

Package ‘nipnTK’

January 28, 2026

Type Package

Title National Information Platforms for Nutrition Anthropometric Data Toolkit

Version 0.2.2

Description An implementation of the National Information Platforms for Nutrition or NiPN's analytic methods for assessing quality of anthropometric datasets that include measurements of weight, height or length, middle upper arm circumference, sex and age. The focus is on anthropometric status but many of the presented methods could be applied to other variables.

License GPL-3

Depends R (>= 2.10)

Imports stats, graphics, withr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, tufte, spelling, covr, kableExtra, bbw

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.3

URL <https://nutriverse.io/nipnTK/>,
<https://github.com/nutriverse/nipnTK>

BugReports <https://github.com/nutriverse/nipnTK/issues>

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Mark Myatt [aut] (ORCID: <<https://orcid.org/0000-0003-1119-1474>>),
Ernest Guevarra [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-4887-4415>>)

Maintainer Ernest Guevarra <ernest@guevarra.io>

Repository CRAN

Date/Publication 2026-01-28 00:30:02 UTC

Contents

ageChildren	3
ageHeaping	4
ageRatioTest	5
ah.ex01	6
as.ex01	6
as.ex02	7
boxText	8
digitPreference	9
dist.ex01	10
dp.ex01	10
dp.ex02	11
dp.ex03	11
flag.ex01	12
flag.ex02	13
flag.ex03	13
fullTable	14
greensIndex	15
histNormal	16
national.SMART	17
outliersMD	18
outliersUV	18
plot.ageChildren	19
plot.ageHeaping	20
plot.digitPreference	20
print.ageChildren	21
print.ageHeaping	22
print.ageRatioTest	23
print.digitPreference	23
print.greensIndex	24
print.sexRatioTest	25
print.skewKurt	25
pyramid.plot	26
qqNormalPlot	27
rl.ex01	28
sexRatioTest	28
skewKurt	29
sp.ex01	30
sp.ex02	31

Index

32

ageChildren

*Goodness of fit to an expected model-based age distribution***Description**

A simple model-based method for calculating expected numbers using exponential decay in a population in which births and deaths balance each other and with a 1:1 male to female sex ratio. This function is built specifically to test goodness of fit for a sample of children aged 6-59 months old grouped into four 1 year age groups and 1 half year age group (6 to less than 18 months, 18 to less than 30 months, 30 to less than 42 months, 42 to less than 54 months, and 54 months to less than 60 months).

Usage

```
ageChildren(age, u5mr = 1)
```

Arguments

age	A vector of ages. Should either be in whole months (integer) or in calculated decimal months (numeric).
u5mr	A numeric value for under five years mortality rate expressed as deaths / 10,000 persons / day. Default is set to 1.

Value

A list of class "ageChildren" with:

Variable	Description
<i>u5mr</i>	Under five years mortality rate as deaths / 10000 persons / day
<i>observed</i>	Table of counts in each (year-centred) age group
<i>expected</i>	Table of expected counts in each (year-centred) age group
<i>X2</i>	Chi-squared test statistic
<i>df</i>	Degrees of freedom for Chi-squared test
<i>p</i>	p-value for Chi-squared test

Examples

```
# Chi-Squared test for age of children in dp.ex02 sample dataset using an
# u5mr of 1 / 10,000 / day.
svy <- dp.ex02
ac <- ageChildren(svy$age, u5mr = 1)
ac

# Apply function to each sex separately
# Males
acM <- ageChildren(svy$age[svy$sex == 1], u5mr = 1)
acM
```

```
# Females
acF <- ageChildren(svy$age[svy$sex == 2], u5mr = 1)

# Simplified call to function by sex
by(svy$age, svy$sex, ageChildren, u5mr = 1)
```

ageHeaping	<i>Age-heaping analysis</i>
------------	-----------------------------

Description

Age heaping is the tendency to report children's ages to the nearest year or adults' ages to the nearest multiple of five or ten years. Age heaping is very common. This is a major reason why data from nutritional anthropometry surveys is often analysed and reported using broad age groups.

