Package ‘heaping’

February 9, 2026

Type Package

Title Correction of Heaping on Individual Level

Version 0.1.0

Description Provides methods for correcting heaping (digit preference) in survey data at the individual record level. Age heaping, where respondents disproportionately report ages ending in 0 or 5, is a common phenomenon that can distort demographic analyses. Unlike traditional smoothing methods that only correct aggregated statistics, this package corrects individual values by replacing a calculated proportion of heaped observations with draws from fitted truncated distributions (log-normal, normal, or uniform). Supports 5-year and 10-year heaping patterns, single heap correction, and optional model-based adjustment to preserve covariate relationships.

License GPL (>= 2)

URL `https://github.com/matthias-da/heaping`

BugReports `https://github.com/matthias-da/heaping/issues`

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 3.5.0)

Imports fitdistrplus, EnvStats, stats

Suggests VIM, ranger, data.table, ggplot2, simPop, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Matthias Templ [aut, cre] (ORCID: <`https://orcid.org/0000-0002-8638-5276`>), Bernhard Meindl [ctb]

Maintainer Matthias Templ <`matthias.templ@gmail.com`>

Repository CRAN

Date/Publication 2026-02-09 19:20:07 UTC

Contents

`heaping-package` *heaping: Correction of Heaping on Individual Level*

Description

Provides methods for correcting heaping (digit preference) in survey data at the individual record level. Age heaping, where respondents disproportionately report ages ending in 0 or 5, is a common phenomenon that can distort demographic analyses.

Main Functions

`correctHeaps` Correct regular age heaping patterns (5-year or 10-year intervals)

`correctSingleHeap` Correct a specific single age heap

Methodology

Unlike traditional smoothing methods that only correct aggregated statistics, this package corrects individual values by replacing a calculated proportion of heaped observations with draws from fitted truncated distributions (log-normal, normal, or uniform).

The correction ratio is determined by comparing the count at each heap to the mean of neighboring ages. Observations exceeding this expected ratio are randomly selected and replaced with values drawn from truncated distributions fitted to the original data.

Model-Based Correction

An optional model-based adjustment using random forests can be applied to ensure that corrected values respect relationships with other variables in the dataset. This requires the **ranger** and **VIM** packages.

Multiple Imputation

Repeated calls to the correction functions can be used to implement multiple imputation, properly reflecting the uncertainty from the correction process.

Author(s)

Matthias Templ <matthias.templ@fhnw.ch>

References

Templ, M. (2024). Correction of heaping on individual level. *Journal TBD*.

Templ, M., Meindl, B., Kowarik, A., Alfons, A., Dupriez, O. (2017). Simulation of Synthetic Populations for Survey Data Considering Auxiliary Information. *Journal of Statistical Software*, **79**(10), 1-38. doi:10.18637/jss.v079.i10

See Also

Useful links:

- https://github.com/matthias-da/heaping
- Report bugs at https://github.com/matthias-da/heaping/issues

bachi *Bachi's Index of Age Heaping*

Description

Bachi's index involves applying the Whipple method repeatedly to determine the extent of preference for each terminal digit (0-9). It equals the sum of positive deviations from 10 percent.

Usage

```
bachi(x, ageMin = 23, ageMax = 77, weight = NULL)
```

Arguments

`x`	numeric vector of individual ages.
`ageMin`	minimum age to include (default 23).
`ageMax`	maximum age to include (default 77, adjusted to fit decades).
`weight`	optional numeric vector of sampling weights.

Details

Calculate Bachi's index to measure digit preference in age data.

The theoretical range is 0 to 90:

- 0: no digit preference (each digit represents 10
- 90: maximum heaping (all ages end in same digit)

For populations with no age heaping, each digit should appear in approximately 10

Value

A single numeric value representing Bachi's index.

Author(s)

Matthias Templ

References

Bachi, R. (1951). The tendency to round off age returns: measurement and correction. *Bulletin of the International Statistical Institute*, **33**(4), 195-222.

See Also

myers for Myers' index, whipple for Whipple's index.

