standard normal
TL00 <- theoLmoms(para) # compute ordinary L-moments
```

`theoLmoms.max.ostat` *Compute the Theoretical L-moments of a Distribution Distribution based on System of Maximum Order Statistic Expectations*

Description

This function computes the theoretical L-moments of a distribution by the following

$$\lambda_r = (-1)^{r-1}\sum_{k=1}^{r}(-1)^{r-k}k^{-1}\binom{r-1}{k-1}\binom{r+k-2}{k-1}\mathrm{E}[X_{1:k}]$$

for the minima (theoLmoms.min.ostat, theoretical L-moments from the minima of order statistics) or

$$\lambda_r = \sum_{k=1}^{r}(-1)^{r-k}k^{-1}\binom{r-1}{k-1}\binom{r+k-2}{k-1}\mathrm{E}[X_{k:k}]$$

for the maxima (theoLmoms.max.ostat, theoretical L-moments from the maxima of order statistics). The functions expect.min.ostat and expect.max.ostat compute the minima ($\mathrm{E}[X_{1:k}]$) and maxima ($\mathrm{E}[X_{k:k}]$), respectively.

If `qua != NULL`, then the first expectation equation shown under expect.max.ostat is used for the order statistic expectations and any function set in `cdf` and `pdf` is ignored.

Usage

```
theoLmoms.max.ostat(para=NULL, cdf=NULL, pdf=NULL, qua=NULL,
                    nmom=4, switch2minostat=FALSE, showterms=FALSE, ...)
```

Arguments

`para`	A distribution parameter list from a function such as `lmom2par` or vec2par.
`cdf`	CDF of the distribution for the parameters.
`pdf`	PDF of the distribution for the parameters.
`qua`	Quantile function for the parameters.

`nmom` The number of L-moments to compute.

`switch2minostat` A logical in which a switch to the expectations of minimum order statistics will be used and `expect.min.ostat` instead of `expect.max.ostat` will be used with expected small change in overall numerics. The function `theoLmoms.min.ostat` provides a direct interface for L-moment computation by minimum order statistics.

`showterms` A logical controlling just a reference message that will show the multipliers on each of the order statistic minima or maxima that comprise the terms within the summations in the above formulae (see Asquith, 2011, p. 95).

`...` Optional, but likely, arguments to pass to `expect.min.ostat` or `expect.max.ostat`. Such arguments will likely tailor the integration limits that can be specific for the distribution in question. Further these arguments might be needed for the cumulative distribution function.

Value

An R `list` is returned.

`lambdas` Vector of the L-moments: first element is λ_1, second element is λ_2, and so on.

`ratios` Vector of the L-moment ratios. Second element is τ, third element is τ_3 and so on.

`trim` Level of symmetrical trimming used in the computation, which will equal `NULL` until trimming support is made.

`leftrim` Level of left-tail trimming used in the computation, which will equal `NULL` until trimming support is made.

`rightrim` Level of right-tail trimming used in the computation, which will equal `NULL` until trimming support is made.

`source` An attribute identifying the computational source of the L-moments: “theoLmoms.max.ostat”.

Note

Perhaps one of the neater capabilities that the `theoLmoms.max.ostat` and `theoLmoms.min.ostat` functions provide is for computing L-moments that are not analytically available from other authors or have no analytical solution.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

theoLmoms, expect.min.ostat, expect.max.ostat

Examples

```
## Not run:
para <- vec2par(c(40,20), type='nor')
A1 <- theoLmoms.max.ostat(para=para, cdf=cdfnor, pdf=pdfnor, switch2minostat=FALSE)
A2 <- theoLmoms.max.ostat(para=para, cdf=cdfnor, pdf=pdfnor, switch2minostat=TRUE)
B1 <- theoLmoms.max.ostat(para=para, qua=quanor, switch2minostat=FALSE)
B2 <- theoLmoms.max.ostat(para=para, qua=quanor, switch2minostat=TRUE)
print(A1$ratios[4]) # reports 0.1226017
print(A2$ratios[4]) # reports 0.1226017
print(B1$ratios[4]) # reports 0.1226012
print(B2$ratios[4]) # reports 0.1226012
# Theoretical value = 0.122601719540891.
# Confirm operational with native R-code being used inside lmomco functions
# Symmetrically correct on whether minima or maxima are used, but some
# Slight change when qnorm() used instead of dnorm() and pnorm().

para <- vec2par(c(40,20), type='exp')
A1 <- theoLmoms.max.ostat(para=para, cdf=cdfexp, pdf=pdfexp, switch2minostat=FALSE)
A2 <- theoLmoms.max.ostat(para=para, cdf=cdfexp, pdf=pdfexp, switch2minostat=TRUE)
B1 <- theoLmoms.max.ostat(para=para, qua=quaexp, switch2minostat=FALSE)
B2 <- theoLmoms.max.ostat(para=para, qua=quaexp, switch2minostat=TRUE)
print(A1$ratios[4]) # 0.1666089
print(A2$ratios[4]) # 0.1666209
print(B1$ratios[4]) # 0.1666667
print(B2$ratios[4]) # 0.1666646
# Theoretical value = 0.1666667

para <- vec2par(c(40,20), type='ray')
A1 <- theoLmoms.max.ostat(para=para, cdf=cdfray, pdf=pdfray, switch2minostat=FALSE)
A2 <- theoLmoms.max.ostat(para=para, cdf=cdfray, pdf=pdfray, switch2minostat=TRUE)
B1 <- theoLmoms.max.ostat(para=para, qua=quaray, switch2minostat=FALSE)
B2 <- theoLmoms.max.ostat(para=para, qua=quaray, switch2minostat=TRUE)
print(A1$ratios[4]) # 0.1053695
print(A2$ratios[4]) # 0.1053695
print(B1$ratios[4]) # 0.1053636
print(B2$ratios[4]) # 0.1053743
# Theoretical value = 0.1053695

## End(Not run)
## Not run:
# The Rice distribution is complex and tailoring of the integration
# limits is needed to effectively trap errors, the limits for the
# Normal distribution above are infinite so no granular control is needed.
para <- vec2par(c(30,10), type="rice")
theoLmoms.max.ostat(para=para, cdf=cdfrice, pdf=pdfrice,
                    lower=0, upper=.Machine$double.max)

## End(Not run)
```

```
## Not run:
para <- vec2par(c(0.6, 1.5), type="emu")
theoLmoms.min.ostat(para, cdf=cdfemu, pdf=pdfemu,
                    lower=0, upper=.Machine$double.max)
theoLmoms.min.ostat(para, cdf=cdfemu, pdf=pdfemu, yacoubsintegral = FALSE,
                    lower=0, upper=.Machine$double.max)

para <- vec2par(c(0.6, 1.5), type="kmu")
theoLmoms.min.ostat(para, cdf=cdfkmu, pdf=pdfkmu,
                    lower=0, upper=.Machine$double.max)
theoLmoms.min.ostat(para, cdf=cdfkmu, pdf=pdfkmu, marcumQ = FALSE,
                    lower=0, upper=.Machine$double.max)

## End(Not run)
## Not run:
# The Normal distribution is used on the fly for the Rice for high to
# noise ratios (SNR=nu/alpha > some threshold). This example will error out.
nu <- 30; alpha <- 0.5
para <- vec2par(c(nu,alpha), type="rice")
theoLmoms.max.ostat(para=para, cdf=cdfrice, pdf=pdfrice,
                    lower=0, upper=.Machine$double.max)

## End(Not run)
```

---

theopwms — *The Theoretical Probability-Weighted Moments using Integration of the Quantile Function*

Description

Compute the theoretical probability-weighted moments (PWMs) for a distribution. A theoretical PWM in integral form is

$$\beta_r = \int_0^1 x(F)\, F^r \,\mathrm{d}F,$$

where $x(F)$ is the quantile function of the random variable X for nonexceedance probability F and r represents the order of the PWM. This function loops across the above equation for each `nmom` set in the argument list. The function $x(F)$ is computed through the `par2qua` function. The distribution type is determined using the `type` attribute of the `para` argument, which is a parameter object of **lmomco** (see `vec2par`).

Usage

```
theopwms(para, nmom=5, minF=0, maxF=1, quafunc=NULL,
                nsim=50000, fold=5,
                silent=TRUE, verbose=FALSE, ...)
```

Arguments

- `para` — A distribution parameter object such as that by `lmom2par` or `vec2par`.
- `nmom` — The number of moments to compute. Default is 5.
- `minF` — The end point of nonexceedance probability in which to perform the integration. Try setting to non-zero (but small) if you have a divergent integral.
- `maxF` — The end point of nonexceedance probability in which to perform the integration. Try setting to non-unity (but close) if you have a divergent integral.
- `quafunc` — An optional and arbitrary quantile function that simply needs to except a nonexceedance probability and the parameter object in `para`. This is a feature that permits computation of the PWMs of a quantile function that does not have to be implemented in the greater overhead hassles of the **lmomco** style. This feature might be useful for estimation of quantile function mixtures or those distributions not otherwise implemented in this package.
- `nsim` — Simulation size for Monte Carlo integration is such is internally deemed necessary (see `silent` argument).
- `fold` — The number of fractions or number of folds of `nsim`, which in other words, means that `nsim` is divided by `folds` and a loop creating `folds` integrations of `nsim/folds` is used from which the mean and mean absolute error of the integrand are computed. This is to try to recover similar output as `integrate()`.
- `silent` — The argument of `silent` for the `try()` operation wrapped on `integrate()`. If set true and the integral is probability divergent, Monte Carlo integration is triggered using `nsim` and `folds`. The user would have to set `verbose=TRUE` to then acquire the returned table in `integrations` of the integration passes including those are or are not Monte Carlo.
- `verbose` — Toggle verbose output. Because the R function `integrate` is used to perform the numerical integration, it might be useful to see selected messages regarding the numerical integration.
- `...` — Additional arguments to pass.

Value

An R `list` is returned.

- `betas` — The PWMs. Note that convention is the have a β_0, but this is placed in the first index `i=1` of the `betas` vector.
- `nsim` — Echo of the `nsim` argument if and only if at least one Monte Carlo integration was required, otherwise this is set to “not needed” on the return.
- `folds` — Echo of the `folds` argument if and only if at least one Monte Carlo integration was required, otherwise this is set to “not needed” on the return.
- `monte_carlo` — A logical vector of whether one or more Monte Carlo integrations was needed for the `r`-th index of the vector during the integrations for the r-th PWM (beta).
- `source` — An attribute identifying the computational source of the probability-weighted moments: “theopwms”.
- `integrations` — If `verbose=TRUE`, then the results of the integrations are a data frame stored here. Otherwise, `integrations` is not present in the list.

Author(s)

W.H. Asquith

References

Hosking, J.R.M., 1990, L-moments–Analysis and estimation of distributions using linear combinations of order statistics: Journal of the Royal Statistical Society, Series B, v. 52, p. 105–124.

See Also

theoLmoms, pwm, pwm2lmom

Examples

```
para     <- vec2par(c(0,1),type='nor') # standard normal
the.pwms <- theopwms(para) # compute PWMs
str(the.pwms)

## Not run:
  # This example has a divergent integral triggered on the beta0. Monte Carlo (MC)
  # integration is thus triggered. The verbose=TRUE saves numerical or MC
  # integration result table to the return.
  para <- vec2par(c(2,2, 1.8673636098392308, -0.1447286792099476), type="kap")
  pwmkap <- lmom2pwm( lmomkap(para) )
  print(pwmkap$betas) # 0.1155903 1.2153105 0.9304619 0.7282926 0.5938137
  pwmthe <- theopwms(para, nmom=5, verbose=TRUE)
  print(pwmthe$betas) # 0.1235817 1.2153104 0.9304619 0.7282926 0.5938137

  para <- vec2par(c(2,2, 0.9898362024687231, -0.5140894097276032), type="kap")
  pwmkap <- lmom2pwm( lmomkap(para) )
  print(pwmkap$betas) # -0.06452787  1.33177963  1.06818379  0.85911124  0.71308145
  pwmthe <- theopwms(para, nmom=5, verbose=TRUE)
  print(pwmthe$betas) # -0.06901669  1.33177952  1.06818379  0.85911123  0.71308144
## End(Not run)
```

---

theoTLmoms *The Theoretical Trimmed L-moments and TL-moment Ratios using Integration of the Quantile Function*

Description

Compute the theoretrical trimmed L-moments (TL-moments) for a vector. The level of symmetrical or asymmetrical trimming is specified. A theoretical TL-moment in integral form is

$$\lambda^{(t_1,t_2)}_r = \underbrace{\frac{1}{r}}_{\text{average of terms}} \sum^{r-1}_{k=0} \overbrace{(-1)^k}^{\text{differences}} \underbrace{\binom{r-1}{k}}_{\text{combinations}} \frac{\overbrace{(r+t_1+t_2)!}^{\text{sample size}}\; I^{(t_1,t_2)}_r}{\underbrace{(r+t_1-k-1)!}_{\text{left tail}}\underbrace{(t_2+k)!}_{\text{right tail}}}\text{, in which}$$

$$I_r^{(t_1,t_2)} = \int_0^1 \underbrace{x(F)}_{\substack{\text{quantile}\\ \text{function}}} \times \overbrace{F^{r+t_1-k-1}}^{\text{left tail}}\overbrace{(1-F)^{t_2+k}}^{\text{right tail}}\,\mathrm{d}F,$$

where $x(F)$ is the quantile function of the random variable X for nonexceedance probability F, t_1 represents the trimming level of the t_1-smallest, t_2 represents the trimming level of the t_2-largest values, r represents the order of the L-moments. This function loops across the above equation for each nmom set in the argument list. The function $x(F)$ is computed through the par2qua function. The distribution type is determined using the type attribute of the para argument—the parameter object.

As of version 1.5.2 of **lmomco**, there exists enhanced error trapping on integration failures in theoTLmoms. The function now abandons operations should any of the integrations for the rth L-moment fail for reasons such as divergent integral or round off problems. The function returns NAs for all L-moments in lambdas and ratios.

Usage