Usage

```
ageHeaping(x, divisor = 12)
```

Arguments

x	A vector of ages. Should either be in whole months (integer) or in calculated decimal months (numeric).
divisor	Divisor (usually 5, 6, 10, or 12); default is 12

Value

A list of class "ageHeaping" with:

Variable	Description
<i>X2</i>	Chi-squared test statistic
<i>df</i>	Degrees of freedom or Chi-squared test
<i>p</i>	p-value for Chi-squared test
<i>tab</i>	Table of remainders (for x \% divisor)
<i>pct</i>	Table of proportions (\%) of remainders for x \% divisor)

Examples

```
# Test for age heaping using SMART survey data in Kabul, Afghanistan (dp.ex02)
# using a divisor of 12
svy <- dp.ex02
ah12 <- ageHeaping(svy$age)
ah12

# Test for age heaping using SMART survey data in Kabul, Afthanistan (dp.ex02)
# using a divisor of 6
```

```
ah6 <- ageHeaping(svy$age, divisor = 6)
ah6
```

ageRatioTest	<i>Age ratio test</i>
--------------	-----------------------

Description

Age ratio test is an age-related test of survey and data quality. In this test, the ratio of the number of children aged from 6 to 29 months to the number of children aged from 30 to 59 months is calculated. This ratio is then compared to an expected ratio (usually set at 0.85). The difference of the observed ratio to the expected ratio is then compared statistically using Chi-squared test.

Usage

```
ageRatioTest(x, ratio = 0.85)
```

Arguments

<code>x</code>	A vector of ages. Should either be in whole months (integer) or in calculated decimal months (numeric).
<code>ratio</code>	Expected age ratio. Default is 0.85.

Value

A list of class "ageRatioTest" with:

Variable	Description
<i>expectedR</i>	Expected sex ratio
<i>expectedP</i>	Expected proportion aged 6:29 months
<i>observedR</i>	Observed sex ratio
<i>observedP</i>	Observed proportion aged 6:29 months
<i>X2</i>	Chi-squared test statistic
<i>df</i>	Degrees of freedom for Chi-squared test
<i>p</i>	p-value for Chi-squared test

Examples

```
# Age-ratio test on survey dataset from Kabul, Afghanistan (~dp.ex02~)
# with an age ratio of 0.85
svy <- dp.ex02
ageRatioTest(svy$age, ratio = 0.85)

# The age ratio test applied to data for each sex separately
by(svy$age, svy$sex, ageRatioTest, ratio = 0.85)
```

ah.ex01

Example dataset for age heaping function

Description

Anthropometric data from a Rapid Assessment Method for Older People (RAM-OP) survey in the Dadaab refugee camp in Garissa, Kenya. This is a survey of people aged sixty years and older.

Usage

ah.ex01

Format

A data frame with 593 observations and 10 variables

Variable	Description
<i>psu</i>	Primary sampling unit
<i>camp</i>	Camp name code
<i>block</i>	Block code
<i>age</i>	Age (years)
<i>sex</i>	Sex
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>demispan</i>	Demispan (cm)
<i>muac</i>	Mid-upper arm circumference (cm)
<i>oedema</i>	Oedema

Source

Data courtesy of [HelpAge International](#)

as.ex01

Example dataset for age and sex distributions function

Description

Data taken from household rosters collected as part of a household survey in Tanzania.

Usage

as.ex01

Format

A data frame of 8736 observations and 2 variables

Variable	Description
<i>age</i>	Age (years)
<i>sex</i>	Sex (1 = Male / 2 = Female)

as.ex02

Example dataset for age and sex distributions function

Description

Census data of Tanzania taken from the Wolfram|Alpha knowledge engine.

Usage

as.ex02

Format

A data frame with 20 observations and 4 variables

Variable	Description
<i>age</i>	Age group
<i>Males</i>	Total male population
<i>Females</i>	Total female population
<i>All</i>	Total population

Source

Wolfram|Alpha

boxText	<i>Plot text in a coloured bounding box.</i>
---------	--

Description

Plot text in a coloured bounding box.