Other heaping indices: coale_li(), heaping_indices(), jdanov(), kannisto(), myers(), noumbissi(), spoorenberg(), whipple()

Examples

```
# No heaping
set.seed(42)
age_uniform <- sample(23:77, 10000, replace = TRUE)
bachi(age_uniform)  # Should be close to 0

# Strong heaping on 0 and 5
age_heaped <- sample(seq(25, 75, by = 5), 5000, replace = TRUE)
bachi(age_heaped)  # Should be high
```

coale_li *Coale-Li Age Heaping Index*

Description

The Coale-Li index was developed to detect age heaping in populations with high proportions of elderly persons. It compares actual counts at specific ages to smoothed reference values using moving averages.

Usage

```
coale_li(x, digit = 0, ageMin = 60, ageMax = max(x), terms = 5, weight = NULL)
```

Arguments

`x`	numeric vector of individual ages.
`digit`	integer (0-9) specifying which terminal digit to evaluate (default 0).
`ageMin`	minimum age to include (default 60).
`ageMax`	maximum age to include (default max(x)).
`terms`	number of terms for moving average smoothing (default 5).
`weight`	optional numeric vector of sampling weights.

Details

Calculate the Coale-Li index for detecting age heaping at older ages.

The method applies double moving averages to create a smooth reference distribution, then calculates the ratio of observed to expected counts for ages ending in a specified digit.

Interpretation:

- 1.0: no preference for the digit
- >1.0: attraction to the digit (heaping)
- <1.0: avoidance of the digit

This index is particularly useful for evaluating data quality at older ages (60+) where heaping on round numbers is common.

Value

A single numeric value representing the Coale-Li index.

Author(s)

Matthias Templ

References

Coale, A. J. and Li, S. (1991). The effect of age misreporting in China on the calculation of mortality rates at very high ages. *Demography*, **28**(2), 293-301.

See Also

kannisto for Kannisto's index, jdanov for Jdanov's index.

Other heaping indices: bachi(), heaping_indices(), jdanov(), kannisto(), myers(), noumbissi(), spoorenberg(), whipple()

Examples

```
# Create age data with heaping at older ages
set.seed(42)
age <- c(sample(60:99, 5000, replace = TRUE),
         rep(seq(60, 90, by = 10), each = 200))  # Add heaping on 0s
coale_li(age, digit = 0)  # Should be > 1
coale_li(age, digit = 5)  # Should be closer to 1
```

correctHeaps *Correct Age Heaping*

Description

Age heaping can cause substantial bias in important demographic measures and thus should be corrected. This function corrects heaping at regular intervals (every 5 or 10 years) by replacing a proportion of heaped observations with draws from fitted truncated distributions.

Usage

```
correctHeaps(
  x,
  heaps = "10year",
  method = "lnorm",
  start = 0,
  fixed = NULL,
  model = NULL,
  dataModel = NULL,
  seed = NULL,
  na.action = "omit",
  verbose = FALSE,
  sd = NULL
)

correctHeaps2(
  x,
  heaps = "10year",
  method = "lnorm",
  start = 0,
  fixed = NULL,
  model = NULL,
  dataModel = NULL,
  seed = NULL,
  na.action = "omit",
  verbose = FALSE,
  sd = NULL
)
```

Arguments

`x` numeric vector of ages (typically integers).

`heaps` character string specifying the heaping pattern:

- `"5year"` heaps are assumed every 5 years (0, 5, 10, 15, ...)
- `"10year"` heaps are assumed every 10 years (0, 10, 20, ...)

Alternatively, a numeric vector specifying custom heap positions.

`method` character string specifying the distribution used for correction:

- `"lnorm"` truncated log-normal distribution (default). Parameters are estimated from the input data.
- `"norm"` truncated normal distribution. Parameters are estimated from the input data.
- `"unif"` uniform distribution within the truncation bounds.
- `"kernel"` kernel density estimation for nonparametric sampling.

`start` numeric value for the starting point of the heap sequence (default 0). Use 5 if heaps occur at 5, 15, 25, ... instead of 0, 10, 20, ... Ignored if `heaps` is a numeric vector.

