

# Package ‘foodquotient’

May 8, 2026

**Type** Package

**Title** Food Quotient and Nutrient Analysis for HSFFQ

**Version** 0.1.1

**Description** Aids in analysing data from a food frequency questionnaire known as the Harvard Service Food Frequency Questionnaire (HSFFQ).

Functions from this package use answers from the HSFFQ to generate estimates of daily consumed micronutrients, calories, macronutrients on an individual level.

The package also calculates food quotients on individual and group levels.

Foodquotient calculation is an often tedious step in the calculation of total human energy expenditure (TEE) using the doubly labeled water method, which is the gold standard for measuring TEE.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Depends** R (>= 2.10)

**URL** <<https://naldc.nal.usda.gov/catalog/32818>>

**NeedsCompilation** no

**Author** Kate Pogue [aut, cre]

**Maintainer** Kate Pogue <katepogue222@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-08-22 23:50:02 UTC

## Contents

age_freq . . . . .	2
fq . . . . .	5
freq . . . . .	6
grams . . . . .	9
hsffq . . . . .	10

macquotient . . . . .	10
macros . . . . .	11
micros . . . . .	12
nutrients . . . . .	13
quotient . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

age_freq	<i>Frequency Factors for American Children with Age of Participant</i>
----------	--

---

### Description

A small set of food frequency questionnaire data including 32 children living in the United States. f1:f85 represents the frequency with which participants consumed 85 respective foods. Numbers 1-9 correspond to the following: 1: never 2: 1-3 times per month 3: once per week 4: 2-4 times per week 5: 5-6 times per week 6: 1 per day 7: 2-3 times per day 8: 4-5 times per day 9: 6 times per day

### Usage

age\_freq

### Format

## 'age\_freq' A data frame with 32 rows and 86 columns:

- a** age of participant
- f1** milk frequency factor
- f2** hot chocolate frequency factor
- f3** cheese frequency factor
- f4** yogurt frequency frequency factor
- f5** ice cream frequency frequency factor
- f6** pudding frequency factor
- f7** orange juice frequency factor
- f8** other juice frequency factor
- f9** fruit drink frequency factor
- f10** banana frequency factor
- f11** peaches frequency factor
- f12** mixed fruit frequency factor
- f13** orange frequency factor
- f14** apple and pear frequency factor
- f15** applesauce frequency factor

- f16** grapes frequency factor
- f17** strawberries frequency factor
- f18** melon frequency factor
- f19** pineapple frequency factor
- f20** raisins frequency factor
- f21** corn frequency factor
- f22** peas frequency factor
- f23** tomato frequency factor
- f24** peppers frequency factor
- f25** carrot frequency factor
- f26** broccoli frequency factor
- f27** green beans frequency factor
- f28** spinach frequency factor
- f29** greens frequency factor
- f30** mixed vegetable frequency factor
- f31** squash frequency factor
- f32** zucchini frequency factor
- f33** fried potatoes frequency factor
- f34** other potatoes frequency factor
- f35** sweet potatoes frequency factor
- f36** cabbage frequency factor
- f37** lettuce frequency factor
- f38** mayonnaise frequency factor
- f39** chips frequency factor
- f40** popcorn frequency factor
- f41** crackers frequency factor
- f42** nuts frequency factor
- f43** cookies frequency factor
- f44** cake frequency factor
- f45** pie frequency factor
- f46** jello frequency factor
- f47** chocolate frequency factor
- f48** candy frequency factor
- f49** coffee frequency factor
- f50** soda frequency factor
- f51** sugarfree soda frequency factor
- f52** beans frequency factor

- f53** rice frequency factor
- f54** pasta frequency factor
- f55** pizza frequency factor
- f56** tacos frequency factor
- f57** mac and cheese frequency factor
- f58** hot dogs frequency factor
- f59** sausage frequency factor
- f60** hamburger frequency factor
- f61** tuna frequency factor
- f62** fried fish frequency factor
- f63** other fish frequency factor
- f64** cold cuts frequency factor
- f65** chicken nuggets frequency factor
- f66** other chicken frequency factor
- f67** pork frequency factor
- f68** beef frequency factor
- f69** organ meats frequency factor
- f70** peanut butter frequency factor
- f71** bread frequency factor
- f72** butter frequency factor
- f73** margarine frequency factor
- f74** vegetabele soup frequency factor
- f75** soup frequency factor
- f76** tortilla frequency factor
- f77** eggs frequency factor
- f78** bacon frequency factor
- f79** hot cereal frequency factor
- f80** cold cereal frequency factor
- f81** donuts frequency factor
- f82** muffins frequency factor
- f83** pancake frequency factor
- f84** bagel frequency factor
- f85** biscuit frequency factor