Usage

```
boxText(
  x,
  y,
  labels,
  cex = 0.75,
  col = "white",
  border = FALSE,
  lwd = 0.5,
  pad = TRUE
)
```

Arguments

x, y	Co-ordinates of text that is to be plotted
labels	Text to be plotted
cex	Character expansion
col	Background colour
border	Border colour
lwd	Border width
pad	Add padding to (L) and (R) ends of bounding box

Examples

```
## Use of boxtext in the ageHeaping plot function
svy <- dp.ex02
ah12 <- ageHeaping(svy$age)

plot.new()
boxText(x = as.numeric(names(ah12$tab)),
  y = max(ah12$tab) * 0.1,
  labels = paste(sprintf(fmt = "%3.1f", ah12$pct), "%", sep = ""),
  cex = 0.5,
  pad = TRUE)
```

digitPreference	<i>Digit preference test</i>
-----------------	------------------------------

Description

Digit preference is the observation that the final number in a measurement occurs with a greater frequency that is expected by chance. This can occur because of rounding, the practice of increasing or decreasing the value in a measurement to the nearest whole or half unit, or because data are made up. The `digitPreference()` function assesses the level by which digit preference exists in a given dataset using a digit preference score (DPS).

Usage

```
digitPreference(x, digits = 1, values = 0:9)
```

Arguments

<code>x</code>	Numeric vector of measurements
<code>digits</code>	Number of decimal places in <code>x</code> . Using <code>digits = 1</code> (e.g.) allows 105 to be treated as 105.0
<code>values</code>	A vector of possible values for the final digit (default = 0:9)

Details

DPS definition from:

Kari Kuulasmaa K, Hense HW, Tolonen H (for the WHO MONICA Project), Quality Assessment of Data on Blood Pressure in the WHO MONICA Project, WHO MONICA Project e-publications No. 9, WHO, Geneva, May 1998 available from <https://www.thl.fi/publications/monica/bp/bpqa.htm>

Value

A list of class "digitPreference" with:

Variable	Description
<i>dps</i>	Digit Preference Score (DPS)
<i>tab</i>	Table of final digit counts
<i>pct</i>	Table of proportions (%) of final digit counts

Examples

```
# Digit preference test applied to anthropometric data from a single state
# from a DHS survey in a West African country
svy <- dp.ex01
digitPreference(svy$wt, digits = 1)
```

dist.ex01

Example dataset for distributions of variables and indices

Description

Anthropometric data from a SMART survey in Kabul, Afghanistan.

Usage

dist.ex01

Format

A data frame with 873 observations and 11 variables

Variable	Description
<i>psu</i>	Primary sampling unit
<i>age</i>	Age of child (months)
<i>sex</i>	Gender of child
<i>weight</i>	Weight of child (kgs)
<i>height</i>	Height of child (cm)
<i>muac</i>	Mid-upper arm circumference (mm)
<i>oedema</i>	Presence or absence of oedema
<i>haz</i>	Height-for-age z-score
<i>waz</i>	Weight-for-age z-score
<i>whz</i>	Weight-for-height z-score
<i>flag</i>	Data quality flag

dp.ex01

Example dataset for digit preference function

Description

Anthropometric data from a single state from a Demographic and Health Survey (DHS) of a West African country.

Usage

dp.ex01

Format

A data frame with 796 observations and 6 variables

Variable	Description
<i>psu</i>	Primary sampling unit
<i>age</i>	Age (months)
<i>sex</i>	Gender
<i>wt</i>	Weight (kg)
<i>ht</i>	height (cm)
<i>oedema</i>	Presence or absence of oedema

dp.ex02

*Example dataset for digit preference function***Description**

Anthropometric data from a SMART survey in Kabul, Afghanistan in a comma-separated-value (CSV) file format. This is a survey of children aged 6-59 months old.

Usage

dp.ex02

Format

A data frame with 873 observations and 7 variables

Variable	Description
<i>psu</i>	Primary sampling unit
<i>age</i>	Age of child (months)
<i>sex</i>	Gender of child
<i>weight</i>	Weight of child (kgs)
<i>height</i>	Height of child (cm)
<i>muac</i>	Mid-upper arm circumference (mm)
<i>oedema</i>	Presence or absence of oedema

dp.ex03

*Example dataset for digit preference***Description**

Anthropometric data for a sample of children living in a refugee camp in a West African country.

Usage

dp.ex03

Format

A data frame with 374 observations and 6 variables

Variable	Description
<i>age</i>	Age (months)
<i>sex</i>	Gender (1 = Male / 2 = Female)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>muac</i>	Mid-upper arm circumference (mm)
<i>oedema</i>	Presence or absence of oedema

flag.ex01

Example dataset for identifying outliers using flags

Description

Anthropometric data from a SMART survey in Sudan.