```
theoTLmoms(para, nmom=5, trim=NULL, leftrim=NULL, rightrim=NULL,
                 minF=0, maxF=1, quafunc=NULL,
                 nsim=50000, fold=5,
                 silent=TRUE, verbose=FALSE, ...)
```

Arguments

para	A distribution parameter object of this package such as by vec2par.
nmom	The number of moments to compute. Default is 5.
trim	Level of symmetrical trimming to use in the computations. Although NULL in the argument list, the default is 0—the usual L-moment is returned.
leftrim	Level of trimming of the left-tail of the sample.
rightrim	Level of trimming of the right-tail of the sample.
minF	The end point of nonexceedance probability in which to perform the integration. Try setting to non-zero (but small) if you have a divergent integral.
maxF	The end point of nonexceedance probability in which to perform the integration. Try setting to non-unity (but close) if you have a divergent integral.
quafunc	An optional and arbitrary quantile function that simply needs to except a nonexceedance probability and the parameter object in para. This is a feature that permits computation of the L-moments of a quantile function that does not have to be implemented in the greater overhead hassles of the **lmomco** style. This feature might be useful for estimation of quantile function mixtures or those distributions not otherwise implemented in this package.
nsim	Simulation size for Monte Carlo integration is such is internally deemed necessary (see silent argument).

fold	The number of fractions or number of folds of nsim, which in other words, means that nsim is divided by folds and a loop creating folds integrations of nsim/folds is used from which the mean and mean absolute error of the integrand are computed. This is to try to recover similar output as integrate().
silent	The argument of silent for the try() operation wrapped on integrate(). If set true and the integral is probability divergent, Monte Carlo integration is triggered using nsim and folds. The user would have to set verbose=TRUE to then acquire the returned table in integrations of the integration passes including those are or are not Monte Carlo.
verbose	Toggle verbose output. Because the R function integrate is used to perform the numerical integration, it might be useful to see selected messages regarding the numerical integration.
...	Additional arguments to pass.

Value

An R list is returned.

lambdas	Vector of the TL-moments. First element is $\lambda_1^{(t_1,t_2)}$, second element is $\lambda_2^{(t_1,t_2)}$, and so on.
ratios	Vector of the L-moment ratios. Second element is $\tau^{(t_1,t_2)}$, third element is $\tau_3^{(t_1,t_2)}$ and so on.
trim	Level of symmetrical trimming used in the computation, which will equal NULL if asymmetrical trimming was used.
leftrim	Level of left-tail trimming used in the computation.
rightrim	Level of right-tail trimming used in the computation.
nsim	Echo of the nsim argument if and only if at least one Monte Carlo integration was required, otherwise this is set to “not needed” on the return.
folds	Echo of the folds argument if and only if at least one Monte Carlo integration was required, otherwise this is set to “not needed” on the return.
monte_carlo	A logical vector of whether one or more Monte Carlo integrations was needed for the r-th index of the vector during the integrations for the r-th L-moment.
source	An attribute identifying the computational source of the L-moments: “theoTL-moms” or switched to “theoLmoms” if this function was dispatched from theoLmoms.
integrations	If verbose=TRUE, then the results of the integrations are a data frame stored here. Otherwise, integrations is not present in the list.

Note

An extended example of a *unique application* of the TL-moments is useful to demonstrate capabilities of the **lmomco** package API. Consider the following example in which the analyst has 21 years of data for a given spatial location. Based on regional analysis, the highest value (the outlier = 21.12) is known to be exotically high but also documentable as not representing say a transcription error in the source database. The regional analysis also shows that the Generalized Extreme Value (GEV) distribution is appropriate.

The analyst is using a complex L-moment computational framework (say a software package called **BigStudy.R**) in which only the input data are under the control of the analyst or it is too risky to modify **BigStudy.R**. Yet, it is desired to somehow acquire robust estimation. The `outlier` value can be accommodated by estimating a pseudo-value and then simply make a substitution in the input data file for **BigStudy.R**.

The following code initiates pseudo-value estimation by storing the original 20 years of data in variable `data.org` and then extending these data with the `outlier`. The usual sample L-moments are computed in `first.lmr` and will only be used for qualitative comparison. A 3-dimensional optimizer will be used for the GEV so the starting point is stored in `first.par`.

```
data.org  <- c(5.19, 2.58, 7.59, 3.22, 7.50, 4.05, 2.54, 9.00, 3.93, 5.15,
               6.80, 2.10, 8.44, 6.11, 3.30, 5.75, 3.52, 3.48, 6.32, 4.07)
outlier   <- 21.12;            the.data  <- c(data.org, outlier)
first.lmr <- lmoms(the.data);  first.par <- pargev(first.lmr)
```

Robustness is acquired by computing the sample TL-moments such that the `outlier` is quantitatively removed by single trimming from the right side as the follow code shows:

```
trimmed.lmr <- TLmoms(the.data, rightrim=1, leftrim=0)
```

The objective now is to fit a GEV to the sample TL-moments in `trimmed.lmr`. However, the right-trimmed only ($t_1 = 0$ and $t_2 = 1$) version of the TL-moments is being used and analytical solutions to the GEV for $t = (0, 1)$ are lacking or perhaps they are too much trouble to derive. The `theoTLmoms` function provides the avenue for progress because of its numerical integration basis for acquistion of the TL-moments. An objective function for the $t_2 = 1$ TL-moments of the GEV is defined and based on the sum of square errors of the first three TL-moments:

```
"afunc" <- function(par, tarlmr=NULL, p=3) {
            the.par  <- vec2par(par, type="gev", paracheck=FALSE)
            fit.tlmr <- theoTLmoms(the.par, rightrim=1, leftrim=0)
            return(sum((tarlmr$lambdas[1:p] - fit.tlmr$lambdas[1:p])^2))
}
```

and then optimize on this function and make a qualitative comparison between the original sample L-moments (untrimmed) to the equivalent L-moments (untrimmed) of the GEV having TL-moments equaling those in `trimmed.lmr`:

```
rt <- optim(first.par$para, afunc, tarlmr=trimmed.lmr)
last.lmr <- lmomgev(vec2par(rt$par, type="gev"))

message("# Original sample    L-moment lambdas: ",
         paste(round(first.lmr$lambdas[1:3], digits=4), collapse=" "))
message("# Targeting back-fit L-moment lambdas: ",
         paste(round(last.lmr$lambdas[ 1:3], digits=4), collapse=" "))
# Original sample    L-moment lambdas: 5.7981 1.8565 0.7287
# Targeting back-fit L-moment lambdas: 5.5916 1.6501 0.5223
```

The primary result on comparison of the λ_r shows that the L-scale drops substantially as does L-skew: ($\tau_3 = 0.7287/1.8565 = 0.3925 \rightarrow \lambda_3^{(t_2=1)} = 0.5223/1.6501 = 0.3165$).

Now that the target L-moments (not TL-moments) are known (`last.lmr`), it is possible to optimize again on the value for the `outlier` that would provide the `last.lmr` within the greater computational framework in use by the analyst.

```
"bfunc" <- function(x, tarlmr=NULL, p=3) {
            sam.lmr <- lmoms(c(data.org, x))
            return(sum((tarlmr$lambdas[1:p] - sam.lmr$lambdas[1:p])^2))
}
suppressWarnings(outlier.rt <- optim(outlier, bfunc, tarlmr=last.lmr))
# silence warning about 1D optimization with optim(), well behaved here

pseudo.outlier <- round(outlier.rt$par, digits=2)
final.lmr <- lmoms(c(data.org, pseudo.outlier))

message("# Resulting new L-moment lambdas: ",
        paste(round(final.lmr$lambdas[1:3], digits=4), collapse=" "))
# Resulting new L-moment lambdas: 5.5914 1.6499 0.5221

message("# Pseudo-value for highest value: ", round(outlier.rt$par, digits=2))
# Pseudo-value for highest value: 16.78
```

Where the second optimization shows that if the largest value for the 21 years of data is given a value of 16.78 instead of its original value of 21.12 that the sample L-moments (untrimmed) will be consistent as if the TL-moments $t = (0, 1)$ has been somehow used without resorting to a risky re-coding of the greater computational framework.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

`theoLmoms`, `TLmoms`, `tlmr2par`

Examples

```
para <- vec2par(c(0, 1), type='nor') # standard normal
TL00 <- theoTLmoms(para) # compute ordinary L-moments
TL30 <- theoTLmoms(para, leftrim=3, rightrim=0) # trim 3 smallest samples

# Let us look at the difference from simulation to theoretrical using
# L-kurtosis and asymmetrical trimming for generalized Lambda dist.
n     <- 100 # really a much larger sample should be used---for speed
P     <- vec2par(c(10000, 10000, 6, 0.4),type='gld')
Lkurt <- TLmoms(quagld(runif(n),P), rightrim=3, leftrim=0)$ratios[4]
```

```
theoLkurt <- theoTLmoms(P, rightrim=3, leftrim=0)$ratios[4]
Lkurt - theoLkurt # as the number for runif goes up, this
                  # difference goes to zero

# Example using the Generalized Pareto Distribution
# to verify computations from theoretical and sample stand point.
n      <- 100 # really a much larger sample should be used---for speed
P      <- vec2par(c(12, 34, 4),type='gpa')
theoTL <- theoTLmoms(P, rightrim=2, leftrim=4)
samTL  <- TLmoms(quagpa(runif(n),P), rightrim=2, leftrim=4)
del    <- samTL$ratios[3] - theoTL$ratios[3] # if n is large difference
                                             # is small
str(del)

## Not run:
  "cusquaf" <- function(f, para, ...) { # Gumbel-Normal product
     g <- vec2par(c(para[1:2]), type="gum")
     n <- vec2par(c(para[3:4]), type="nor")
     return(par2qua(f,g)*par2qua(f,n))
  }
  para <- c(5.6, .45, 3, .3)
  theoTLmoms(para, quafunc=cusquaf) # L-skew = 0.13038711
## End(Not run)

## Not run:
  # This example has a divergent integral triggered on the last of the inner
  # loop of the 4th L-moment call. Monte Carlo (MC) integration is thus triggered.
  # The verbose=TRUE saves numerical or MC integration result table to the return.
  para   <-  vec2par(c(2.00,  2.00, -0.20, -0.55), type="kap")
  lmrbck <- lmomkap(   para, nmom=5)
  # print(lmrbck$lambdas) 3.1189568 1.9562688 0.4700229 0.4078741 0.1974055
  lmrthe <- theoTLmoms2(para, nmom=5, verbose=TRUE)             # seed dependent
  # print(lmrthe$lambdas) 3.1189569 1.9562686 0.4700227 0.4068539 0.1974049
  parkap(lmrbck)$para # 2.00       2.00     -0.20      -0.55
  parkap(lmrthe)$para # 2.018883  1.986761  -0.202422  -0.570451 # seed dependent
## End(Not run)
```

`TLmom` *A Sample Trimmed L-moment*

Description

A sample trimmed L-moment (TL-moment) is computed for a vector. The $r \geq 1$ order of the L-moment is specified as well as the level of symmetrical trimming. A trimmed TL-moment $\hat{\lambda}^{(t_1,t_2)}_r$ is

$$\hat{\lambda}^{(t_1,t_2)}_r = \frac{1}{r}\sum_{i=t_1+1}^{n-t_2}\left[\frac{\sum_{k=0}^{r-1}(-1)^k\binom{r-1}{k}\binom{i-1}{r+t_1-1-k}\binom{n-i}{t_2+k}}{\binom{n}{r+t_1+t_2}}\right]x_{i:n},$$

where t_a represents the trimming level of the t_2-largest or t_1-smallest values, r represents the order of the L-moment, n represents the sample size, and $x_{i:n}$ represents the ith sample order statistic ($x_{1:n} \leq x_{2:n} \leq \cdots \leq x_{n:n}$).

Usage

```
TLmom(x, order, trim=NULL, leftrim=NULL, rightrim=NULL, sortdata=TRUE)
```

Arguments

`x`	A vector of data values.
`order`	L-moment order to use in the computations. Default is 1 (the mean).
`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` is in the argument list, the default is 0—the usual L-moment is returned.
`leftrim`	Level of trimming of the left-tail of the sample, which should be left to `NULL` if no or symmetrical trimming is used.
`rightrim`	Level of trimming of the right-tail of the sample, which should be left to `NULL` if no or symmetrical trimming is used.
`sortdata`	A logical switch on whether the data should be sorted. The default is TRUE.

Value

An R `list` is returned.

`lambda`	The TL-moment of order=`order`, $\hat{\lambda}^{(t_1,t_2)}_r$ where r is the moment order, t_1 is left-tail trimming, and t_2 is right-tail trimming.
`order`	L-moment order computed. Default is 1 (the mean).
`trim`	Level of symmetrical trimming used in the computation.
`leftrim`	Level of left-tail trimming used in the computation, which will equal `trim` if symmetrical trimming was used.
`rightrim`	Level of right-tail trimming used in the computation, which will equal `trim` if symmetrical trimming was used.

Note

The presence of the `sortdata` switch can be dangerous. L-moment computation requires that the data be sorted into the "order statistics". Thus the default behavior of `sortdata=TRUE` is required when the function is called on its own. In practice, this function would almost certainly not be used on its own because multiple trimmed L-moments would be needed. Multiple trimmed L-moments are best computed by `TLmoms`, which calls `TLmom` multiple times. The function `TLmoms` takes over the sort operation on the data and passes `sortdata=FALSE` to `TLmom` for efficiency. (The point of this discussion is that CPU time is not wasted sorting the data more than once.)

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

TLmoms

Examples

```
X1 <- rcauchy(30)
TL <- TLmom(X1,order=2,trim=1)
```

TLmoms *The Sample Trimmed L-moments and L-moment Ratios*

Description

Compute the sample trimmed L-moments (TL-moments) for a vector. The level of symmetrical trimming is specified. The mathematical expression for a TL-moment is seen under TLmom. The TLmoms function loops across that expression and the TLmom function for each nmom=r set in the argument list.

Usage