**Source**

<Baylor Human Evolutionary Biology and Health Lab>

---

fq	<i>Frequency Factor</i>
----	-------------------------

---

### Description

The Frequency Factor function converts values 1-9, representing different frequency factor responses from the hsfq, to average daily servings consumed for that individual.

### Usage

```
fq(f)
```

### Arguments

f                    1-9, representing different frequency factor responses from the hsfq. These can be in a dataframe, vector, or just single values

### Value

a dataframe, vector, or single value of the same dimension as the input, with each position holding the average daily servings consumed for each food (columns) for each individual(rows).

### Examples

```
test <- c(1, 5, 7, 3, 9, 2, 4, 3, 6, 8)
fq(test)

rquestionnaire <- function(n, n_food_questions = 85) {
  mat <- matrix(
    sample(1:9, n_food_questions*n, replace = TRUE),
    nrow = n, ncol = n_food_questions
  )
  df <- data.frame( age = round(runif(n, 2, 11), digits = 1) )
  cbind(df, as.data.frame(mat))
}
df <- rquestionnaire(6)

fq(df)
```

---

 freq

*Frequency Factors for American Children*


---

### Description

A small set of data including 32 children living in the United States. f1:f85 represents the frequency with which participants consumed 85 respective foods. Numbers 1-9 correspond to the following: 1: never 2: 1-3 times per month 3: once per week 4: 2-4 times per week 5: 5-6 times per week 6: 1 per day 7: 2-3 times per day 8: 4-5 times per day 9: 6 times per day

### Usage

freq

### Format

## 'freq' A data frame with 32 rows and 85 columns:

- f1** milk frequency factor
- f2** hot chocolate frequency factor
- f3** cheese frequency factor
- f4** yogurt frequency frequency factor
- f5** ice cream frequency frequency factor
- f6** pudding frequency factor
- f7** orange juice frequency factor
- f8** other juice frequency factor
- f9** fruit drink frequency factor
- f10** banana frequency factor
- f11** peaches frequency factor
- f12** mixed fruit frequency factor
- f13** orange frequency factor
- f14** apple and pear frequency factor
- f15** applesauce frequency factor
- f16** grapes frequency factor
- f17** strawberries frequency factor
- f18** melon frequency factor
- f19** pineapple frequency factor
- f20** raisins frequency factor
- f21** corn frequency factor
- f22** peas frequency factor

- f23** tomato frequency factor
- f24** peppers frequency factor
- f25** carrot frequency factor
- f26** broccoli frequency factor
- f27** green beans frequency factor
- f28** spinach frequency factor
- f29** greens frequency factor
- f30** mixed vegetable frequency factor
- f31** squash frequency factor
- f32** zucchini frequency factor
- f33** fried potatoes frequency factor
- f34** other potatoes frequency factor
- f35** sweet potatoes frequency factor
- f36** cabbage frequency factor
- f37** lettuce frequency factor
- f38** mayonnaise frequency factor
- f39** chips frequency factor
- f40** popcorn frequency factor
- f41** crackers frequency factor
- f42** nuts frequency factor
- f43** cookies frequency factor
- f44** cake frequency factor
- f45** pie frequency factor
- f46** jello frequency factor
- f47** chocolate frequency factor
- f48** candy frequency factor
- f49** coffee frequency factor
- f50** soda frequency factor
- f51** sugarfree soda frequency factor
- f52** beans frequency factor
- f53** rice frequency factor
- f54** pasta frequency factor
- f55** pizza frequency factor
- f56** tacos frequency factor
- f57** mac and cheese frequency factor
- f58** hot dogs frequency factor
- f59** sausage frequency factor