Usage

flag.ex01

Format

A data frame with 786 observations and 11 variables

Variable	Description
<i>psu</i>	Primary sampling unit
<i>child</i>	Child ID
<i>age</i>	Age (months)
<i>sex</i>	Gender (1 = Male / 2 = Female)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>muac</i>	Mid-upper arm circumference (mm)
<i>oedema</i>	Presence or absence of oedema
<i>haz</i>	Height-for-age z-score
<i>waz</i>	Weight-for-age z-score
<i>whz</i>	Weight-for-height z-score

flag.ex02

*Example dataset for identifying outliers using flags***Description**

Anthropometric data from a survey of children 11 years or older attending school in Ethiopia.

Usage

flag.ex02

Format

A data.frame with 973 observations and 7 variables.

Variable	Description
<i>school</i>	School ID
<i>sex</i>	Gender (1 = Male / 2 = Female)
<i>ageMonths</i>	Age (months)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>haz</i>	Height-for-age z-score
<i>baz</i>	Body mass index (BMI)-for-age z-score

flag.ex03

*Example dataset for identifying outliers using flags***Description**

Anthropometric data from a national survey in Nigeria.

Usage

flag.ex03

Format

A data frame with 18330 observations and 10 variables

Variable	Description
<i>psu</i>	Primary sampling unit
<i>region</i>	Region code
<i>state</i>	State
<i>age</i>	Age (months)

<i>sex</i>	Gender (1 = Male / 2 = Female)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>haz</i>	Height-for-age z-score
<i>waz</i>	Weight-for-age z-score
<i>whz</i>	Weight-for-height z-score

fullTable	<i>Fill out a one-dimensional table to include a specified range of values</i>
-----------	--

Description

Fill out a one-dimensional table to include a specified range of values

Usage

```
fullTable(x, values = min(x, na.rm = TRUE):max(x, na.rm = TRUE))
```

Arguments

x	A vector to tabulate
values	A vector of values to be included in a table. Default is: min(x, na.rm = TRUE):max(x, na.rm = TRUE)

Value

A table object including zero cells

Examples

```
# Generate some artificial data and then apply `fullTable()`
set.seed(0)
finalDigits <- sample(x = 0:9, size = 1000, replace = TRUE)
fullTable(finalDigits)
```

greensIndex

*Green's Index of Dispersion***Description**

Implementation of the Green's Index of Dispersion by bootstrap. The sampling distribution of the Green's Index is not well described hence bootstrapping is used to test whether the distribution of cases across primary sampling units is random.

Usage

```
greensIndex(data, psu, case, replicates = 999)
```

Arguments

data	Survey dataset (as an R data.frame)
psu	Name of variable holding PSU (cluster) data as a character vector of length = 1 (e.g. psu)
case	Name of variable holding case status as a character vector of length = 1 (e.g. GAM). The function assumes that case status is coded with 1 = case
replicates	Number of bootstrap replicates (default is 9999)

Details

The value of Green's Index can range between $-1/(n - 1)$ for maximum uniformity (specific to the dataset) and one for maximum clumping. The interpretation of Green's Index is straightforward:

Green's Index Value	Interpretation
<i>Green's Index close to 0</i>	Random
<i>Green's Index greater than 0</i>	Clumped (i.e. more clumped than random)
<i>Green's Index less than 0</i>	Uniform (i.e. more uniform than random)

Value

A list of class GI with names:

Variable	Description
<i>GI</i>	Estimate of Green's index
<i>LCL</i>	95% LCL for GI
<i>UCL</i>	95% UCL for GI
<i>minGI</i>	Minimum possible GI (maximum uniformity) for the data
<i>p</i>	p-value (H0: = Random distribution of cases across PSUs)