```
TLmoms(x, nmom, trim=NULL, leftrim=NULL, rightrim=NULL, vecit=FALSE)
```

Arguments

x	A vector of data values.
nmom	The number of moments to compute. Default is 5.
trim	Level of symmetrical trimming to use in the computations. Although NULL is in the argument list, the default is 0—the usual L-moment is returned.
leftrim	Level of trimming of the left-tail of the sample, which should be left to NULL if no or symmetrical trimming is used.
rightrim	Level of trimming of the right-tail of the sample, which should be left to NULL if no or symmetrical trimming is used.
vecit	A logical to return the first two $\lambda_i \in 1, 2$ and then the $\tau_i \in 3, \cdots$ where the length of the returned vector is controlled by the nmom argument. This argument will store the trims in the attributes of the returned vector, but caution is advised if vec2par were to be used on the vector because that function does not consult the trimming.

Value

An R `list` is returned.

`lambdas`	Vector of the TL-moments. First element is $\hat{\lambda}_1^{(t_1,t_2)}$, second element is $\hat{\lambda}_2^{(t_1,t_2)}$, and so on.
`ratios`	Vector of the L-moment ratios. Second element is $\hat{\tau}^{(t_1,t_2)}$, third element is $\hat{\tau}_3^{(t_1,t_2)}$ and so on.
`trim`	Level of symmetrical trimming used in the computation.
`leftrim`	Level of left-tail trimming used in the computation, which will equal `trim` if symmetrical trimming was used.
`rightrim`	Level of right-tail trimming used in the computation, which will equal `trim` if symmetrical trimming was used.
`source`	An attribute identifying the computational source of the L-moments: “TLmoms”.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

`TLmom`, `lmoms`, and `lmorph`

Examples

```
X1 <- rcauchy(30)
TL <- TLmoms(X1,nmom=6,trim=1)

# This trimming with remove the 1 and the two 4s. All values passed on to the TLmom()
# function then are equal and number of L-moments is too big as well. TLmom() returns
# NaN but these are intercepted and systematically changed to NAs.
TLmoms(c(1,2,2,2,4,4), leftrim=1, rightrim=2, nmom=6)$lambdas
# [1]  2  0  0 NA NA NA

# Example of zero skewness (Berry Boessenkool)
TLmoms(c(3.2, 4.4, 4.8, 2.6, 3.6))
```

`tlmr2par` *Sample Trimmed L-moments to Fitted Distribution*

Description

Parameter estimation of a distribution given initial estimate of the parameters of the distribution to the sample trimmed L-moment (TL-moment) using numerical optimization. Thought the TL-moments can be used with substantial depth into either tail and need not be symmetrically trimmed, the TL-moments do not appear as useful when substantial tail trimming is needed, say for mix population mitigation. Then censored or truncation methods might be preferred. The `x2xlo` family of operations can be used for conditional left-tail truncation, which is not uncommon in frequency analyses of rail-tail interest water resources phenomena.

Usage

```
tlmr2par(x, type, init.para=NULL, trim=NULL, leftrim=NULL, rightrim=NULL, ...)
```

Arguments

`x`	A vector of data values.
`type`	Three character (minimum) distribution type (for example, `type="gev"`, see `dist.list`.
`init.para`	Initial parameters as a vector Θ or as an **lmomco** parameter "object" from say `vec2par`. If a vector is given, then internally `vec2par` is called with distribution equal to `type`.
`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` is in the argument list, the default is 0—the usual L-moment is returned.
`leftrim`	Level of trimming of the left-tail of the sample, which should be left to `NULL` if no or symmetrical trimming is used.
`rightrim`	Level of trimming of the right-tail of the sample, which should be left to `NULL` if no or symmetrical trimming is used.
`...`	Other arguments to pass to the `optim()` function.

Value

An R `list` is returned. This list should contain at least the following items, but some distributions such as the `revgum` have extra.

`type`	The type of distribution in three character (minimum) format.
`para`	The parameters of the distribution.
`text`	Optional material. If the solution fails but the optimization appears to converge, then this element is inserted into the list and the `para` will be all `NA`.
`source`	Attribute specifying source of the parameters.
`rt`	The list from the `optim()` function.
`init.para`	A copy of the initial parameters given.

Author(s)

W.H. Asquith

References

Elamir, E.A.H., and Seheult, A.H., 2003, Trimmed L-moments: Computational Statistics and Data Analysis, v. 43, pp. 299–314.

See Also

theoTLmoms, TLmoms, lmr2par

Examples

```
# (1) An example to check that trim(0,0) should recover whole sample
the.data <- rlmomco(140, vec2par(c(3, 0.4, -0.1), type="pe3"))
wild.guess <- vec2par(c(mean(the.data), 1, 0),    type="pe3")
pe3whole <- lmom2par(lmoms(the.data),            type="pe3")
pe3trimA  <- tlmr2par(the.data, "pe3", init.para=wild.guess, leftrim=0,  rightrim=0)
pe3trimB  <- tlmr2par(the.data, "pe3", init.para=wild.guess, leftrim=10, rightrim=3)
message("PE3 parent       = ", paste0(pe3whole$para, sep=" "))
message("PE3 whole sample = ", paste0(pe3whole$para, sep=" "))
message("PE3 trim( 0, 0)  = ", paste0(pe3trimA$para, sep=" "))
message("PE3 trim(10, 3)  = ", paste0(pe3trimB$para, sep=" ")) #


# (2) An example with "real" outliers
FF <- lmomco::nonexceeds(); qFF <- qnorm(FF); type <- "gev"
the.data <- c(3.064458, 3.139879, 3.167317, 3.225309, 3.324282, 3.330414,
             3.3304140, 3.340444, 3.357935, 3.376577, 3.378398, 3.392697,
             3.4149730, 3.421604, 3.424882, 3.434569, 3.448706, 3.451786,
             3.4517860, 3.462398, 3.465383, 3.469822, 3.491362, 3.501059,
             3.5224440, 3.523746, 3.527630, 3.527630, 3.531479, 3.546543,
             3.5932860, 3.597695, 3.600973, 3.614897, 3.620136, 3.660865,
             3.6848450, 3.820858, 4.708421)
the.data <- sort(the.data) # though already sorted, backup for plotting needs

# visually, looks like 4 outliers to the left and one outlier to the right
# perhaps the practical situation is that we do not wan the left tail to
# mess up the right when fitting a distribution because maybe the practical
# aspects are the that right tail is of engineering interest, but then we
# have some idea that the one very large event is of questionable suitability
t1 <- 4; t2 <- 1 # see left and right trimming and then estimation parameters
whole.para <- lmom2par(lmoms(the.data), type=type)
trim.para  <- tlmr2par(the.data, type, init.para=whole.para, leftrim=t1, rightrim=t2)

n <- length(the.data)
cols <- rep(grey(0.5), n)
pchs <- rep(1, n)
if(t1 != 0) {
  cols[      1 :t1] <- "red"
  cols[(n-t2+1):n ] <- "purple"
```

```
}
if(t2 != 0) {
  pchs[      1 :t1] <- 16
  pchs[(n-t2+1):n ] <- 16
}
plot( qFF, qlmomco(FF, whole.para), type="l", lwd=2, ylim=c(3.1,4.8),
           xlab="Standard normal variate",
           ylab="Some phenomena, log10(cfs)")
lines(qFF, qlmomco(FF, trim.para), col=4, lwd=3)
points(qnorm(pp(the.data)), sort(the.data), pch=pchs, col=cols)
legend("topleft", c("L-moments",
                    paste0("TL-moments(", t1, ",", t2,")")), bty="n",
                   lty=c(1,1), lwd=c(2,3), col=c(1,4))
# see the massive change from the whole sample to the trim(t1,t2) curve
```

---

tlmrcau *Compute Select TL-moment ratios of the Cauchy Distribution*

Description

This function computes select TL-moment ratios of the Cauchy distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α.

Usage

```
tlmrcau(trim=NULL, leftrim=NULL, rightrim=NULL, xi=0, alpha=1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`alpha`	Scale parameter of the distribution.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the theoTLmoms function.

Author(s)

W.H. Asquith

See Also

quacau, theoTLmoms

Examples

```
## Not run:
tlmrcau(trim=2)
tlmrcau(trim=2, xi=2) # another slow example

## End(Not run)
```

---

tlmrexp *Compute Select TL-moment ratios of the Exponential Distribution*

Description

This function computes select TL-moment ratios of the Exponential distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α.

Usage

```
tlmrexp(trim=NULL, leftrim=NULL, rightrim=NULL, xi=0, alpha=1)
```

Arguments

trim	Level of symmetrical trimming to use in the computations. Although NULL in the argument list, the default is 0—the usual L-moment ratios are returned.
leftrim	Level of trimming of the left-tail of the sample.
rightrim	Level of trimming of the right-tail of the sample.
xi	Location parameter of the distribution.
alpha	Scale parameter of the distribution.

Value

An R list is returned.

tau2	A vector of the $\tau_2^{(t_1,t_2)}$ values.
tau3	A vector of the $\tau_3^{(t_1,t_2)}$ values.
tau4	A vector of the $\tau_4^{(t_1,t_2)}$ values.
tau5	A vector of the $\tau_5^{(t_1,t_2)}$ values.
tau6	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the theoTLmoms function.

Author(s)

W.H. Asquith

See Also

quaexp, theoTLmoms

Examples

```
## Not run:
tlmrexp(trim=2)
tlmrexp(trim=2, xi=2) # another slow example

## End(Not run)
```

---

tlmrgev *Compute Select TL-moment ratios of the Generalized Extreme Value Distribution*

Description

This function computes select TL-moment ratios of the Generalized Extreme Value distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α. If the message

```
Error in integrate(XofF, 0, 1) : the integral is probably divergent
```

occurs then careful adjustment of the shape parameter κ parameter range is very likely required. Remember that TL-moments with nonzero trimming permit computation of TL-moments into parameter ranges beyond those recognized for the usual (untrimmed) L-moments.

Usage

```
tlmrgev(trim=NULL, leftrim=NULL, rightrim=NULL,
        xi=0, alpha=1, kbeg=-.99, kend=10, by=.1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`alpha`	Scale parameter of the distribution.
`kbeg`	The beginning κ value of the distribution.
`kend`	The ending κ value of the distribution.
`by`	The increment for the `seq()` between `kbeg` and `kend`.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Author(s)

W.H. Asquith

See Also

`quagev`, `theoTLmoms`

Examples

```
## Not run:
tlmrgev(leftrim=12, rightrim=1, xi=0,   alpha=2 )
tlmrgev(leftrim=12, rightrim=1, xi=100, alpha=20) # another slow example

## End(Not run)
## Not run:
  # Plot and L-moment ratio diagram of Tau3 and Tau4
  # with exclusive focus on the GEV distribution.
  plotlmrdia(lmrdia(), autolegend=TRUE, xleg=-.1, yleg=.6,
             xlim=c(-.8, .7), ylim=c(-.1, .8),
             nolimits=TRUE, noglo=TRUE, nogpa=TRUE, nope3=TRUE,
             nogno=TRUE, nocau=TRUE, noexp=TRUE, nonor=TRUE,
             nogum=TRUE, noray=TRUE, nouni=TRUE)

  # Compute the TL-moment ratios for trimming of one
  # value on the left and four on the right. Notice the
  # expansion of the kappa parameter space from > -1 to
  # something near -5.
  J <- tlmrgev(kbeg=-4.99, leftrim=1, rightrim=4)
  lines(J$tau3, J$tau4, lwd=2, col=3) # BLUE CURVE

  # Compute the TL-moment ratios for trimming of four
  # values on the left and one on the right.
  J <- tlmrgev(kbeg=-1.99, leftrim=4, rightrim=1)
  lines(J$tau3, J$tau4, lwd=2, col=4) # GREEN CURVE

  # The kbeg and kend can be manually changed to see how
  # the resultant curve expands or contracts on the
  # extent of the L-moment ratio diagram.

## End(Not run)
## Not run:
  # Following up, let us plot the two quantile functions
  LM  <- vec2par(c(0,1,-0.99), type='gev', paracheck=FALSE)
  TLM <- vec2par(c(0,1,-4.99), type='gev', paracheck=FALSE)
  F <- nonexceeds()
  plot(qnorm(F),  quagev(F, LM), type="l")
  lines(qnorm(F), quagev(F, TLM, paracheck=FALSE), col=2)
  # Notice how the TLM parameterization runs off towards
  # infinity much much earlier than the conventional
  # near limits of the GEV.