- f60** hamburger frequency factor
- f61** tuna frequency factor
- f62** fried fish frequency factor
- f63** other fish frequency factor
- f64** cold cuts frequency factor
- f65** chicken nuggets frequency factor
- f66** other chicken frequency factor
- f67** pork frequency factor
- f68** beef frequency factor
- f69** organ meats frequency factor
- f70** peanut butter frequency factor
- f71** bread frequency factor
- f72** butter frequency factor
- f73** margarine frequency factor
- f74** vegetable soup frequency factor
- f75** soup frequency factor
- f76** tortilla frequency factor
- f77** eggs frequency factor
- f78** bacon frequency factor
- f79** hot cereal frequency factor
- f80** cold cereal frequency factor
- f81** donuts frequency factor
- f82** muffins frequency factor
- f83** pancake frequency factor
- f84** bagel frequency factor
- f85** biscuit frequency factor

**Source**

<Baylor Human Evolutionary Biology Lab>

---

grams

*Grams*

---

### Description

The grams function takes the age of a participant and their responses on the hsfq to generate an estimate of the participant's total daily grams consumed for each food.

### Usage

```
grams(row)
```

### Arguments

row                    A numeric vector with components 'age', representing the age of the participant, and 'f1' to 'f85', representing different frequency factor responses from the hsfq.

### Value

A numeric vector of length 85, representing the estimated total daily grams of each food consumed for the participant.

### Examples

```
random_integers <- sample(1:9, 85, replace=TRUE)
vec <- c(6.2, random_integers)
grams(vec)

rquestionnaire <- function(n, n_food_questions = 85) {
  mat <- matrix(
    sample(1:9, n_food_questions*n, replace = TRUE),
    nrow = n, ncol = n_food_questions
  )
  df <- data.frame( age = round(runif(n, 2, 11), digits = 1) )
  cbind(df, as.data.frame(mat))
}
df <- rquestionnaire(2)

df_results <- data.frame()
for (i in 1:nrow(df)) {
  result <- grams(df[i,])
  df_results <- rbind(df_results, result)
}
```

---

`hsffq`*Harvard Food Frequency Questionnaire Nutrition Information*

---

**Description**

This dataframe is used internally by the functions of `foodquotient` and includes portion size information by age for each of the 85 foods included in the HSFFQ from the HSFFQ user's manual. Additionally, nutrient information is included for each of the 85 foods, pulled from the USDA's public search tool.

**Usage**`hsffq()`**Value**

A portion size and nutrient information reference data frame.

---

`macquotient`*Food Quotient Based on Macronutrients*

---

**Description**

The `macquotient` function calculates a food quotient for a participant based on average daily protein, carbs, and fat consumed for an individual or a group. In contrast to the `quotient` function, `macquotient` is able to generate reliable average food quotients for a group of people rather than only individual level. Group level estimates are recommended in some studies to control for response bias.

**Usage**`macquotient(row)`**Arguments**

`row` contains three components. `p` average daily grams of protein consumed `f` average daily grams of fat consumed `c/` average daily grams of carbohydrates consumed

**Value**

one value per participant will be returned, representing the food quotient for the individual

**Examples**

```
vec <- c(34.5,43, 212.4)
macquotient(vec)

vec1 <- c(34.5,43, 212.4)
vec2 <- c(40.1,52, 240)
df <- rbind(vec1, vec2)

df_results <- data.frame()
for (i in 1:nrow(df)) {
  result <- macquotient(df[i,])
  df_results <- rbind(df_results, result)
}
```

---

 macros

*Macronutrients*


---

**Description**

The *Macronutrients* function takes the age of a participant and their responses on the hsfq to generate estimates of the participant's total daily protein, carbohydrate, and fat consumed for each food.

**Usage**

```
macros(row)
```

**Arguments**

row	vector with 86 entries consisting of 2 components f1:f85 1-9, representing different frequency factor responses from the hsfq. These will be stored in columns 2-86 in the row you plug in A value representing participant's age. This will be stored in column 1 of the input row
-----	---

**Value**

the row or dataframe returned will have 3 entries, representing total daily amounts of protein, carbohydrates, and fat for each participant

**Examples**

```
random_integers <- sample(1:9, 85, replace=TRUE)
vec <- c(6.2, random_integers)
grams(vec)

rquestionnaire <- function(n, n_food_questions = 85) {
  mat <- matrix(
```

```

    sample(1:9, n_food_questions*n, replace = TRUE),
    nrow = n, ncol = n_food_questions
  )
  df <- data.frame( age = round(runif(n, 2, 11), digits = 1) )
  cbind(df, as.data.frame(mat))
}
df <- rquestionnaire(3)

df_results <- data.frame()
for (i in 1:nrow(df)) {
  result <- macros(df[i,])
  df_results <- rbind(df_results, result)
}

```

---

micros

*Micronutrients*


---

### Description

The *Micronutrients* function takes the age of a participant and their responses on the *hsffq* to generate an estimate of the participant's total daily micronutrients consumed for each food.