Examples

```
# Apply Green's Index using anthropometric data from a SMART survey in Sudan
# (flag.ex01)
svy <- flag.ex01
svy$flag <- 0
svy$flag <- ifelse(!is.na(svy$haz) & (svy$haz < -6 | svy$haz > 6),
                  svy$flag + 1, svy$flag)
svy$flag <- ifelse(!is.na(svy$whz) & (svy$whz < -5 | svy$whz > 5),
                  svy$flag + 2, svy$flag)
svy$flag <- ifelse(!is.na(svy$waz) & (svy$waz < -6 | svy$waz > 5),
                  svy$flag + 4, svy$flag)
svy <- svy[svy$flag == 0, ]
svy$stunted <- ifelse(svy$haz < -2, 1, 2)

## set seed to 0 to replicate results
set.seed(0)
greensIndex(data = svy, psu = "psu", case = "stunted")
```

histNormal

Histogram with normal curve superimposed to help with “by-eye” assessments of normality of distribution

Description

Histogram with normal curve superimposed to help with “by-eye” assessments of normality of distribution

Usage

```
histNormal(
  x,
  xlab = deparse(substitute(x)),
  ylab = "Frequency",
  main = deparse(substitute(x)),
  breaks = "Sturges",
  ylim = NULL
)
```

Arguments

x	A numeric vector
xlab	x-axis label
ylab	y-axis label
main	Plot title
breaks	Passed to hist() function (?hist for details)
ylim	y-axis limits

Examples

```
# histNormal() with data from a SMART survey in Kabul, Afghanistan
# (dist.ex01)
svy <- dist.ex01
histNormal(svy$muac)
histNormal(svy$haz)
histNormal(svy$waz)
histNormal(svy$whz)
```

national.SMART	<i>Add SMART flags to a stratified sample survey (e.g. MICS, DHS, national SMART)</i>
----------------	---

Description

Add SMART flags to a stratified sample survey (e.g. MICS, DHS, national SMART)

Usage

```
national.SMART(x, strata, indices = c("haz", "whz", "waz"))
```

Arguments

x	Survey dataset (as an R data.frame) with indices present
strata	Name of column in x that defines the strata
indices	Names of columns in x containing indices

Value

A data.frame with same structure as x with a flagSMART column added. This column is coded using sums of powers of two

Examples

```
# Use the national.SMART() function to flag indices from a national
# SMART survey in Nigeria (flag.ex03)
svy <- flag.ex03
svyFlagged <- national.SMART(x = svy, strata = "state")

# Exclude records with flagging codes relevant to whz:
svyFlagged <- svyFlagged[!(svyFlagged$flagSMART %in% c(2, 3, 6, 7)), ]
```

outliersMD	<i>Mahalanobis distance to detect bivariate outliers</i>
------------	--

Description

Mahalanobis distance to detect bivariate outliers

Usage

```
outliersMD(x, y, alpha = 0.001)
```

Arguments

x	Numeric vector
y	Numeric vector
alpha	Critical alpha value to detect an outlier (defaults to 0.001)

Value

A logical vector (TRUE for an outlier at $p < \alpha$)

Examples

```
# Use outliersMD() to detect outliers in an anthropometric data from
# a SMART survey from the Democratic Republic of Congo (sp.ex01)
svy <- sp.ex01
svy[outliersMD(svy$height,svy$weight), ]
```

outliersUV	<i>IQR to detect univariate outliers</i>
------------	--

Description

IQR to detect univariate outliers

Usage

```
outliersUV(x, fence = 1.5)
```

Arguments

x	Numeric vector
fence	IQR multiplier (defaults to 1.5)

Value

A logical vector (TRUE for an outlier)

Examples

```
# Use outliersUV() to detect univariate outliers in an anthropometric
# dataset from a SMART survey from Angola (rl.ex01)
svy <- rl.ex01
svy[outliersUV(svy$muac), ]
```

plot.ageChildren	<i>Plot helper function for ageChildren() function</i>
------------------	--

Description

Plot helper function for [ageChildren\(\)](#) function

Usage

```
## S3 method for class 'ageChildren'
plot(x, ...)
```

Arguments

x	Object resulting from applying ageChildren() function
...	Additional barplot() graphical parameters

Value

Bar plot comparing table of observed counts vs table of expected counts

Examples

```
# Plot Chi-Squared test for age of children in dp.ex02 sample dataset using
# an u5mr of 1 / 10,000 / day.
svy <- dp.ex02
ac <- ageChildren(svy$age, u5mr = 1)
plot(ac)
```

plot.ageHeaping *plot() helper functions for ageHeaping() functions*

Description

`plot()` helper functions for `ageHeaping()` functions

Usage

```
## S3 method for class 'ageHeaping'
plot(x, main = "", xlab = "Remainder", ylab = "Frequency", cex = 0.75, ...)
```