## End(Not run)
```

---

tlmrglo	*Compute Select TL-moment ratios of the Generalized Logistic Distribution*

Description

This function computes select TL-moment ratios of the Generalized Logistic distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α. If the message

```
Error in integrate(XofF, 0, 1) : the integral is probably divergent
```

occurs then careful adjustment of the shape parameter κ parameter range is very likely required. Remember that TL-moments with nonzero trimming permit computation of TL-moments into parameter ranges beyond those recognized for the usual (untrimmed) L-moments.

Usage

```
tlmrglo(trim=NULL, leftrim=NULL, rightrim=NULL,
        xi=0, alpha=1, kbeg=-.99, kend=0.99, by=.1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`alpha`	Scale parameter of the distribution.
`kbeg`	The beginning κ value of the distribution.
`kend`	The ending κ value of the distribution.
`by`	The increment for the `seq()` between `kbeg` and `kend`.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Author(s)

W.H. Asquith

See Also

quaglo, theoTLmoms

Examples

```
## Not run:
tlmrglo(leftrim=1, rightrim=3, xi=0, alpha=4)
tlmrglo(leftrim=1, rightrim=3, xi=32, alpha=83) # another slow example

## End(Not run)
## Not run:
  # Plot and L-moment ratio diagram of Tau3 and Tau4
  # with exclusive focus on the GLO distribution.
  plotlmrdia(lmrdia(), autolegend=TRUE, xleg=-.1, yleg=.6,
             xlim=c(-.8, .7), ylim=c(-.1, .8),
             nolimits=TRUE, nogev=TRUE, nogpa=TRUE, nope3=TRUE,
             nogno=TRUE, nocau=TRUE, noexp=TRUE, nonor=TRUE,
             nogum=TRUE, noray=TRUE, nouni=TRUE)

  # Compute the TL-moment ratios for trimming of one
  # value on the left and four on the right. Notice the
  # expansion of the kappa parameter space from
  # -1 < k < -1 to something larger based on manual
  # adjustments until blue curve encompassed the plot.
  J <- tlmrglo(kbeg=-2.5, kend=1.9, leftrim=1, rightrim=4)
  lines(J$tau3, J$tau4, lwd=2, col=2) # RED CURVE

  # Compute the TL-moment ratios for trimming of four
  # values on the left and one on the right.
  J <- tlmrglo(kbeg=-1.65, kend=3, leftrim=4, rightrim=1)
  lines(J$tau3, J$tau4, lwd=2, col=4) # BLUE CURVE

  # The kbeg and kend can be manually changed to see how
  # the resultant curve expands or contracts on the
  # extent of the L-moment ratio diagram.

## End(Not run)
## Not run:
  # Following up, let us plot the two quantile functions
  LM  <- vec2par(c(0,1,0.99), type='glo', paracheck=FALSE)
  TLM <- vec2par(c(0,1,3.00), type='glo', paracheck=FALSE)
  F <- nonexceeds()
  plot(qnorm(F),  quaglo(F, LM), type="l")
  lines(qnorm(F), quaglo(F, TLM, paracheck=FALSE), col=2)
  # Notice how the TLM parameterization runs off towards
  # infinity much much earlier than the conventional
  # near limits of the GLO.
```

```
## End(Not run)
```

`tlmrgno` *Compute Select TL-moment ratios of the Generalized Normal Distribution*

Description

This function computes select TL-moment ratios of the Generalized Normal distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α. If the message

```
Error in integrate(XofF, 0, 1) : the integral is probably divergent
```

occurs then careful adjustment of the shape parameter κ parameter range is very likely required. Remember that TL-moments with nonzero trimming permit computation of TL-moments into parameter ranges beyond those recognized for the usual (untrimmed) L-moments.

Usage

```
tlmrgno(trim=NULL, leftrim=NULL, rightrim=NULL,
        xi=0, alpha=1, kbeg=-3, kend=3, by=.1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`alpha`	Scale parameter of the distribution.
`kbeg`	The beginning κ value of the distribution.
`kend`	The ending κ value of the distribution.
`by`	The increment for the `seq()` between `kbeg` and `kend`.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the theoTLmoms function.

Author(s)

W.H. Asquith

See Also

quagno, theoTLmoms, tlmrln3

Examples

```
## Not run:
tlmrgno(leftrim=3, rightrim=2, xi=0, alpha=2)
tlmrgno(leftrim=3, rightrim=2, xi=120, alpha=55) # another slow example

## End(Not run)
## Not run:
  # Plot and L-moment ratio diagram of Tau3 and Tau4
  # with exclusive focus on the GNO distribution.
  plotlmrdia(lmrdia(), autolegend=TRUE, xleg=-.1, yleg=.6,
             xlim=c(-.8, .7), ylim=c(-.1, .8),
             nolimits=TRUE, nogev=TRUE, nogpa=TRUE, nope3=TRUE,
             noglo=TRUE, nocau=TRUE, noexp=TRUE, nonor=TRUE,
             nogum=TRUE, noray=TRUE, nouni=TRUE)

  # Compute the TL-moment ratios for trimming of one
  # value on the left and four on the right.
  J <- tlmrgno(kbeg=-3.5, kend=3.9, leftrim=1, rightrim=4)
  lines(J$tau3, J$tau4, lwd=2, col=2) # RED CURVE

  # Compute the TL-moment ratios for trimming of four
  # values on the left and one on the right.
  J <- tlmrgno(kbeg=-4, kend=4, leftrim=4, rightrim=1)
  lines(J$tau3, J$tau4, lwd=2, col=4) # BLUE CURVE

  # The kbeg and kend can be manually changed to see how
  # the resultant curve expands or contracts on the
  # extent of the L-moment ratio diagram.

## End(Not run)
## Not run:
  # Following up, let us plot the two quantile functions
  LM  <- vec2par(c(0,1,0.99), type='gno', paracheck=FALSE)
  TLM <- vec2par(c(0,1,3.00), type='gno', paracheck=FALSE)
  F <- nonexceeds()
  plot(qnorm(F),  quagno(F, LM), type="l")
  lines(qnorm(F), quagno(F, TLM, paracheck=FALSE), col=2)
  # Notice how the TLM parameterization runs off towards
  # infinity much much earlier than the conventional
```

```
  # near limits of the GNO.

## End(Not run)
```

---

tlmrgpa *Compute Select TL-moment ratios of the Generalized Pareto*

Description

This function computes select TL-moment ratios of the Generalized Pareto distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α. If the message

```
Error in integrate(XofF, 0, 1) : the integral is probably divergent
```

occurs then careful adjustment of the shape parameter κ parameter range is very likely required. Remember that TL-moments with nonzero trimming permit computation of TL-moments into parameter ranges beyond those recognized for the usual (untrimmed) L-moments.

Usage

```
tlmrgpa(trim=NULL, leftrim=NULL, rightrim=NULL,
        xi=0, alpha=1, kbeg=-.99, kend=10, by=.1)
```

Arguments

trim	Level of symmetrical trimming to use in the computations. Although NULL in the argument list, the default is 0—the usual L-moment ratios are returned.
leftrim	Level of trimming of the left-tail of the sample.
rightrim	Level of trimming of the right-tail of the sample.
xi	Location parameter of the distribution.
alpha	Scale parameter of the distribution.
kbeg	The beginning κ value of the distribution.
kend	The ending κ value of the distribution.
by	The increment for the seq() between kbeg and kend.

Value

An R list is returned.

tau2	A vector of the $\tau_2^{(t_1,t_2)}$ values.
tau3	A vector of the $\tau_3^{(t_1,t_2)}$ values.
tau4	A vector of the $\tau_4^{(t_1,t_2)}$ values.
tau5	A vector of the $\tau_5^{(t_1,t_2)}$ values.
tau6	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the theoTLmoms function.

Author(s)

W.H. Asquith

See Also

quagpa, theoTLmoms

Examples

```
## Not run:
tlmrgpa(leftrim=7, rightrim=2, xi=0, alpha=31)
tlmrgpa(leftrim=7, rightrim=2, xi=143, alpha=98) # another slow example

## End(Not run)
## Not run:
  # Plot and L-moment ratio diagram of Tau3 and Tau4
  # with exclusive focus on the GPA distribution.
  plotlmrdia(lmrdia(), autolegend=TRUE, xleg=-.1, yleg=.6,
             xlim=c(-.8, .7), ylim=c(-.1, .8),
             nolimits=TRUE, nogev=TRUE, noglo=TRUE, nope3=TRUE,
             nogno=TRUE, nocau=TRUE, noexp=TRUE, nonor=TRUE,
             nogum=TRUE, noray=TRUE, nouni=TRUE)

  # Compute the TL-moment ratios for trimming of one
  # value on the left and four on the right. Notice the
  # expansion of the kappa parameter space from k > -1.
  J <- tlmrgpa(kbeg=-3.2, kend=50, by=.05, leftrim=1, rightrim=4)
  lines(J$tau3, J$tau4, lwd=2, col=2) # RED CURVE
  # Notice the gap in the curve near tau3 = 0.1

  # Compute the TL-moment ratios for trimming of four
  # values on the left and one on the right.
  J <- tlmrgpa(kbeg=-1.6, kend=8, leftrim=4, rightrim=1)
  lines(J$tau3, J$tau4, lwd=2, col=3) # GREEN CURVE

  # The kbeg and kend can be manually changed to see how
  # the resultant curve expands or contracts on the
  # extent of the L-moment ratio diagram.

## End(Not run)
## Not run:
  # Following up, let us plot the two quantile functions
  LM  <- vec2par(c(0,1,0.99), type='gpa', paracheck=FALSE)
  TLM <- vec2par(c(0,1,3.00), type='gpa', paracheck=FALSE)
  F <- nonexceeds()
  plot(qnorm(F),  quagpa(F, LM), type="l")
  lines(qnorm(F), quagpa(F, TLM, paracheck=FALSE), col=2)
```

```
  # Notice how the TLM parameterization runs off towards
  # infinity much much earlier than the conventional
  # near limits of the GPA.

## End(Not run)
```

`tlmrgum` *Compute Select TL-moment ratios of the Gumbel Distribution*

Description

This function computes select TL-moment ratios of the Gumbel distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α.

Usage

```
tlmrgum(trim=NULL, leftrim=NULL, rightrim=NULL, xi=0, alpha=1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`alpha`	Scale parameter of the distribution.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Author(s)

W.H. Asquith

See Also

quagum, theoTLmoms

Examples

```
## Not run:
tlmrgum(trim=2)
tlmrgum(trim=2, xi=2) # another slow example

## End(Not run)
```

---

tlmrln3 — *Compute Select TL-moment ratios of the 3-Parameter Log-Normal Distribution*

Description

This function computes select TL-moment ratios of the Log-Normal3 distribution for defaults of $\zeta = 0$ and $\mu_{\mathrm{log}} = 0$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ζ and μ_{log}. If the message

```
Error in integrate(XofF, 0, 1) : the integral is probably divergent
```

occurs then careful adjustment of the shape parameter σ_{log} parameter range is very likely required. Remember that TL-moments with nonzero trimming permit computation of TL-moments into parameter ranges beyond those recognized for the usual (untrimmed) L-moments.

Usage

```
tlmrln3(trim=NULL, leftrim=NULL, rightrim=NULL,
        zeta=0, mulog=0, sbeg=0.01, send=3.5, by=.1)
```

Arguments

trim	Level of symmetrical trimming to use in the computations. Although NULL in the argument list, the default is 0—the usual L-moment ratios are returned.
leftrim	Level of trimming of the left-tail of the sample.
rightrim	Level of trimming of the right-tail of the sample.
zeta	Location parameter of the distribution.
mulog	Mean of the logarithms of the distribution.
sbeg	The beginning σ_{log} value of the distribution.
send	The ending σ_{log} value of the distribution.
by	The increment for the seq() between sbeg and send.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Author(s)

W.H. Asquith

See Also

`qualn3`, `theoTLmoms`, `tlmrgno`

Examples

```
## Not run:
  # Recalling that generalized Normal and log-Normal3 are
  # the same with the GNO being the more general.

  # Plot and L-moment ratio diagram of Tau3 and Tau4
  # with exclusive focus on the GNO distribution.
  plotlmrdia(lmrdia(), autolegend=TRUE, xleg=-.1, yleg=.6,
             xlim=c(-.8, .7), ylim=c(-.1, .8),
             nolimits=TRUE, noglo=TRUE, nogpa=TRUE, nope3=TRUE,
             nogev=TRUE, nocau=TRUE, noexp=TRUE, nonor=TRUE,
             nogum=TRUE, noray=TRUE, nouni=TRUE)

  LN3 <- tlmrln3(sbeg=.001, mulog=-1)
  lines(LN3$tau3, LN3$tau4) # See how it overplots the GNO
  # for right skewness. So only part of the GNO is covered.

  # Compute the TL-moment ratios for trimming of one
  # value on the left and four on the right.
  J <- tlmrgno(kbeg=-3.5, kend=3.9, leftrim=1, rightrim=4)
  lines(J$tau3, J$tau4, lwd=2, col=2) # RED CURVE

  LN3 <- tlmrln3(, leftrim=1, rightrim=4, sbeg=.001)
  lines(LN3$tau3, LN3$tau4) # See how it again over plots
  # only part of the GNO

## End(Not run)
```

`tlmrnor` *Compute Select TL-moment ratios of the Normal Distribution*

Description

This function computes select TL-moment ratios of the Normal distribution for defaults of $\mu = 0$ and $\sigma = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for μ and σ.

Usage

```
tlmrnor(trim=NULL, leftrim=NULL, rightrim=NULL, mu=0, sigma=1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`mu`	Location parameter (mean) of the distribution.
`sigma`	Scale parameter (standard deviation) of the distribution.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.
`tau5`	A vector of the $\tau_5^{(t_1,t_2)}$ values.
`tau6`	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Author(s)

W.H. Asquith

See Also

`quanor`, `theoTLmoms`

Examples

```
## Not run:
tlmrnor(leftrim=2, rightrim=1)
tlmrnor(leftrim=2, rightrim=1, mu=100, sigma=1000) # another slow example

## End(Not run)
```

`tlmrpe3` *Compute Select TL-moment ratios of the Pearson Type III*

Description

This function computes select TL-moment ratios of the Pearson Type III distribution for defaults of $\xi = 0$ and $\beta = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α. If the message

```
Error in integrate(XofF, 0, 1) : the integral is probably divergent
```

occurs then careful adjustment of the shape parameter β parameter range is very likely required. Remember that TL-moments with nonzero trimming permit computation of TL-moments into parameter ranges beyond those recognized for the usual (untrimmed) L-moments. The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Usage

```
tlmrpe3(trim=NULL, leftrim=NULL, rightrim=NULL,
        xi=0, beta=1, abeg=-.99, aend=0.99, by=.1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`beta`	Scale parameter of the distribution.
`abeg`	The beginning α value of the distribution.
`aend`	The ending α value of the distribution.
`by`	The increment for the `seq()` between `abeg` and `aend`.

Value

An R `list` is returned.

`tau2` A vector of the $\tau_2^{(t_1,t_2)}$ values.

`tau3` A vector of the $\tau_3^{(t_1,t_2)}$ values.

`tau4` A vector of the $\tau_4^{(t_1,t_2)}$ values.