`fixed` numeric vector of indices indicating observations that should not be changed. Useful for preserving known accurate values.

`model` optional formula for model-based correction. When provided, a random forest model is fit to predict age from other variables, and the correction direction is adjusted to be consistent with this prediction. Requires packages **ranger** and **VIM**.

`dataModel` data frame containing variables for the model formula. Required when `model` is specified. Missing values are imputed using k-nearest neighbors via `kNN`.

`seed` optional integer for random seed to ensure reproducibility. If `NULL` (default), no seed is set.

`na.action` character string specifying how to handle `NA` values:

- `"omit"` remove NA values before processing, then restore positions (default)
- `"fail"` stop with an error if NA values are present

`verbose` logical. If `TRUE`, return a list with corrected values and diagnostic information. If `FALSE` (default), return only the corrected vector.

`sd` optional numeric value for standard deviation when `method = "norm"`. If `NULL` (default), estimated from the data using MAD (median absolute deviation) of non-heap ages, which is robust to the heaping.

Details

Correct for age heaping at regular intervals using truncated distributions.

For method “lnorm”, a truncated log-normal distribution is fit to the whole age distribution. Then for each age heap (at 0, 5, 10, 15, ... or 0, 10, 20, ...) random numbers from a truncated log-normal distribution (with lower and upper bounds) are drawn.

The correction range depends on the heap type:

- For 5-year heaps: values are drawn from ±2 years around the heap
- For 10-year heaps: values are drawn in two groups, ±4 and ±5 years around the heap

The ratio of observations to replace is calculated by comparing the count at each heap age to the arithmetic mean of the two neighboring ages. For example, for age heap 5, the ratio is: count(age=5) / mean(count(age=4), count(age=6)).

Method “norm” uses truncated normal distributions instead. The choice between “lnorm” and “norm” depends on whether the age distribution is right-skewed (use “lnorm”) or more symmetric (use “norm”). Many distributions with heaping problems are right-skewed.

Method “unif” draws from truncated uniform distributions around the age heaps, providing a simpler baseline approach.

Method “kernel” uses kernel density estimation to sample replacement values, providing a nonparametric alternative that adapts to the local data distribution.

Repeated calls of this function mimic multiple imputation, i.e., repeating this procedure m times provides m corrected datasets that properly reflect the uncertainty from the correction process. Use the `seed` parameter to ensure reproducibility.

Value

If `verbose = FALSE`, a numeric vector of the same length as `x` with heaping corrected. If `verbose = TRUE`, a list with:

corrected the corrected numeric vector

n_changed total number of values changed

changes_by_heap named vector of changes per heap age

ratios named vector of heaping ratios per heap age

method method used

seed seed used (if any)

Author(s)

Matthias Templ, Bernhard Meindl

References

Templ, M. (2026). Correction of heaping on individual level. *Journal TBD*.

Templ, M., Meindl, B., Kowarik, A., Alfons, A., Dupriez, O. (2017). Simulation of Synthetic Populations for Survey Data Considering Auxiliary Information. *Journal of Statistical Software*, **79**(10), 1-38. doi:10.18637/jss.v079.i10

See Also

`correctSingleHeap` for correcting a single specific heap.

Other heaping correction: `correctSingleHeap()`

Examples

```
# Create artificial age data with log-normal distribution
set.seed(123)
age <- rlnorm(10000, meanlog = 2.466869, sdlog = 1.652772)
age <- round(age[age < 93])

# Artificially introduce 5-year heaping
year5 <- seq(0, max(age), 5)
age5 <- sample(c(age, age[age %in% year5]))

# Correct with reproducible results
age5_corrected <- correctHeaps(age5, heaps = "5year", method = "lnorm", seed = 42)

# Get diagnostic information
result <- correctHeaps(age5, heaps = "5year", verbose = TRUE, seed = 42)
print(result$n_changed)
print(result$ratios)

# Use kernel method for nonparametric correction
age5_kernel <- correctHeaps(age5, heaps = "5year", method = "kernel", seed = 42)

# Custom heap positions (e.g., heaping at 12, 18, 21)
custom_heaps <- c(12, 18, 21)
age_custom <- correctHeaps(age5, heaps = custom_heaps, method = "lnorm", seed = 42)
```

`correctSingleHeap` *Correct a Single Age Heap*

Description

While `correctHeaps` corrects regular heaping patterns, this function allows correction of a single specific heap value. This is useful when heaping occurs at irregular intervals or when only a particular age shows excessive heaping.