Arguments

<code>x</code>	Object resulting from applying the <code>ageHeaping()</code> function
<code>main</code>	Title of plot
<code>xlab</code>	x-axis label; default is Remainder
<code>ylab</code>	y-axis label; default is Frequency
<code>cex</code>	Character expansion (numeric); default is 0.75
<code>...</code>	Additional <code>plot()</code> graphical parameters

Value

Barplot of frequency of remainders of age when divided by a specified divisor

Examples

```
# Plot age heaping test results on SMART survey data in Kabul, Afghanistan
# (dp.ex02) using a divisor of 12
svy <- dp.ex02
ah12 <- ageHeaping(svy$age)
plot(ah12)
```

plot.digitPreference *plot() helper function for digitPreference() function*

Description

`plot()` helper function for `digitPreference()` function

Usage

```
## S3 method for class 'digitPreference'
plot(x, main = "", xlab = "Final Digit", ylab = "Frequency", cex = 0.75, ...)
```

Arguments

x	Object resulting from applying the <code>digitPreference()</code> function.
main	Title of plot
xlab	x-axis label; default is "Final Digit"
ylab	y-axis label; default is "Frequency"
cex	Character expansion; default is 0.75
...	Additional <code>plot()</code> parameters

Value

Plotted output of `digitPreference()` function comparing the frequencies of the various final digits

Examples

```
# Plot output of digit preference test applied to anthropometric data from a
# single state from a DHS survey in a West African country
svy <- dp.ex01
digitPreference(svy$wt, digits = 1)
plot(digitPreference(svy$wt, digits = 1))
```

print.ageChildren	<code>print()</code> helper function for <code>ageChildren()</code> function
-------------------	--

Description

`print()` helper function for `ageChildren()` function

Usage

```
## S3 method for class 'ageChildren'
print(x, ...)
```

Arguments

x	Object resulting from applying <code>ageChildren()</code> function
...	Additional <code>print()</code> arguments

Value

Printed output of `ageChildren()` function

Examples

```
# Print Chi-Squared test for age of children in dp.ex02 sample dataset using
# an u5mr of 1 / 10,000 / day.
svy <- dp.ex02
ac <- ageChildren(svy$age, u5mr = 1)
print(ac)
```

print.ageHeaping	<i>print() helper functions for ageHeaping() functions</i>
------------------	--

Description

`print()` helper functions for `ageHeaping()` functions

Usage

```
## S3 method for class 'ageHeaping'
print(x, ...)
```

Arguments

<code>x</code>	Object resulting from applying the <code>ageHeaping()</code> function
<code>...</code>	Additional <code>print()</code> arguments

Value

Printed output of the `ageHeaping()` function

Examples

```
# Print age heaping test on SMART survey data in Kabul, Afghanistan (dp.ex02)
# using a divisor of 12
svy <- dp.ex02
ah12 <- ageHeaping(svy$age)
print(ah12)
```

print.ageRatioTest *print()* helper function for `ageRatioTest()` function

Description

print() helper function for `ageRatioTest()` function

Usage

```
## S3 method for class 'ageRatioTest'  
print(x, ...)
```

Arguments

x Object resulting from applying `ageRatioTest()` function
... Additional *print()* arguments

Value

Printed output of `ageRatioTest()` function

Examples

```
# Print age-ratio test results for survey dataset from Kabul, Afghanistan  
svy <- dp.ex02  
print(ageRatioTest(svy$age, ratio = 0.85))
```

print.digitPreference *print()* helper function for `digitPreference()` function

Description

print() helper function for `digitPreference()` function

Usage

```
## S3 method for class 'digitPreference'  
print(x, ...)
```

Arguments

x Object resulting from applying the `digitPreference()` function.
... Additional *print()* parameters

Value

Printed output of `digitPreference()` function

Examples

```
# Print output of digit preference test applied to anthropometric data from a
#single state from a DHS survey in a West African country
svy <- dp.ex01
print(digitPreference(svy$wt, digits = 1))
```

print.greensIndex	<code>print()</code> helper function for print.greensIndex() function
-------------------	---