`tau5` A vector of the $\tau_5^{(t_1,t_2)}$ values.

`tau6` A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the `theoTLmoms` function.

Author(s)

W.H. Asquith

See Also

`quape3`, `theoTLmoms`

Examples

```
## Not run:
tlmrpe3(leftrim=2, rightrim=4, xi=0, beta=2)
tlmrpe3(leftrim=2, rightrim=4, xi=100, beta=20) # another slow example
  # Plot and L-moment ratio diagram of Tau3 and Tau4
  # with exclusive focus on the PE3 distribution.
  plotlmrdia(lmrdia(), autolegend=TRUE, xleg=-.1, yleg=.6,
             xlim=c(-.8, .7), ylim=c(-.1, .8),
             nolimits=TRUE, nogev=TRUE, nogpa=TRUE, noglo=TRUE,
             nogno=TRUE, nocau=TRUE, noexp=TRUE, nonor=TRUE,
             nogum=TRUE, noray=TRUE, nouni=TRUE)

  # Compute the TL-moment ratios for trimming of one
  # value on the left and four on the right. Notice the
  # expansion of the alpha parameter space from
  # -1 < a < -1 to something larger based on manual
  # adjustments until blue curve encompassed the plot.
  J <- tlmrpe3(abeg=-15, aend=6, leftrim=1, rightrim=4)
  lines(J$tau3, J$tau4, lwd=2, col=2) # RED CURVE

  # Compute the TL-moment ratios for trimming of four
  # values on the left and one on the right.
  J <- tlmrpe3(abeg=-6, aend=10, leftrim=4, rightrim=1)
  lines(J$tau3, J$tau4, lwd=2, col=4) # BLUE CURVE

  # The abeg and aend can be manually changed to see how
```

```
  # the resultant curve expands or contracts on the
  # extent of the L-moment ratio diagram.

## End(Not run)
## Not run:
  # Following up, let us plot the two quantile functions
  LM  <- vec2par(c(0,1,0.99), type='pe3', paracheck=FALSE)
  TLM <- vec2par(c(0,1,3.00), type='pe3', paracheck=FALSE)
  F <- nonexceeds()
  plot(qnorm(F),  quape3(F, LM), type="l")
  lines(qnorm(F), quape3(F, TLM, paracheck=FALSE), col=2)
  # Notice how the TLM parameterization runs off towards
  # infinity much much earlier than the conventional
  # near limits of the PE3.

## End(Not run)
```

---

tlmrray *Compute Select TL-moment ratios of the Rayleigh Distribution*

Description

This function computes select TL-moment ratios of the Rayleigh distribution for defaults of $\xi = 0$ and $\alpha = 1$. This function can be useful for plotting the trajectory of the distribution on TL-moment ratio diagrams of $\tau_2^{(t_1,t_2)}$, $\tau_3^{(t_1,t_2)}$, $\tau_4^{(t_1,t_2)}$, $\tau_5^{(t_1,t_2)}$, and $\tau_6^{(t_1,t_2)}$. In reality, $\tau_2^{(t_1,t_2)}$ is dependent on the values for ξ and α.

Usage

```
tlmrray(trim=NULL, leftrim=NULL, rightrim=NULL, xi=0, alpha=1)
```

Arguments

`trim`	Level of symmetrical trimming to use in the computations. Although `NULL` in the argument list, the default is 0—the usual L-moment ratios are returned.
`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`xi`	Location parameter of the distribution.
`alpha`	Scale parameter of the distribution.

Value

An R `list` is returned.

`tau2`	A vector of the $\tau_2^{(t_1,t_2)}$ values.
`tau3`	A vector of the $\tau_3^{(t_1,t_2)}$ values.
`tau4`	A vector of the $\tau_4^{(t_1,t_2)}$ values.

tau5	A vector of the $\tau_5^{(t_1,t_2)}$ values.
tau6	A vector of the $\tau_6^{(t_1,t_2)}$ values.

Note

The function uses numerical integration of the quantile function of the distribution through the theoTLmoms function.

Author(s)

W.H. Asquith

See Also

quaray, theoTLmoms

Examples

```
## Not run:
tlmrray(leftrim=2, rightrim=1, xi=0, alpha=2)
tlmrray(leftrim=2, rightrim=1, xi=10, alpha=2) # another slow example

## End(Not run)
```

---

tttlmomco *Total Time on Test Transform of Distributions*

Description

This function computes the Total Time on Test Transform Quantile Function for a quantile function $x(F)$ (par2qua, qlmomco). The TTT is defined by Nair et al. (2013, p. 171–172, 176) has several expressions

$$T(u) = \mu - (1-u)M(u),$$
$$T(u) = x(u) - uR(u),$$
$$T(u) = (1-u)x(u) + \mu L(u),$$

where $T(u)$ is the total time on test for nonexceedance probability u, $M(u)$ is the residual mean quantile function (rmlmomco), $x(u)$ is a constant for $x(F = u)$, $R(u)$ is the reversed mean residual quantile function (rrmlmomco), $L(u)$ is the Lorenz curve (lrzlmomco), and μ as the following definitions

$$\mu \equiv \lambda_1(u = 0) \text{ first L-moment of residual life for } u = 0,$$
$$\mu \equiv \lambda_1(x(F)) \text{ first L-moment of the quantile function,}$$
$$\mu \equiv \mu(0) \text{ conditional mean for } u = 0.$$

The definitions imply that within numerical tolerances that $\mu(0)$ (cmlmomco) should be equal to $T(1)$, which means that the conditional mean that the 0th percentile in life has been reached equals that total time on test for the 100th percentile. The later can be interpreted as meaning that each of realization of the lifetime distribution for the respective sample size lived to its expected ordered lifetimes.

Usage

```
tttlmomco(f, para)
```

Arguments

f	Nonexceedance probability ($0 \leq F \leq 1$).
para	The parameters from lmom2par or vec2par.

Value

Total time on test value for F.

Note

The second definition for μ is used and in **lmomco** code the implementation for nonexceedance probability f and parameter object para is

```
Tu <- par2qua(f, para) - f*rrmlmomco(f, para) # 2nd def.
```

but other possible implementations for the first and third definitions respectively are

```
Tu <- cmlmomco(f=0, para) - (1-f)*rmlmomco(f, para) # 1st def.
Tu <- (1-f)*par2qua(f, para) + cmlmomco(f=0, para)*lrzlmomco(f, para) # 3rd def.
```

Author(s)

W.H. Asquith

References

Nair, N.U., Sankaran, P.G., and Balakrishnan, N., 2013, Quantile-based reliability analysis: Springer, New York.

See Also

qlmomco, rmlmomco, rrmlmomco, lrzlmomco

Examples

```
# It is easiest to think about residual life as starting at the origin, units in days.
A <- vec2par(c(0.0, 2649, 2.11), type="gov") # so set lower bounds = 0.0
tttlmomco(0.5, A)  # The median lifetime = 859 days

f <- c(0.25,0.75) # All three computations report: 306.2951 and 1217.1360 days.
Tu1 <- cmlmomco(f=0, A) - (1-f)* rmlmomco(f, A)
Tu2 <-    par2qua(f, A) -    f * rrmlmomco(f, A)
Tu3 <- (1-f)*par2qua(f, A) + cmlmomco(f=0, A)*lrzlmomco(f, A)

if(abs(cmlmomco(0,A) - tttlmomco(1,A)) < 1E-4) {
   print("These two quantities should be nearly identical.\n")
}
```

tulia6Eprecip — *Annual Maximum Precipitation Data for Tulia 6E, Texas*

Description

Annual maximum precipitation data for Tulia 6E, Texas

Usage

```
data(tulia6Eprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(tulia6Eprecip)
summary(tulia6Eprecip)
```

tuliaprecip — *Annual Maximum Precipitation Data for Tulia, Texas*

Description

Annual maximum precipitation data for Tulia, Texas

Usage

```
data(tuliaprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(tuliaprecip)
summary(tuliaprecip)
```

TX38lgtrmFlow — *First six L-moments of logarithms of annual mean streamflow and variances for 35 selected long-term U.S. Geological Survey streamflow-gaging stations in Texas*

Description

L-moments of annual mean streamflow for 35 long-term U.S. Geological Survey (USGS) streamflow-gaging stations (streamgages) with at least 49 years of natural and unregulated record through water year 2012 (Asquith and Barbie, 2014). Logarithmic transformations of annual mean streamflow at each of the 35 streamgages were done. For example, logarithmic transformation of strictly positive hydrologic data is done to avoid conditional probability adjustment for the zero values; values equal to zero must be offset to avoid using a logarithm of zero. A mathematical benefit of using logarithmic transformation is that probability distributions with infinite lower and upper limits become applicable. An arbitrary value of 10 cubic feet per second was added to the streamflows for each of the 35 streamgages prior to logarithmic transformation to accommodate mean annual streamflows equal to zero (no flow). These data should be referred to as the offset-annual mean streamflow. The offsetting along the real-number line permits direct use of logarithmic transformations without the added complexity of conditional probability adjustment for zero values in magnitude and frequency analyses.

The first six sample L-moments of the base-10 logarithms of the offset-annual mean streamflow were computed using the `lmoms(..., nmom=6)`. The sampling variances of each corresponding L-moment are used to compute regional or study-area values for the L-moments through weighted-mean computation. The available years of record for each of 35 stations is so large as to produce severe numerical problems in matrices needed for sampling variances using the recently developed the exact-analytical bootstrap for L-moments method (Wang and Hutson, 2013) (`lmoms.bootbarvar`). In order to compute sampling variances for each of the sample L-moments for each streamgage, replacement-bootstrap simulation using the `sample(..., replace=TRUE)` function with 10,000 replications with replacement.

Usage

```
data(TX38lgtrmFlow)
```

Format

An R `data.frame` with

STATION The USGS streamgage number.

YEARS The number of years of data record.

Mean The arthimetic mean (λ_1) of $\log_{10}(x+10)$, where x is the vector of data.

Lscale The L-scale (λ_2) of the log10-offset data.

LCV The coefficient of L-variation (τ_2) of the log10-offset data.

Lskew The L-skew (τ_3) of the log10-offset data.

Lkurtosis The L-kurtosis (τ_4) of the log10-offset data.

Tau5 The τ_5 of the log10-offset data.

Tau6 The τ_6 of the log10-offset data.

VarMean The estimated sampling variance for λ_1 multiplied by 1000.

VarLscale The estimated sampling variance for λ_2 multiplied by 1000.

VarLCV The estimated sampling variance for τ_2 multiplied by 1000.

VarLskew The estimated sampling variance for τ_3 multiplied by 1000.

VarLkurtosis The estimated sampling variance for τ_4 multiplied by 1000.

VarTau5 The estimated sampling variance for τ_5 multiplied by 1000.

VarTau6 The estimated sampling variance for τ_6 multiplied by 1000.

Note

The title of this dataset indicates 35 stations, and 35 stations is the length of the data. The name of the dataset `TX38lgtrmFlow` and the source of the data (Asquith and Barbie, 2014) reflects 38 stations. It was decided to not show the data for 3 of the stations because a trend was detected but the dataset had already been named. The inconsistency will have to stand.

References

Asquith, W.H., and Barbie, D.L., 2014, Trend analysis and selected summary statistics of annual mean streamflow for 38 selected long-term U.S. Geological Survey streamflow-gaging stations in Texas, water years 1916–2012: U.S. Geological Survey Scientific Investigations Report 2013–5230, 16 p.

Wang, D., and Hutson, A.D., 2013, Joint confidence region estimation of L-moments with an extension to right censored data: Journal of Applied Statistics, v. 40, no. 2, pp. 368–379.

Examples

```
data(TX38lgtrmFlow)
summary(TX38lgtrmFlow)
## Not run:
# Need to load libraries in this order
library(lmomco); library(lmomRFA)
data(TX38lgtrmFlow)
TxDat <- TX38lgtrmFlow
```

```
TxDat <- TxDat[,-c(4)]; TxDat <- TxDat[,-c(8:15)]
summary(regtst(TxDat))
TxDat2 <- TxDat[-c(11, 28),] # Remove 08082700 Millers Creek near Munday
                             # Remove 08190500 West Nueces River at Brackettville
# No explanation for why Millers Creek is so radically discordant with the other
# streamgages with the possible exception that its data record does not span the
# drought of the 1950s like many of the other streamgages.
# The West Nueces is a highly different river from even nearby streamgages. It
# is a problem in flood frequency analysis too. So not surprizing to see this
# streamgage come up as discordant.
summary(regtst(TxDat2))
S <- summary(regtst(TxDat2))
# The results suggest that none of the 3-parameter distributions are suitable.
# The bail out solution using the Wakeby distribution is accepted. Our example
# will continue on by consideration of the two 4-parameter distributions
# available. A graphical comparison between three frequency curves will be made.
kap <- S$rpara
rmom <- S$rmom
lmr <- vec2lmom(rmom, lscale=FALSE)
aep <- paraep4(lmr)
F <- as.numeric(unlist(attributes(S$quant)$dimnames[2]))
plot(qnorm(F), S$quant[6,], type="l", lwd=3, lty=2,
     xlab="Nonexceedance probability (as standard normal variate)",
     ylab="Frequency factor (dimensionless)")
lines(qnorm(F), quakap(F, kap), col=4, lwd=2)
lines(qnorm(F), quaaep4(F, aep), col=2)
legend(-1, 0.8, c("Wakeby distribution (5 parameters)",
                  "Kappa distribution (4 parameters)",
                  "Asymmetrical Exponential Power distribution (4 parameters)"),
       bty = "n", cex=0.75, lwd=c(3,2,1), lty=c(2,1,1), col=c(1,4,2)
      )
# Based on general left tail behavior the Wakeby distribution is not acceptable.
# Based on general right tail behavior the AEP is preferred.
#
# It is recognized that the regional analysis provided by regtst() indicates
# substantial heterogeneity by all three definitions of that statistic. Further
# analysis to somehow compensate for climatological and general physiographic
# differences between the watersheds might be able to compensate for the
# heterogeneity. Such an effort is outside scope of this example.
#
# Suppose that the following data set is available for particular stream site from
# a short record streamgage, let us apply the dimensionless frequency curve as
# defined by the asymmetric exponential power distribution. Lettuce also use the
# 50-year drought as an example. This recurrence interval has a nonexceedance
# probability of 0.02. Lastly, there is the potential with this particular process
# to compute a negative annual mean streamflow, when this happens truncate to zero.
data <- c(11.9, 42.8, 36, 20.4, 43.8, 30.7, 91.1, 54.7, 43.7, 17, 28.7, 20.5, 81.2)
xbar <- mean(log10(data + 10)) # shift, log, and mean
# Note the application of the "the index method" within the exponentiation.
tmp.quantile <- 10^(xbar*quaaep4(0.02, aep)) - 10 # detrans, offset
Q50yeardrought <- ifelse(tmp.quantile < 0, 0, tmp.quantile)
# The value is 2.53 cubic feet per second average streamflow.
```

```
## End(Not run)
```

`USGSsta01515000peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 01515000*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 01515000. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta01515000peaks)
```

Format

An R `data.frame` with

Date The date of the annual peak streamflow.

Streamflow Annual peak streamflow data in cubic feet per second.

Flags Qualification flags on the streamflow data.

Stage Annual peak stage (gage height, river height) in feet.

Flags.1 Qualification flags on the gage height data.

Examples

```
data(USGSsta01515000peaks)
## Not run: plot(USGSsta01515000peaks)
```

`USGSsta02366500peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 02366500*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 02366500. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta02366500peaks)
```

Format

An R `data.frame` with

Date The date of the annual peak streamflow.