Usage

```
correctSingleHeap(
  x,
  heap,
  before = 2,
  after = 2,
  method = "lnorm",
  fixed = NULL,
  seed = NULL,
  na.action = "omit",
  verbose = FALSE,
  sd = NULL
)
```

Arguments

`x` numeric vector representing ages (typically integers).

`heap` numeric value specifying the age for which heaping should be corrected. Must be present in `x`.

`before` numeric value specifying the number of years before the heap to use as the lower bound for replacement values. Will be rounded to an integer. Default is 2.

`after` numeric value specifying the number of years after the heap to use as the upper bound for replacement values. Will be rounded to an integer. Default is 2.

`method` character string specifying the distribution used for correction:

- `"lnorm"` truncated log-normal distribution (default). Parameters are estimated from the input data.
- `"norm"` truncated normal distribution. Parameters are estimated from the input data.
- `"unif"` uniform distribution within the truncation bounds.
- `"kernel"` kernel density estimation for nonparametric sampling.

`fixed` numeric vector of indices indicating observations that should not be changed. Useful for preserving known accurate values.

`seed` optional integer for random seed to ensure reproducibility.

`na.action` character string specifying how to handle `NA` values: `"omit"` (default) or `"fail"`.

`verbose` logical. If `TRUE`, return diagnostic information.

`sd` optional numeric value for standard deviation when `method = "norm"`.

Details

Correct a specific age heap in a vector containing ages.

Value

A numeric vector of the same length as `x` with the specified heap corrected, or a list with diagnostics if `verbose = TRUE`.

Author(s)

Matthias Templ, Bernhard Meindl

See Also

correctHeaps for correcting regular heaping patterns.

Other heaping correction: correctHeaps()

Examples

```
# Create artificial age data
set.seed(123)
age <- rlnorm(10000, meanlog = 2.466869, sdlog = 1.652772)
age <- round(age[age < 93])

# Artificially introduce a heap at age 23
age23 <- c(age, rep(23, length = sum(age == 23)))

# Correct with reproducible results
age23_corrected <- correctSingleHeap(age23, heap = 23, before = 5, after = 5,
                                     method = "lnorm", seed = 42)

# Get diagnostic information
result <- correctSingleHeap(age23, heap = 23, before = 5, after = 5,
                            verbose = TRUE, seed = 42)
print(result$n_changed)
```

---

heaping_indices *Calculate All Heaping Indices*

Description

This function calculates all available heaping indices for a given age vector, providing a comprehensive assessment of data quality.

Usage

```
heaping_indices(x, weight = NULL)
```

Arguments

x	numeric vector of individual ages.
weight	optional numeric vector of sampling weights.

Details

Convenience function to calculate multiple heaping indices at once.

Value

A named list with all heaping indices:

whipple_standard Standard Whipple index (100 = no heaping)

whipple_modified Modified Whipple index (0 = no heaping)

myers Myers' blended index (0 = no heaping)

bachi Bachi's index (0 = no heaping)

spoorenberg Total Modified Whipple index (0 = no heaping)

noumbissi_0 Noumbissi's index for digit 0 (1 = no heaping)

noumbissi_5 Noumbissi's index for digit 5 (1 = no heaping)

Author(s)

Matthias Templ

See Also

Other heaping indices: `bachi()`, `coale_li()`, `jdanov()`, `kannisto()`, `myers()`, `noumbissi()`, `spoorenberg()`, `whipple()`

Examples

```
set.seed(42)
# Uniform ages (no heaping)
age_uniform <- sample(20:70, 10000, replace = TRUE)
heaping_indices(age_uniform)

# Heaped ages
age_heaped <- sample(seq(20, 70, by = 5), 5000, replace = TRUE)
heaping_indices(age_heaped)
```

`jdanov` *Jdanov's Old-Age Heaping Index*

Description

Jdanov's index is designed to detect age heaping at very old ages (typically 95+), where data quality is often poorest. It applies the Whipple principle to specific old-age values.