### Usage

```
micros(row)
```

### Arguments

row	contains two components. f1:f85 1-9, representing different frequency factor responses from the <i>hsffq</i> . These will be stored in columns 2-86 in the row you plug in . A value representing participant's age. This will be stored in column 1 of the input row
-----	---

### Value

the row or dataframe returned will have 7 entries, representing total daily amounts of 7 micronutrients for each participant

### Examples

```

random_integers <- sample(1:8, 85, replace=TRUE)
vec <- c(6.2, random_integers)
micros(vec)

rquestionnaire <- function(n, n_food_questions = 85) {
  mat <- matrix(
    sample(1:9, n_food_questions*n, replace = TRUE),
    nrow = n, ncol = n_food_questions
  )
}

```

```

)
df <- data.frame( age = round(runif(n, 2, 11), digits = 1) )
cbind(df, as.data.frame(mat))
}
df <- questionnaire(4)

df_results <- data.frame()
for (i in 1:nrow(df)) {
  result <- micros(df[i,])
  df_results <- rbind(df_results, result)
}

```

---

nutrients

*Nutrients*


---

### Description

The Nutrients function takes the age of a participant and their responses on the hsfq to generate an estimate of the participant's total daily micronutrients, macronutrients, and calories consumed for each food

### Usage

```
nutrients(row)
```

### Arguments

row	/contains two components. f1:f85 1-9, representing different frequency factor responses from the hsfq. These will be stored in columns 2-86 in the row you plug in A value representing participant's age. This will be stored in column 1 of the input row
-----	---

### Value

the row or dataframe returned will have 11 entries, representing total daily amounts of 7 micronutrients, 3 macronutrients, and calories for each participant. These columns will be labeled

### Examples

```

random_integers <- sample(1:8, 85, replace=TRUE)
vec <- c(6.2, random_integers)
nutrients(vec)

questionnaire <- function(n, n_food_questions = 85) {
  mat <- matrix(
    sample(1:9, n_food_questions*n, replace = TRUE),
    nrow = n, ncol = n_food_questions
  )
}

```

```

)
df <- data.frame( age = round(runif(n, 2, 11), digits = 1) )
cbind(df, as.data.frame(mat))
}
df <- rquestionnaire(5)

df_results <- data.frame()
for (i in 1:nrow(df)) {
  result <- nutrients(df[i,])
  df_results <- rbind(df_results, result)
}

```

---

quotient

*Food quotient based on hsffq results*


---

### Description

The quotient function calculates individual level food quotients based on the individual's answers to the hsffq. This function is only recommended to calculate at the individual level.

### Usage

```
quotient(row)
```

### Arguments

row	contains two components. f1:f85 1-9, representing different frequency factor responses from the hsffq. These will be stored in columns 2-86 in the row you plug in A value representing participant's age. This will be stored in column 1 of the input row/
-----	--

### Value

one value per participant will be returned, representing the food quotient for the individual

### Examples

```

random_integers <- sample(1:8, 85, replace=TRUE)
vec <- c(6.2, random_integers)
quotient(vec)

rquestionnaire <- function(n, n_food_questions = 85) {
  mat <- matrix(
    sample(1:9, n_food_questions*n, replace = TRUE),
    nrow = n, ncol = n_food_questions
  )
  df <- data.frame( age = round(runif(n, 2, 11), digits = 1) )

```

```
      cbind(df, as.data.frame(mat))
    }
df <- rquestionnaire(6)

df_results <- data.frame()
for (i in 1:nrow(df)) {
  result <- quotient(df[i,])
  df_results <- rbind(df_results, result)
}
```

# Index

## \* datasets

age\_freq, 2

freq, 6

age\_freq, 2

fq, 5

freq, 6

grams, 9

hsffq, 10

macquotient, 10

macros, 11

micros, 12

nutrients, 13

quotient, 14