Streamflow Annual peak streamflow data in cubic feet per second.

Flags Qualification flags on the streamflow data.

Stage Annual peak stage (gage height, river height) in feet.

Flags.1 Qualification flags on the gage height data.

Examples

```
data(USGSsta02366500peaks)
## Not run: plot(USGSsta02366500peaks)
```

`USGSsta05405000peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 05405000*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 05405000. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta05405000peaks)
```

Format

An R `data.frame` with

agency_cd Agency code.

site_no Agency station number.

peak_dt The date of the annual peak streamflow.

peak_tm Time of the peak streamflow.

peak_va Annual peak streamflow data in cubic feet per second.

peak_cd Qualification flags on the streamflow data.

gage_ht Annual peak stage (gage height, river height) in feet.

gage_ht_cd Qualification flags on the gage height data.

year_last_pk Peak streamflow reported is the highest since this year.

ag_dt Date of maximum gage-height for water year (if not concurrent with peak).

ag_tm Time of maximum gage-height for water year (if not concurrent with peak).

ag_gage_ht Maximum gage height for water year in feet (if not concurrent with peak).

ag_gage_ht_cd Maximum gage height code.

Examples

```
data(USGSsta05405000peaks)
## Not run: plot(USGSsta05405000peaks)
```

USGSsta06766000dvs *Daily Mean Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 06766000*

Description

Daily mean streamflow data for U.S. Geological Survey streamflow-gaging station 06766000 PLATTE RIVER AT BRADY, NE. The qualification code `X01_00060_00003_cd` values are:

A Approved for publication — Processing and review completed.

1 Daily value is write protected without any remark code to be printed.

Usage

```
data(USGSsta06766000dvs)
```

Format

An R `data.frame` with

agency_cd The agency code `USGS`.

site_no The station identification number.

datetime The date and time of the data.

X01_00060_00003 The daily mean streamflow data in cubic feet per second.

X01_00060_00003_cd A code on the data value.

Examples

```
data(USGSsta06766000dvs)
## Not run: plot(USGSsta06766000dvs)
```

USGSsta08151500peaks *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 08151500*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 08151500. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta08151500peaks)
```

Format

An R `data.frame` with

Date The date of the annual peak streamflow.

Streamflow Annual peak streamflow data in cubic feet per second.

Flags Qualification flags on the streamflow data.

Stage Annual peak stage (gage height, river height) in feet.

Examples

```
data(USGSsta08151500peaks)
## Not run: plot(USGSsta08151500peaks)
```

`USGSsta08167000peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 08167000*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 08167000. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta08167000peaks)
```

Format

An R `data.frame` with

agency_cd Agency code.

site_no Agency station number.

peak_dt The date of the annual peak streamflow.

peak_tm Time of the peak streamflow.

peak_va Annual peak streamflow data in cubic feet per second.

peak_cd Qualification flags on the streamflow data.

gage_ht Annual peak stage (gage height, river height) in feet.

gage_ht_cd Qualification flags on the gage height data.

year_last_pk Peak streamflow reported is the highest since this year.

ag_dt Date of maximum gage-height for water year (if not concurrent with peak).

ag_tm Time of maximum gage-height for water year (if not concurrent with peak).

ag_gage_ht Maximum gage height for water year in feet (if not concurrent with peak).

ag_gage_ht_cd Maximum gage height code.

Examples

```
data(USGSsta08167000peaks)
## Not run: plot(USGSsta08167000peaks)
```

`USGSsta08190000peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 08190000*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 08190000. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta08190000peaks)
```

Format

An R `data.frame` with

agency_cd Agency code.

site_no Agency station number.

peak_dt The date of the annual peak streamflow.

peak_tm Time of the peak streamflow.

peak_va Annual peak streamflow data in cubic feet per second.

peak_cd Qualification flags on the streamflow data.

gage_ht Annual peak stage (gage height, river height) in feet.

gage_ht_cd Qualification flags on the gage height data.

year_last_pk Peak streamflow reported is the highest since this year.

ag_dt Date of maximum gage-height for water year (if not concurrent with peak).

ag_tm Time of maximum gage-height for water year (if not concurrent with peak).

ag_gage_ht Maximum gage height for water year in feet (if not concurrent with peak).

ag_gage_ht_cd Maximum gage height code.

Examples

```
data(USGSsta08190000peaks)
## Not run: plot(USGSsta08190000peaks)
```

`USGSsta09442000peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 09442000*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 09442000. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta09442000peaks)
```

Format

An R `data.frame` with

Date The date of the annual peak streamflow.

Streamflow Annual peak streamflow data in cubic feet per second.

Flags Qualification flags on the streamflow data.

Stage Annual peak stage (gage height, river height) in feet.

Examples

```
data(USGSsta09442000peaks)
## Not run: plot(USGSsta09442000peaks)
```

`USGSsta14321000peaks` *Annual Peak Streamflow Data for U.S. Geological Survey Streamflow-Gaging Station 14321000*

Description

Annual peak streamflow data for U.S. Geological Survey streamflow-gaging station 14321000. The peak streamflow-qualification codes `Flag` are:

1 Discharge is a Maximum Daily Average

2 Discharge is an Estimate

3 Discharge affected by Dam Failure

4 Discharge less than indicated value, which is Minimum Recordable Discharge at this site

5 Discharge affected to unknown degree by Regulation or Diversion

6 Discharge affected by Regulation or Diversion

7 Discharge is an Historic Peak

8 Discharge actually greater than indicated value

9 Discharge due to Snowmelt, Hurricane, Ice-Jam or Debris Dam breakup

A Year of occurrence is unknown or not exact

B Month or Day of occurrence is unknown or not exact

C All or part of the record affected by Urbanization, Mining, Agricultural changes, Channelization, or other

D Base Discharge changed during this year

E Only Annual Maximum Peak available for this year

The gage height qualification codes `Flag.1` are:

1 Gage height affected by backwater

2 Gage height not the maximum for the year

3 Gage height at different site and(or) datum

4 Gage height below minimum recordable elevation

5 Gage height is an estimate

6 Gage datum changed during this year

Usage

```
data(USGSsta14321000peaks)
```

Format

An R `data.frame` with

Date The date of the annual peak streamflow.

Streamflow Annual peak streamflow data in cubic feet per second.

Flags Qualification flags on the streamflow data.

Stage Annual peak stage (gage height, river height) in feet.

Flags.1 Qualification flags on the gage height data.

Examples

```
data(USGSsta14321000peaks)
## Not run: plot(USGSsta14321000peaks)
```

---

vec2lmom *Convert a Vector of L-moments to a L-moment Object*

Description

This function converts a vector of L-moments to a L-moment object of **lmomco**. The object is an R `list`. This function is intended to facilitate the use of L-moments (and TL-moments) that the user might have from other sources. L-moments and L-moment ratios of arbitrary length are supported.

Because in typical practice, the $k \geq 3$ order L-moments are dimensionless ratios (τ_3, τ_4, and τ_5), this function computes λ_3, λ_4, λ_5 from λ_2 from the ratios. However, typical practice is not set on the use of λ_2 or τ as measure of dispersion. Therefore, this function takes an `lscale` optional logical (`TRUE|FALSE`) argument—if λ_2 is provided and `lscale=TRUE`, then τ is computed by the function and if τ is provided, then λ_2 is computed by the function.

Usage

```
vec2lmom(vec, lscale=TRUE,
         trim=NULL, leftrim=NULL, rightrim=NULL, checklmom=TRUE)
```

Arguments

`vec`	A vector of L-moment values in λ_1, λ_2 or τ, τ_3, τ_4, and τ_5 order.
`lscale`	A logical switch on the type of the second value of first argument. L-scale (λ_2) or LCV (τ). Default is `TRUE`, the second value in the first argument is λ_2.
`trim`	Level of symmetrical trimming, which should equal `NULL` if asymmetrical trimming is used.

leftrim
: Level of trimming of the left-tail of the sample, which will equal `NULL` even if `trim = 1` if the trimming is symmetrical.

rightrim
: Level of trimming of the right-tail of the sample, which will equal `NULL` even if `trim = 1` if the trimming is symmetrical.

checklmom
: Should the `lmom` be checked for validity using the `are.lmom.valid` function. Normally this should be left as the default unless TL-moments are being constructed in lieu of using `vec2TLmom`.

Value

An R `list` is returned.

Author(s)

W.H. Asquith

See Also

`lmoms`, `vec2pwm`

Examples

```
lmr <- vec2lmom(c(12,0.6,0.34,0.20,0.05),lscale=FALSE)
```

---

vec2par *Convert a Vector of Parameters to a Parameter Object of a Distribution*

Description

This function converts a vector of parameters to a parameter object of a distribution. The type of distribution is specified in the argument list: `aep4`, `cau`, `exp`, `gam`, `gep`, `gev`, `glo`, `gno`, `gpa`, `gum`, `kap`, `kur`, `lap`, `lmrq`, `ln3`, `nor`, `pe3`, `ray`, `revgum`, `rice`, `st3`, `texp`, `wak`, and `wei`. These abbreviations and only these are used in routing logic within **lmomco**. There is no provision for fuzzy matching. However, if the distribution type is not identified, then the function issues a warning, but goes ahead and creates the parameter list and of course can not check for the validity of the parameters. If one has a need to determine on-the-fly the number of parameters in a distribution as supported in **lmomco**, then see the `dist.list` function.

Usage

```
vec2par(vec, type, nowarn=FALSE, paracheck=TRUE, ...)
```

Arguments

- `vec` — A vector of parameter values for the distribution specified by type.
- `type` — Three character distribution type (for example, `type='gev'`).
- `nowarn` — A logical switch on warning suppression. If `TRUE` then `options(warn=-1)` is made and restored on return. This switch is to permit calls in which warnings are not desired as the user knows how to handle the returned value—say in an optimization algorithm.
- `paracheck` — A logical controlling whether the parameters and checked for validity. Overriding of this check might be extremely important and needed for use of the distribution quantile function in the context of TL-moments with nonzero trimming.
- `...` — Additional arguments for the `are.par.valid` call that is made internally.

Details

If the distribution is a Reverse Gumbel (`type=revgum`) or Generalized Pareto (`type=gpa`), which are 2-parameter or 3-parameter distributions, the third or fourth value in the vector is the ζ of the distribution. ζ represents the fraction of the sample that is noncensored, or number of observed (noncensored) values divided by the sample size. The ζ represents censoring on the right, that is there are unknown observations above a threshold or the largest observed sample. Consultation of `parrevgum` or `pargpaRC` should elucidate the censoring discussion.

Value

An R `list` is returned. This list should contain at least the following items, but some distributions such as the `revgum` have extra.

- `type` — The type of distribution in three character format.
- `para` — The parameters of the distribution.
- `source` — Attribute specifying source of the parameters—“vec2par”.

Note

If the `type` is not amongst the official list given above, then the `type` given is loaded into the `type` element of the returned list and an other element `isuser = TRUE` is also added. There is no `isuser` created if the distribution is supported by **lmomco**. This is an attempt to given some level of flexibility so that others can create their own distributions or conduct research on derivative code from **lmomco**.

Author(s)

W.H. Asquith

See Also

`lmom2par`, `par2vec`

Examples

```
para <- vec2par(c(12,123,0.5),'gev')
Q <- quagev(0.5,para)

my.custom <- vec2par(c(2,2), type='myowndist') # Think about making your own
```

---

vec2pwm *Convert a Vector of Probability-Weighted Moments to a Probability-Weighted Moments Object*

Description

This function converts a vector of probability-weighted moments (PWM) to a PWM object of **lmomco**. The object is an R `list`. This function is intended to facilitate the use of PWM that the user might have from other sources. The first five PWMs are supported (β_0, β_1, β_2, β_3, β_4) if `as.list=FALSE` otherwise the β_r are unlimited.

Usage

```
vec2pwm(vec, as.list=FALSE)
```

Arguments

`vec`	A vector of PWM values in (β_0, β_1, β_2, β_3, β_4) order.
`as.list`	A logical controlling the returned data structure.

Value

An R `list` is returned if `as.list=TRUE`.

`BETA0`	The first PWM, which is equal to the arithmetic mean.
`BETA1`	The second PWM.
`BETA2`	The third PWM.
`BETA3`	The fourth PWM.
`BETA4`	The fifth PWM.
`source`	Source of the PWMs: "vec2pwm".

Another R `list` is returned if `as.list=FALSE`.

`betas`	The PWMs.
`source`	Source of the PWMs: "vec2pwm".

Author(s)

W.H. Asquith

See Also

vec2lmom, lmom2pwm, pwm2lmom

Examples