Usage

```
jdanov(x, Agei = c(95, 100, 105), weight = NULL)
```

Arguments

`x`	numeric vector of individual ages.
`Agei`	numeric vector of specific ages to evaluate (default c(95, 100, 105)).
`weight`	optional numeric vector of sampling weights.

Details

Calculate Jdanov's index for detecting heaping at very old ages.

The index compares counts at specified old ages to the surrounding 5-year age groups, similar to the standard Whipple approach but focused on the oldest ages where heaping is most problematic.

Interpretation:

- 100: no heaping
- >100: preference for the specified ages
- 500: maximum heaping (all ages at specified values)

Value

A single numeric value representing Jdanov's index.

Author(s)

Matthias Templ

References

Jdanov, D. A., Scholz, R. D., and Shkolnikov, V. M. (2008). Official population statistics and the Human Mortality Database estimates of populations aged 80+ in Germany and nine other European countries. *Demographic Research*, **19**, 1169-1196.

See Also

`kannisto` for Kannisto's index, `coale_li` for Coale-Li index.

Other heaping indices: `bachi()`, `coale_li()`, `heaping_indices()`, `kannisto()`, `myers()`, `noumbissi()`, `spoorenberg()`, `whipple()`

Examples

```
# Create old-age data with heaping
set.seed(42)
age <- c(sample(90:110, 2000, replace = TRUE),
         rep(c(95, 100, 105), each = 100))  # Add heaping
jdanov(age)  # Should be > 100

# No heaping
age_uniform <- sample(90:110, 2000, replace = TRUE)
jdanov(age_uniform)  # Should be close to 100
```

kannisto *Kannisto's Age Heaping Index*

Description

Kannisto's index compares the count at a specific age to a geometric mean of surrounding ages, providing a measure of heaping that is robust to exponentially declining populations at old ages.

Usage

```
kannisto(x, Agei = 90, weight = NULL)
```

Arguments

`x`	numeric vector of individual ages.
`Agei`	single age value to evaluate (default 90).
`weight`	optional numeric vector of sampling weights.

Details

Calculate Kannisto's index for detecting heaping at a specific old age.

Unlike other indices that use arithmetic means, Kannisto's index uses geometric means of neighboring ages, which is more appropriate for old-age populations where counts decline exponentially.

The index is calculated as the ratio of the count at age `Agei` to the geometric mean of counts at ages `Agei-2` through `Agei+2`.

Interpretation:

- 1.0: no heaping at the specified age
- >1.0: heaping (attraction to the age)
- <1.0: avoidance of the age

Value

A single numeric value representing Kannisto's index.

Author(s)

Matthias Templ

References

Kannisto, V. (1999). Assessing the information on age at death of old persons in national vital statistics. *Validation of Exceptional Longevity, Odense Monographs on Population Aging*, **6**, 235-249.

See Also

jdanov for Jdanov's index, coale_li for Coale-Li index.

Other heaping indices: bachi(), coale_li(), heaping_indices(), jdanov(), myers(), noumbissi(), spoorenberg(), whipple()

Examples

```
# Create old-age data with heaping at 90
set.seed(42)
age <- c(sample(85:95, 2000, replace = TRUE),
         rep(90, 200))  # Add heaping at 90
kannisto(age, Agei = 90)  # Should be > 1

# No heaping
age_uniform <- sample(85:95, 2000, replace = TRUE)
kannisto(age_uniform, Agei = 90)  # Should be close to 1
```

myers *Myers' Blended Index of Age Heaping*

Description

Myers' index measures preferences for each of the ten possible terminal digits (0-9) as a blended index. It is based on the principle that in the absence of age heaping, the aggregate population of each age ending in one of the digits 0 to 9 should represent 10 percent of the total population.