Description

`print()` helper function for print.greensIndex() function

Usage

```
## S3 method for class 'greensIndex'
print(x, ...)
```

Arguments

x	Object resulting from applying the <code>greensIndex()</code> function
...	Additional <code>print()</code> parameters

Value

Printed output of `greensIndex()` function

Examples

```
# Apply Green's Index using anthropometric data from a SMART survey in Sudan
# (flag.ex01)
svy <- flag.ex01
svy$flag <- 0
svy$flag <- ifelse(!is.na(svy$haz) & (svy$haz < -6 | svy$haz > 6), svy$flag + 1, svy$flag)
svy$flag <- ifelse(!is.na(svy$whz) & (svy$whz < -5 | svy$whz > 5), svy$flag + 2, svy$flag)
svy$flag <- ifelse(!is.na(svy$waz) & (svy$waz < -6 | svy$waz > 5), svy$flag + 4, svy$flag)
svy <- svy[svy$flag == 0, ]
svy$stunted <- ifelse(svy$haz < -2, 1, 2)
gi <- greensIndex(data = svy, psu = "psu", case = "stunted")
print(gi)
```

print.sexRatioTest *print()* helper function for `sexRatioTest()` function

Description

`print()` helper function for `sexRatioTest()` function

Usage

```
## S3 method for class 'sexRatioTest'  
print(x, ...)
```

Arguments

x	Output resulting from applying the <code>sexRatioTest()</code> function
...	Additional <code>print()</code> parameters

Value

Printed output of `sexRatioTest()` function

Examples

```
# Use sexRatioTest() on household roster data from a survey in Tanzania  
# (as.ex01) and census data of Tanzania extracted from Wolfram|Alpha knowledge  
# engine (as.ex02)  
svy <- as.ex01  
ref <- as.ex02  
censusM <- sum(ref$Males)  
censusF <- sum(ref$Females)  
srt <- sexRatioTest(svy$sex, codes = c(1, 2), pop = c(censusM, censusF))  
print(srt)
```

print.skewKurt *print()* helper function for `skewKurt()` function

Description

`print()` helper function for `skewKurt()` function

Usage

```
## S3 method for class 'skewKurt'  
print(x, ...)
```

Arguments

x	Object resulting from applying the <code>skewKurt()</code> function
...	Additional <code>print()</code> parameters

Value

Printed output of `skewKurt()` function

Examples

```
# Use skewKurt() on an anthropometric data from a SMART survey in
# Kabul, Afghanistan (dist.ex01)
svy <- dist.ex01
sk <- skewKurt(svy$muac)
print(sk)
```

pyramid.plot

Pyramid plot function for creating population pyramids.

Description

Pyramid plot function for creating population pyramids.

Usage

```
pyramid.plot(
  x,
  g,
  main = paste("Pyramid plot of", deparse(substitute(x)), "by", deparse(substitute(g))),
  xlab = paste(deparse(substitute(g)), "(", levels(as.factor(g))[1], "/",
    levels(as.factor(g))[2], ")"),
  ylab = deparse(substitute(x)),
  col = "white",
  ...
)
```

Arguments

x	Vector of ages (usually grouped)
g	Vector of groups (usually sex)
main	Plot title
xlab	x-axis label
ylab	y-axis label

col	Colours for bars. Either a single colour (default is col = "white") for all bars, two colours (e.g. col = c("lightblue", "pink")) for left hand side bars and right hand side bars respectively, or many colours allocated on a checkerboard basis to each bar
...	Other graphical parameters

Value

A table of x by g (invisible)

Examples

```
# Use pyramid.plot() on anthropometric data from a SMART survey in
# Kabul, Afghanistan (dp.ex02)
svy <- dp.ex02
pyramid.plot(svy$age, svy$sex)
```

qqNormalPlot

Normal quantile-quantile plot

Description

Normal quantile-quantile plot

Usage

```
qqNormalPlot(x)
```

Arguments

x	A numeric vector
---	------------------

Examples

```
# qqNormalPlot() with data from a SMART survey in Kabul, Afghanistan
# (dist.ex01)
svy <- dist.ex01
qqNormalPlot(svy$muac)
qqNormalPlot(svy$haz)
qqNormalPlot(svy$waz)
qqNormalPlot(svy$whz)
```

r1.ex01

Example dataset for checking ranges and legal values

Description

Anthropometric data from a SMART survey in Angola.