```
pwm <- vec2pwm(c(12,123,12,12,54))
```

---

vec2TLmom *Convert a Vector of TL-moments to a TL-moment Object*

Description

This function converts a vector of trimmed L-moments (TL-moments) to a TL-moment object of **lmomco** by dispatch to vec2lmom. The object is an R `list`. This function is intended to facilitate the use of TL-moments that the user might have from other sources. The trimming on the left-tail is denoted by t and the trimming on the right-tail is denoted as s. The first five TL-moments are $\lambda_1^{(t,s)}$, $\lambda_2^{(t,s)}$, $\lambda_3^{(t,s)}$, $\lambda_4^{(t,s)}$, $\lambda_5^{(t,s)}$, $\tau^{(t,s)}$, $\tau_3^{(t,s)}$, $\tau_4^{(t,s)}$, and $\tau_5^{(t,s)}$. The function supports TL-moments and TL-moment ratios of arbitrary length. Because in typical practice the $k \geq 3$ order L-moments are dimensionless ratios ($\tau_3^{(t,s)}$, $\tau_4^{(t,s)}$, and $\tau_5^{(t,s)}$), this function computes $\lambda_3^{(t,s)}$, $\lambda_4^{(t,s)}$, $\lambda_5^{(t,s)}$ from $\lambda_2^{(t,s)}$ and the ratios. However, typical practice is not set on the use of $\lambda_2^{(t,s)}$ or $\tau^{(t,s)}$ as measure of dispersion. Therefore, this function takes an `lscale` optional logical argument—if $\lambda_2^{(t,s)}$ is provided and `lscale=TRUE`, then τ is computed by the function and if τ is provided, then $\lambda_2^{(t,s)}$ is computed by the function. The trim level of the TL-moment is required. Lastly, it might be common for $t = s$ and hence symmetrical trimming is used.

Usage

```
vec2TLmom(vec, ...)
```

Arguments

`vec`	A vector of L-moment values in $\lambda_1^{(t,s)}$, $\lambda_2^{(t,s)}$ or $\tau^{(t,s)}$, $\tau_3^{(t,s)}$, $\tau_4^{(t,s)}$, and $\tau_5^{(t,s)}$ order.
`...`	The arguments used by vec2lmom.

Value

An R `list` is returned where t represents the `trim` level.

`lambdas`	Vector of the TL-moments. First element is $\lambda_1^{(t,s)}$, second element is $\lambda_2^{(t,s)}$, and so on.
`ratios`	Vector of the L-moment ratios. Second element is $\tau^{(t,s)}$, third element is $\tau_3^{(t,s)}$ and so on.
`trim`	Level of symmetrical trimming, which should equal `NULL` if asymmetrical trimming is used.

`leftrim`	Level of trimming of the left-tail of the sample.
`rightrim`	Level of trimming of the right-tail of the sample.
`source`	An attribute identifying the computational source of the L-moments: "TLmoms".

Note

The motiviation for this function that arrange trivial arguments for `vec2lmom` is that it is uncertain how TL-moments will grow in the research community and there might someday be a needed for alternative support without having to touch `vec2lmom`. Plus there is nice function name parallelism in having a dedicated function for the TL-moments as there is for L-moments and probability-weighted moments.

Author(s)

W.H. Asquith

See Also

`TLmoms`, `vec2lmom`

Examples

```
TL <- vec2TLmom(c(12,0.6,0.34,0.20,0.05),lscale=FALSE,trim=1)
```

`vegaprecip` *Annual Maximum Precipitation Data for Vega, Texas*

Description

Annual maximum precipitation data for Vega, Texas

Usage

```
data(vegaprecip)
```

Format

An R `data.frame` with

YEAR The calendar year of the annual maxima.

DEPTH The depth of 7-day annual maxima rainfall in inches.

References

Asquith, W.H., 1998, Depth-duration frequency of precipitation for Texas: U.S. Geological Survey Water-Resources Investigations Report 98–4044, 107 p.

Examples

```
data(vegaprecip)
summary(vegaprecip)
```

---

x2pars *Estimate an Ensemble of Parameters from Three Different Methods*

Description

This function acts as a frontend to estimate an ensemble of parameters from the methods of L-moments (lmr2par), maximum likelihood (MLE, mle2par), and maximum product of spacings (MPS, mps2par). The parameters estimated by the L-moments are used as the initial parameter guesses for the subsequent calls to MLE and MPS.

Usage

```
x2pars(x, verbose=TRUE, ...)
```

Arguments

x	A vector of data values.
verbose	A logical to control a sequential message ahead of each method.
...	The additional arguments, if ever used.

Value

A list having

lmr	Parameters from method of L-moments. This is expected to be NULL if the method fails, and the NULL is tested for in pars2x.
mle	Parameters from MLE. This is expected to be NULL if the method fails, and the NULL is tested for in pars2x.
mps	Parameters from MPS. This is expected to be NULL if the method fails, and the NULL is tested for in pars2x.

Author(s)

W.H. Asquith

See Also

pars2x

Examples

```
## Not run:
# Simulate from GLO and refit it. Occasionally, the simulated data
# will result in MLE or MPS failing to converge, just a note to users.
set.seed(3237)
x <- rlmomco(126, vec2par(c(2.5, 0.7, 0.3), type="glo"))
three.para.est <- x2pars(x, type="glo")
print(three.para.est$lmr$para) # 2.5598083 0.6282518 0.1819538
print(three.para.est$mle$para) # 2.5887340 0.6340132 0.2424734
print(three.para.est$mps$para) # 2.5843058 0.6501916 0.2364034
## End(Not run)
```

---

x2xlo *Conversion of a Vector through a Left-Hand Threshold to Setup Conditional Probability Computations*

Description

This function takes a vector of numerical values and subselects the values above and those equal to or less than the `leftout` argument and assigns plotting positions based on the `a` argument (passed into the pp function) and returns a list providing helpful as well as necessary results needed for conditional probability adjustment to support for general magnitude and frequency analysis as often is needed in hydrologic applications. This function only performs very simple vector operations. The real features for conditional probability application are found in the `f2flo` and `f2flo` functions.

Usage

```
x2xlo(x, leftout=0, a=0, ghost=NULL)
```

Arguments

`x`	A vector of values.
`leftout`	The lower threshold for which to leave out. The default of zero sets up for conditional probability adjustments for values equal (or less than) zero. This argument is called "left out" so as to reinforce the idea that it is a lower threshold hold on which to "leave out" data.
`a`	The plotting position coefficient passed to pp.
`ghost`	A ghosting or shadowing variable to be dragged along and then split up according to the lower threshold. If not `NULL`, then the output also contains `ghostin` and `ghostout`. This is a useful feature say if the year of data collection is associated with `x` and the user wants a convenient way to keep the proper association with the year. This feature is only for the convenience of the user and does not represent some special adjustment to the underlying concepts. A warning is issued if the lengths of `x` and `ghost` are not the same, but the function continues proceeding.

Value

An R list is returned.

xin	The subselection of values greater than the leftout threshold.
ppin	The plotting positions of the subselected values greater than the leftout threshold. These plotting positions correspond to those data values in xin.
xout	The subselection of values less than or equal to the leftout threshold.
ppout	The plotting positions of the subselected values less than or equal to the leftout threshold. These plotting positions correspond to those data values in xout.
pp	The plotting position of the largest value left out of xin.
thres	The threshold value provided by the argument leftout.
nin	Number of values greater than the threshold.
nlo	Number of values less than or equal to the threshold.
n	Total number of values: nin + nlo.
source	The source of the parameters: "x2xlo".

Author(s)

W.H. Asquith

See Also

f2flo, flo2f, f2f, xlo2qua, par2qua2lo

Examples

```
## Not run:
set.seed(62)
Fs <- nonexceeds()
type <- "exp"; parent <- vec2par(c(0,13.4), type=type)
X <- rlmomco(100, parent); a <- 0; PP <- pp(X, a=a); Xs <- sort(X)
par <- lmom2par(lmoms(X), type=type)
plot(PP, Xs, type="n", xlim=c(0,1), ylim=c(.1,100), log="y",
     xlab="NONEXCEEDANCE PROBABILITY", ylab="RANDOM VARIATE")
points(PP, Xs, col=3, cex=2, pch=0, lwd=2)
X[X < 2.1] <- X[X < 2.1]/2 # create some low outliers
Xlo <- x2xlo(X, leftout=2.1, a=a)
parlo <- lmom2par(lmoms(Xlo$xin), type=type)
points(Xlo$ppout, Xlo$xout, pch=4, col=1)
points(Xlo$ppin, Xlo$xin,   col=4, cex=.7)
lines(Fs, qlmomco(Fs, parent), lty=2, lwd=2)
lines(Fs, qlmomco(Fs, par),    col=2, lwd=4)
lines(sort(c(Xlo$ppin,.999)),
      qlmomco(f2flo(sort(c(Xlo$ppin,.999)), pp=Xlo$pp), parlo), col=4, lwd=3)
# Notice how in the last line plotted that the proper plotting positions of the data
# greater than the threshold are passed into the f2flo() function that has the effect
# of mapping conventional nonexceedance probabilities into the conditional probability
# space. These mapped probabilities are then passed into the quantile function.
```

```
legend(.3,1, c("Simulated random variates",
                "Values to 'leave' (condition) out because x/2 (low outliers)",
                "Values to 'leave' in", "Exponential parent",
                "Exponential fitted to whole data set",
                "Exponential fitted to left-in values"), bty="n", cex=.75,
                pch   =c(0,4,1,NA,NA,NA), col=c(3,1,4,1,2,4), pt.lwd=c(2,1,1,1),
                pt.cex=c(2,1,0.7,1),      lwd=c(0,0,0,2,2,3),    lty=c(0,0,0,2,1,1))

## End(Not run)
```

---

xlo2qua — *Conversion of a Vector through a Left-Hand Threshold to Setup Conditional Probability Computations*

Description

This function takes a vector of nonexceedance probabilities, a parameter object, and the object of the conditional probabability structure and computes the quantiles. This function only performs very simple vector operations. The real features for conditional probability application are found in the `x2xlo` and `f2flo` functions.

Usage

```
xlo2qua(f, para=NULL, xlo=NULL, augasNA=FALSE, sort=FALSE, fillthres=TRUE,
           retrans=function(x) x, paracheck=TRUE, ...)
```

Arguments

`f`	Nonexceedance probability ($0 \leq F \leq 1$). Be aware, these are sorted internally.
`para`	Parameters from `parpe3` or `vec2par`.
`xlo`	Mandatory result from `x2xlo` containing the content needed for internal call to `f2flo` and then vector augmentation with the threshold within the `xlo`. If this is left as `NULL`, then the function simply calls the quantile function for the parameters in `para`.
`augasNA`	A logical to switch out the threshold of `xlo` for `NA`.
`sort`	A logical whose default adheres to long-term assembly of **lmomco** behavior with working with conditional trunction. Setting this to true, triggers hand assembly of the the unsorted returned quantiles with support for `NA` and more flexibility than `x2xlo` as originally designed. If sort is true, then the `f` is permitted to contain `NA` values.
`fillthres`	A logical to trigger `qua[qua <= xlo$thres] <- xlo$thres` or replacement of computed values less than the threshold with the threshold. The argument `augasNA` is consulted after `fillthres`.
`retrans`	A retransformation function for the quantiles after they are computed according to the `para`.
`paracheck`	A logical controlling whether the parameters are checked for validity.
`...`	Additional arguments, if needed, dispatched to `par2qua`.

Value

A vector of quantiles (sorted) for the nonexceedance probabilities and padding as needed to the threshold within the `xlo` object.

Author(s)

W.H. Asquith

See Also

`f2flo`, `flo2f`, `f2f`, `x2xlo`

Examples

```
# This seed produces a quantile below the threshold for the FF nonexceedances and
# triggers the qua[qua <= xlo$thres] <- xlo$thres inside xlo2qua().

set.seed(2)
FF  <- nonexceeds();  LOT <- 0 # low-outlier threshold

XX  <- 10^rlmomco(20, vec2par(c(3, 0.7, 0.3), type="pe3"))
XX  <- c(rep(LOT, 5), XX)
# Pack the LOT values to the simulation, note that in most practical applications
# involving logarithms, that zeros rather than LOTs would be more apt, but this
# demonstration is useful because of the qua[qua <= xlo$thres] (see sources).
# Now, make the xlo object using the LOT as the threshold---the out of sample flag.

xlo <- x2xlo(XX, leftout=LOT)
pe3 <- parpe3( lmoms( log10(xlo$xin) ) )
# Fit the PE3 to the log10 of those values remaining in the sample.

QQ  <- xlo2qua(FF, para=pe3, xlo=xlo, retrans=function(x) 10^x)
# This line does all the work. Saves about four lines of code and streamlines
# logic when making frequency curves from the parameters and the xlo.

# Demonstrate this frequency curve to the observational sample.
plot(FF, QQ, log="y", type="l", col=grey(0.8))
points(pp(XX), sort(XX), col="red")

# Notice that with logic here and different seeds that XX could originally have
# values less than the threshold, so one would not have the lower tail all
# plotting along the threshold and a user might want to make other decisions.
QZ  <- xlo2qua(FF, para=pe3, xlo=xlo, augasNA=TRUE, retrans=function(x) 10^x)
lines(FF, QZ, col="blue")
# See how the QZ does not plot until about FF=0.2 because of the augmentation
# as NA (augasNA) being set true.