Usage

```
myers(x, ageMin = 23, ageMax = 82, weight = NULL)
```

Arguments

x	numeric vector of individual ages.
ageMin	minimum age to include (default 23).
ageMax	maximum age to include (default 82).
weight	optional numeric vector of sampling weights.

Details

Calculate Myers' blended index to measure digit preference in age data.

The index uses a blending technique that weights earlier ages more for digit preference calculation and later ages more for avoidance, creating a balanced measure across the age range.

The theoretical range is 0 to 90:

- 0: no digit preference (perfect data)
- 90: all ages reported with same terminal digit (maximum heaping)

Value

A single numeric value representing Myers' blended index.

Author(s)

Matthias Templ

References

Myers, R. J. (1940). Errors and bias in the reporting of ages in census data. *Transactions of the Actuarial Society of America*, **41**, 395-415.

Myers, R. J. (1954). Accuracy of age reporting in the 1950 United States Census. *Journal of the American Statistical Association*, **49**(268), 826-831.

See Also

`bachi` for Bachi's index, `whipple` for Whipple's index.

Other heaping indices: `bachi()`, `coale_li()`, `heaping_indices()`, `jdanov()`, `kannisto()`, `noumbissi()`, `spoorenberg()`, `whipple()`

Examples

```
# No heaping (uniform ages)
set.seed(42)
age_uniform <- sample(23:82, 10000, replace = TRUE)
myers(age_uniform)  # Should be close to 0

# Strong heaping on ages ending in 0 or 5
age_heaped <- sample(seq(25, 80, by = 5), 5000, replace = TRUE)
myers(age_heaped)  # Should be high
```

noumbissi *Noumbissi's Digit Heaping Index*

Description

Noumbissi's method improves on Whipple's method by extending its basic principle to all ten digits. It compares the count of ages ending in a specific digit to the count in 5-year age groups centered on that digit.

Usage

```
noumbissi(
  x,
  digit = 0,
  ageMin = 20 + digit,
  ageMax = ageMin + 30,
  weight = NULL
)
```

Arguments

`x`	numeric vector of individual ages.
`digit`	integer (0-9) specifying which terminal digit to evaluate (default 0).
`ageMin`	minimum age to include (default 20 + digit).
`ageMax`	maximum age to include (default ageMin + 30).
`weight`	optional numeric vector of sampling weights.

Details

Calculate Noumbissi's index for a specific terminal digit.

The index compares the number of persons reporting ages ending in a specific digit to one-fifth of the population in the 5-year age groups centered on those ages.

Interpretation:

- 1.0: no preference for the digit
- >1.0: preference (attraction) to the digit
- <1.0: avoidance of the digit

Value

A single numeric value representing Noumbissi's index for the specified digit.

Author(s)

Matthias Templ

References

Noumbissi, A. (1992). L'indice de Whipple modifie: une application aux donnees du Cameroun, de la Suede et de la Belgique. *Population*, **47**(4), 1038-1041.

See Also

`spoorenberg` for Total Modified Whipple index, `whipple` for original Whipple's index.

Other heaping indices: `bachi()`, `coale_li()`, `heaping_indices()`, `jdanov()`, `kannisto()`, `myers()`, `spoorenberg()`, `whipple()`

Examples

```
# No heaping
set.seed(42)
age_uniform <- sample(20:70, 10000, replace = TRUE)
noumbissi(age_uniform, digit = 0)  # Should be close to 1
noumbissi(age_uniform, digit = 5)  # Should be close to 1

# Heaping on digit 0
age_heap0 <- sample(seq(20, 70, by = 10), 5000, replace = TRUE)
noumbissi(age_heap0, digit = 0)  # Should be > 1
```

---

samp *Sample Data for Heaping Correction Examples*

Description

A stratified random sample of demographic and income data from a synthetic population generated using the **simPop** package based on EU-SILC data. This dataset can be used to demonstrate and test heaping correction methods.