## Not run:
# Needs library(copBasic); library(MGBT) # too
Asite <- "08148500"; Bsite <- "08150000"; dtype <- "gev"
AB    <- MGBT::jointPeaks(Asite, Bsite) # tables of the peaks and pairwise peaks
A     <- AB$Asite_no[AB$Asite_no$appearsSystematic == TRUE, ] # only record when
```

```
B     <- AB$Bsite_no[AB$Bsite_no$appearsSystematic == TRUE, ] # monitoring occurring
QA    <- A$peak_va; Alot <- 0 # cfs (just protection from zeros, more sophisticated)
QB    <- B$peak_va; Blot <- 0 # cfs (work might be needed for better thresholds)
Alo   <- x2xlo(QA, leftout=Alot) # A xlo object
Blo   <- x2xlo(QB, leftout=Blot) # B xlo object
Apara <- lmr2par(log10(Alo$xin), type=dtype) # note log10
Bpara <- lmr2par(log10(Blo$xin), type=dtype) # note log10
Aupr  <- 10^supdist(Apara)$support[2]
Bupr  <- 10^supdist(Bpara)$support[2]
UVsS  <- AB$AB[, c("U", "V")] # isolate paired empirical probabilities
rhoS  <- copBasic::rhoCOP(as.sample=TRUE,     para=UVsS) # Spearman rho
infS  <- copBasic::LzCOPpermsym(cop=EMPIRcop, para=UVsS, as.vec=TRUE)
# a vector of permutation (variable exchangability) distances

tparf <- function(par) { c(log(par[1] -1), log(par[2]),  # transform for optimization
                   qnorm(punif(par[3],  min=-1, max=1))) }
rparf <- function(par) { c(exp(par[1])+1,  exp(par[2]),  # re-transformation to copula
                   qunif(pnorm(par[3]), min=-1, max=1)) }

ofunc <- function(par) { # objective function
  mypara <- rparf(par)   # re-transform to copula space
  mypara <- list(cop=GHcop, para=mypara[1:2], breve=mypara[3]) # asymmetry by breveCOP()
  rhoT   <- copBasic::rhoCOP(cop=breveCOP, para=mypara) # Spearman rho
  infT   <- copBasic::LzCOPpermsym(cop=breveCOP, para=mypara, as.vec=TRUE)
  err    <- mean( (infT - infS)^2 ) + (rhoT - rhoS)^2 # sum of square-like errors
  return(err)
}
init.par <- tparf(c(2, 1, 0)); rt <- NULL # init parameters and root
try( rt <- optim(init.par, ofunc) )
cpara <- rparf(rt$par) # re-transformation
cpara <- list(cop=GHcop, para=cpara[1:2], breve=cpara[3]) # copula parameters for
# an double-parameter Gumbel copula with permutation asymmetry via the breve.

ns <- 1000 # years of bivariate simulation
UVsim <- copBasic::rCOP(ns, cop=breveCOP, para=cpara, resamv01=TRUE) # simulation
AS <- xlo2qua(UVsim[,1], para=Apara, xlo=Alo, sort=FALSE,  # **** see xlo2qua() use
                         retrans=function(x) 10^x, paracheck=FALSE)
BS <- xlo2qua(UVsim[,2], para=Bpara, xlo=Blo, sort=FALSE,  # **** see xlo2qua() use
                         retrans=function(x) 10^x, paracheck=FALSE)

FF  <- seq(0.001, 0.999, by=0.001); qFF <- qnorm(FF) # probabilities for marginal curve
AF <- xlo2qua(FF, para=Apara, xlo=Alo, sort=FALSE,         # **** see xlo2qua() use
                  retrans=function(x) 10^(x), paracheck=FALSE)
BF <- xlo2qua(FF, para=Bpara, xlo=Blo, sort=FALSE,         # **** see xlo2qua() use
                  retrans=function(x) 10^(x), paracheck=FALSE)
# There might be a small region in the lower-left corner that is not attainable by the
# use of the thresholding. Let us add the complexity to the example by working out
# about the minimum points on the curves w/o more sophisticated computation.
mx <- min(c(AS, AF), na.rm=TRUE); my <- min(c(BS, BF), na.rm=TRUE)
# The use of the mx and my help us with a polygon to come, but also help us to set
# some axis limits that are especially suitable to see the entire situation of the
# simulation canvasing [0,1]^2 but the quantiles through the univariate margins might
# have truncation because of handling of the lower-tail by the threshold.
```

```
# finally plot the bivariate relation
plot(AB$AB$Apeak_va, AB$AB$Bpeak_va, log="xy", type="n",
     xlim=range(c(mx, QA, AS, ifelse(is.finite(Aupr), Aupr, NA)), na.rm=TRUE),
     ylim=range(c(my, QB, BS, ifelse(is.finite(Bupr), Bupr, NA)), na.rm=TRUE),
     xlab=paste0("Paired water-year peak streamflow for streamgage ", Asite),
     ylab=paste0("Paired water-year peak streamflow for streamgage ", Bsite))
cr <- 10^par()$usr[c(1, 3)]             # finish forming the region in the lower-left
px <- c(cr[1], mx, mx, cr[1], cr[1])    # corner that is truncated away; we do this
py <- c(cr[2], cr[2], my, my, cr[2])    # this because log10() used and in practical
polygon(px, py, col="wheat", border=NA) # applications at best zeros might be data
abline(v=mx, lty=2, lwd=0.8); abline(h=my, lty=2, lwd=0.8) # further demarcation
if( is.finite(Aupr) ) abline(v=Aupr, lty=2, lwd=1.5, col="purple") # upper limit
if( is.finite(Bupr) ) abline(h=Bupr, lty=2, lwd=1.5, col="purple") # upper limit
points(AS, BS, pch=21, col="red", bg="white") # now plot the simulations
points(AB$AB$Apeak_va, AB$AB$Bpeak_va, cex=AB$AB$cex, # now plot the observed data that
       col="black", bg=grey(AB$AB$cex/2), pch=21) # defined the parameter estimation of
legend("bottomright",                            # the copula then draw a legend.
     c("Paired streamflow (fill lightens/size increases as days apart increases)",
       paste0(ns, " years simulated by copula and GEV margins")), bty="o", cex=0.8,
       pch=c(21,21), col=c("black","red"), pt.cex=c(1.3,1), pt.bg=c(grey(0.7),"white"))

ST <- round(1/(1-kfuncCOP(0.99, cop=breveCOP, para=cpara)), digits=0)
message("Super-critical return period for ",
               "primary return period of 100 years is ", ST, " years.")

#  move on to showing the univariate margins by parametric fit with left-truncation
plot(qnorm(pp(QA)), sort(QA), log="y", pch=21, bg="white", main=Asite,
     ylim=range(c(QA, AF, Aupr), na.rm=TRUE),
     xlab="Standard normal variate", ylab="Peak streamflow, in cfs")
abline(h=Aupr, lty=2, lwd=1.5, col="purple")
lines(qFF, AF, lwd=3, col="seagreen")
legend("bottomright",
     c(paste0("Marginal distribution by ", toupper(dtype)),
       "Upper bounds of fitted distribution",
       "Systematic peaks by Weibull plotting position"), bty="o", seg.len=3,
       pch=c(NA,NA,21), col=c("seagreen","purple","black"), bg="white", cex=0.8,
       lty=c(1, 2, NA), lwd=c(3, 1.5, NA), pt.bg=c(NA, NA, "white"))

plot(qnorm(pp(QB)), sort(QB), log="y", pch=21, bg="white", main=Bsite,
     ylim=range(c(QB, BF, Bupr), na.rm=TRUE),
     xlab="Standard normal variate", ylab="Peak streamflow, in cfs")
abline(h=Bupr, lty=2, lwd=1.5, col="purple")
lines(qFF, BF, lwd=3, col="seagreen")
legend("bottomright",
     c(paste0("Marginal distribution by ", toupper(dtype)),
       "Upper bounds of fitted distribution",
       "Systematic peaks by Weibull plotting position"), bty="o", seg.len=3,
       pch=c(NA,NA,21), col=c("seagreen","purple","black"), bg="white", cex=0.8,
       lty=c(1, 2, NA), lwd=c(3, 1.5, NA), pt.bg=c(NA, NA, "white")) #
## End(Not run)
```

`z.par2cdf` *Blipping Cumulative Distribution Functions*

Description

This function acts as a front end or dispatcher to the distribution-specific cumulative distribution functions but also provides for blipping according to

$$F(x) = 0$$

for $x \leq z$ and

$$F(x) = p + (1-p)G(x)$$

for $x > z$ where z is a threshold value. The z is not tracked as part of the parameter object. This might arguably be a design flaw, but the function will do its best to test whether the z given is compatable (but not necessarily equal to $\hat{x} = x(0)$) with the quantile function $x(F)$ (`z.par2qua`). Lastly, please refer to the finiteness check in the Examples to see how one might accommodate $-\infty$ for $F = 0$ on a standard normal variate plot.

A recommended practice when working with this function is the insertion of the x value at $F = p$. Analogous practice is suggested for `z.par2qua` (see that documentation).

Usage

```
z.par2cdf(x, p, para, z=0, ...)
```

Arguments

`x`	A real value vector.
`p`	Nonexceedance probability of the `z` value. This probability could simply be the portion of record having zero values if `z=0`.
`para`	The parameters from `lmom2par` or `vec2par`.
`z`	Threshold value.
`...`	The additional arguments are passed to the cumulative distribution function such as `paracheck=FALSE` for the Generalized Lambda distribution (`cdfgld`).

Value

Nonexceedance probability ($0 \leq F \leq 1$) for `x`.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

z.par2qua, par2cdf

Examples

```
set.seed(21)
the.gpa   <- vec2par(c(100,1000,0.1),type='gpa')
fake.data <- rlmomco(30,the.gpa) # simulate some data
fake.data <- sort(c(fake.data,rep(0,10))) # add some zero observations
# going to tick to the inside and title right axis as well, so change some
# plotting parameters
par(mgp=c(3,0.5,0), mar=c(5,4,4,3))
# next compute the parameters for the positive data
gpa.all <- pargpa(lmoms(fake.data))
gpa.nzo <- pargpa(lmoms(fake.data[fake.data > 0]))
n   <- length(fake.data) # sample size
p   <- length(fake.data[fake.data == 0])/n # est. prob of zero value
F   <- nonexceeds(sig6=TRUE); F <- sort(c(F,p)); qF <- qnorm(F)
# The following x vector obviously contains zero, so no need to insert it.
x   <- seq(-100, max(fake.data)) # absurd for x<0, but testing implementation
PP  <- pp(fake.data) # compute plotting positions of sim. sample
plot(fake.data, qnorm(PP), xlim=c(0,4000), yaxt="n", ylab="") # plot the sample
add.lmomco.axis(las=2, tcl=0.5, side=2, twoside=FALSE,
                                        side.type="NPP", otherside.type="SNV")
lines(quagpa(F,gpa.all), qF) # the parent (without zeros)
cdf <- qnorm(z.par2cdf(x,p,gpa.nzo))
cdf[! is.finite(cdf)] <- min(fake.data,qnorm(PP)) # See above documentation
lines(x, cdf,lwd=3) # fitted model with zero conditional
# now repeat the above code over and over again and watch the results
par(mgp=c(3,1,0), mar=c(5,4,4,2)+0.1) # restore defaults
```

---

z.par2qua *Blipping Quantile Functions*

Description

This function acts as a front end or dispatcher to the distribution-specific quantile functions but also provides for blipping for zero (or other) threshold according to

$$x(F) = 0$$

for $0 \le F \le p$ and

$$x_G\left(\frac{F-p}{1-p}\right)$$

for $F > p$. This function is generalized for $z \neq 0$. The z is not tracked as part of the parameter object. This might arguably be a design flaw, but the function will do its best to test whether the z given is compatable (but not necessarily equal to $\hat{x} = x(0)$) with the quantile function $x(F)$.

A recommended practice when working with this function when F values are generated for various purposes, such as for graphics, then the value of p should be inserted into the vector, and the vector obviously sorted (see the line using the nonexceeds function). This should be considered as well when z.par2cdf is used but with the insertion of the x value at $F = p$.

Usage

```
z.par2qua(f, p, para, z=0, ...)
```

Arguments

f	Nonexceedance probabilities ($0 \leq F \leq 1$).
p	Nonexceedance probability of z value.
para	The parameters from lmom2par or vec2par.
z	Threshold value.
...	The additional arguments are passed to the quantile function such as paracheck = FALSE for the Generalized Lambda distribution (quagld).

Value

Quantile value for f.

Author(s)

W.H. Asquith

References

Asquith, W.H., 2011, Distributional analysis with L-moment statistics using the R environment for statistical computing: Createspace Independent Publishing Platform, ISBN 978–146350841–8.

See Also

z.par2cdf, par2qua

Examples

```
# define the real parent (or close)
the.gpa   <- vec2par(c(100,1000,0.1),type='gpa')
fake.data <- rlmomco(30,the.gpa) # simulate some data
fake.data <- sort(c(fake.data, rep(0,10))) # add some zero observations

par(mgp=c(3,0.5,0)) # going to tick to the inside, change some parameters
# next compute the parameters for the positive data
gpa.all <- pargpa(lmoms(fake.data))
gpa.nzo <- pargpa(lmoms(fake.data[fake.data > 0]))
n   <- length(fake.data) # sample size
p   <- length(fake.data[fake.data == 0])/n # est. prob of zero value
F   <- nonexceeds(sig6=TRUE); F <- sort(c(F,p)); qF <- qnorm(F)
PP  <- pp(fake.data) # compute plotting positions of sim. sample
plot(qnorm(PP), fake.data, ylim=c(0,4000), xaxt="n", xlab="") # plot the sample
add.lmomco.axis(las=2, tcl=0.5, twoside=TRUE, side.type="SNV", otherside.type="NA")
```

```
lines(qF,quagpa(F,gpa.all)) # the parent (without zeros)
lines(qF,z.par2qua(F,p,gpa.nzo),lwd=3) # fitted model with zero conditional
par(mgp=c(3,1,0)) # restore defaults
```

Index