Usage

```
samp
```

Format

A data frame with 25 variables:

db030 Household ID

hsize Household size

age Age in years

rb090 Gender

db040 Region (Bundesland)

pid Person ID

weight Original sampling weight

pl031 Economic status

pb220a Citizenship status

pb190 Marital status

pe040 Education level

pl111 Employment status

pgrossIncomeCat Personal gross income category

pgrossIncome Personal gross income

py010g	Employee cash or near cash income
py021g	Company car income
py050g	Self-employment income
py080g	Private pension income
py090g	Unemployment benefits
py100g	Old-age benefits
py110g	Survivor benefits
py120g	Sickness benefits
py130g	Disability benefits
py140g	Education-related allowances
.weight	Sampling weight from stratified sampling

Source

Generated using **simPop** from EU-SILC 2013 public use file. The full synthetic population can be regenerated using the script `inst/scripts/create_pop.R`.

See Also

`eusilc13puf` for the original data source.

Examples

```
data(samp)
head(samp)

# Check age distribution
hist(samp$age, breaks = 50, main = "Age Distribution")

# Introduce artificial heaping and correct it
age_heaped <- round(samp$age / 5) * 5
age_corrected <- correctHeaps(age_heaped, heaps = "5year")
```

`spoorenberg` *Spoorenberg’s Total Modified Whipple Index*

Description

The Total Modified Whipple Index extends Noumbissi’s approach by summing the absolute deviations from 1 for all ten digits, providing an overall measure of age heaping across all terminal digits.

Usage

```
spoorenberg(x, ageMin = 20, ageMax = 64, weight = NULL)
```

Arguments

`x`	numeric vector of individual ages.
`ageMin`	minimum age to include (default 20).
`ageMax`	maximum age to include (default 64).
`weight`	optional numeric vector of sampling weights.

Details

Calculate the Total Modified Whipple Index (Wtot) proposed by Spoorenberg.

The index is calculated as:

$$W_{tot} = \sum_{i=0}^{9} |1 - W_i|$$

where W_i is Noumbissi's index for digit i.

Interpretation:

- 0: no heaping (perfect data)
- Higher values indicate more heaping
- Maximum theoretical value is 16 (if all ages end in one digit)

Value

A single numeric value representing the Total Modified Whipple Index.

Author(s)

Matthias Templ

References

Spoorenberg, T. and Dutreuilh, C. (2007). Quality of age reporting: extension and application of the modified Whipple's index. *Population*, **62**(4), 729-741.

See Also

`noumbissi` for single-digit index, `whipple` for original Whipple's index.

Other heaping indices: `bachi()`, `coale_li()`, `heaping_indices()`, `jdanov()`, `kannisto()`, `myers()`, `noumbissi()`, `whipple()`

Examples

```
# No heaping
set.seed(42)
age_uniform <- sample(20:64, 10000, replace = TRUE)
spoorenberg(age_uniform)  # Should be close to 0

# Strong heaping on 0 and 5
age_heaped <- sample(seq(20, 60, by = 5), 5000, replace = TRUE)
```

```
spoorenberg(age_heaped)  # Should be high
```

sprague *Sprague Index (Multipliers)*

Description

The Sprague method uses multipliers to estimate population counts for each single year of age from 5-year interval data. This is useful for creating smooth single-year age distributions from grouped census data.

Usage

```
sprague(x)
```

Arguments

x	numeric vector of population counts in five-year age intervals. Must have exactly 17 elements corresponding to age groups 0-4, 5-9, ..., 75-79, 80+.

Details

Disaggregate 5-year age group counts into single-year ages using Sprague multipliers.

The input must be population counts for 17 five-year age groups: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70-74, 75-79, and 80+.

The Sprague multipliers are applied differently depending on the position of the age group:

- **Lowest groups** (0-4): Uses only following age groups
- **Low groups** (5-9): Uses mostly following age groups
- **Normal groups** (10-74): Uses symmetric weighting
- **High groups** (75-79): Uses mostly preceding age groups
- **Highest groups** (80+): Returned as-is (open-ended)

The total population is preserved: sum of output equals sum of input.

Value

A named numeric vector with 81 elements: single-year population counts for ages 0, 1, 2, ..., 79, and the 80+ group.

Author(s)

Matthias Templ

References

Calot, G. and Sardon, J.-P. (1998). *Methodology for the calculation of Eurostat's demographic indicators*. Detailed report by the European Demographic Observatory.

Sprague, T. B. (1880). Explanation of a new formula for interpolation. *Journal of the Institute of Actuaries*, **22**, 270-285.

See Also

`whipple` for measuring age heaping.

Examples

```
# Example from World Bank data
x <- data.frame(
  age = as.factor(c(
    "0-4", "5-9", "10-14", "15-19", "20-24",
    "25-29", "30-34", "35-39", "40-44", "45-49",
    "50-54", "55-59", "60-64", "65-69", "70-74", "75-79", "80+"
  )),
  pop = c(
    1971990, 2095820, 2157190, 2094110, 2116580,
    2003840, 1785690, 1502990, 1214170, 796934,
    627551, 530305, 488014, 364498, 259029, 158047, 125941
  )
)

# Apply Sprague multipliers
s <- sprague(x$pop)
head(s, 20)  # First 20 single-year ages

# Verify population is preserved
all.equal(sum(s), sum(x$pop))
```

whipple *Whipple Index (Original and Modified)*

Description

The Whipple index is a demographic measure used to detect and quantify age heaping (digit preference) in population data. This function implements both the original (standard) and modified versions of the index.

Usage

```
whipple(x, method = "standard", weight = NULL)
```

Arguments

`x`	numeric vector holding the ages of persons.
`method`	character string specifying which index to calculate: `"standard"` Original Whipple index (default). Ranges 0-500, with 100 indicating no heaping. `"modified"` Modified Whipple index. Ranges 0-1, with 0 indicating no heaping.
`weight`	optional numeric vector holding the sampling weights of each person. Must be the same length as `x`. If `NULL` (default), unweighted counts are used.

Details

Calculate the original or modified Whipple index to evaluate age heaping.

The original Whipple index is obtained by summing the number of persons in the age range between 23 and 62, and calculating the ratio of reported ages ending in 0 or 5 to one-fifth of the total sample. A linear decrease in the number of persons of each age within the age range is assumed. Therefore, low ages (0-22 years) and high ages (63 years and above) are excluded from analysis since this assumption is not plausible.

The original Whipple index ranges from:

- 0: when digits 0 and 5 are never reported
- 100: no preference for 0 or 5 (perfect data)
- 500: when only digits 0 and 5 are reported (maximum heaping)

For the modified Whipple index, age heaping is calculated for all ten digits (0-9). For each digit, the degree of preference or avoidance is determined, and the modified Whipple index is given by the absolute sum of these (indices - 1), scaled between 0 and 1:

- 0: ages are distributed perfectly equally across all digits
- 1: all age values end with the same digit

Value

A single numeric value representing the Whipple index.

Author(s)

Matthias Templ

References

Shryock, H. S. and Siegel, J. S. (1976). *The Methods and Materials of Demography*. New York: Academic Press.

Spoorenberg, T. and Dutreuilh, C. (2007). Quality of age reporting: extension and application of the modified Whipple's index. *Population*, **62**(4), 729-741.

See Also

sprague for disaggregating 5-year age groups.

Other heaping indices: bachi(), coale_li(), heaping_indices(), jdanov(), kannisto(), myers(), noumbissi(), spoorenberg()

Examples

```
# Equally distributed ages (no heaping)
set.seed(42)
age_uniform <- sample(1:100, 5000, replace = TRUE)
whipple(age_uniform)                    # Should be close to 100
whipple(age_uniform, method = "modified")  # Should be close to 0

# Strong heaping on 5 and 10 (ages ending in 0 or 5 only)
age_5year <- sample(seq(0, 100, by = 5), 5000, replace = TRUE)
whipple(age_5year)                     # Should be 500
whipple(age_5year, method = "modified")   # Should be close to 0.8

# Extreme heaping on 10 only (ages ending in 0 only)
age_10year <- sample(seq(0, 100, by = 10), 5000, replace = TRUE)
whipple(age_10year)                    # Should be 500
whipple(age_10year, method = "modified")  # Should be close to 1

# Using weights
weights <- runif(5000)
whipple(age_uniform, weight = weights)
```

Index