Usage

```
r1.ex01
```

Format

A data frame with 906 observations and 6 variables

Variable	Description
<i>age</i>	Age (months)
<i>sex</i>	Gender (1 = Male / 2 = Female)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>muac</i>	Mid-upper arm circumference (mm)
<i>oedema</i>	Presence or absence of oedema

sexRatioTest

Sex Ratio Test

Description

Sex Ratio Test

Usage

```
sexRatioTest(sex, codes = c(1, 2), pop = c(1, 1))
```

Arguments

sex	A vector of values that indicate sex
codes	Codes used to identify males and females (in that order)
pop	Relative populations of males and females (in that order)

Value

A list of class "sexRatioTest" with:

Variable	Description
<i>pM</i>	Observed proportion male
<i>eM</i>	Expected proportion male
<i>X2</i>	Chi-squared test statistic
<i>df</i>	Degrees of freedom for Chi-squared test
<i>p</i>	p-value for Chi-squared test

Examples

```
# Use sexRatioTest() on household roster data from a survey in Tanzania
# (as.ex01) and census data of Tanzania extracted from Wolfram|Alpha knowledge
# engine (as.ex02)
svy <- as.ex01
ref <- as.ex02
censusM <- sum(ref$Males)
censusF <- sum(ref$Females)
sexRatioTest(svy$sex, codes = c(1, 2), pop = c(censusM, censusF))
```

skewKurt	<i>Skew and kurtosis</i>
----------	--------------------------

Description

Skew and kurtosis

Usage

```
skewKurt(x)
```

Arguments

x Numeric vector

Value

A list of class "skewKurt" with:

Variable	Description
<i>s</i>	Skewness with direction
<i>s.se</i>	Standard error of skewness
<i>s.z</i>	Test statistic ($s.z = s / s.se$)
<i>s.p</i>	p-value ($s \neq 0$)
<i>k</i>	Excess kurtosis with direction

<i>k.se</i>	Standard error of excess kurtosis
<i>k.z</i>	Test statistic ($k.z = k / k.se$)
<i>k.p</i>	p-value ($k \neq 0$)

Examples

```
# Use skewKurt() on an anthropometric data from a SMART survey in
# Kabul, Afghanistan (dist.ex01)
svy <- dist.ex01
skewKurt(svy$muac)
```

sp.ex01	<i>Example dataset for using scatterplots to identify outliers</i>
---------	--

Description

Anthropometric data from a SMART survey in the Democratic Republic of Congo.

Usage

```
sp.ex01
```

Format

A data frame with 895 observations and 6 variables

Variable	Description
<i>age</i>	Age (months)
<i>sex</i>	Gender (1 = Male / 2 = Female)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>muac</i>	Mid-upper arm circumference (mm)
<i>oedema</i>	Presence or absence of oedema

sp.ex02*Example dataset for using scatterplots to identify outliers*

Description

Anthropometric data from a survey of school-age (i.e., between 5 and 15 years) children from Pakistan

Usage

sp.ex02

Format

A data frame with 849 observations and 9 variables

Variable	Description
<i>region</i>	Region code
<i>school</i>	School code
<i>ageMonths</i>	Age (months)
<i>sex</i>	Sex (1 = Male / 2 = Female)
<i>weight</i>	Weight (kg)
<i>height</i>	Height (cm)
<i>haz</i>	Height-for-age z-score
<i>waz</i>	Weight-for-age z-score
<i>baz</i>	Body mass index (BMI)-for-age z-score

Index

* datasets

- ah.ex01, [6](#)
- as.ex01, [6](#)
- as.ex02, [7](#)
- dist.ex01, [10](#)
- dp.ex01, [10](#)
- dp.ex02, [11](#)
- dp.ex03, [11](#)
- flag.ex01, [12](#)
- flag.ex02, [13](#)
- flag.ex03, [13](#)
- rl.ex01, [28](#)
- sp.ex01, [30](#)
- sp.ex02, [31](#)

ageChildren, [3](#)
ageChildren(), [19](#), [21](#)
ageHeaping, [4](#)
ageHeaping(), [20](#), [22](#)
ageRatioTest, [5](#)
ageRatioTest(), [23](#)
ah.ex01, [6](#)
as.ex01, [6](#)
as.ex02, [7](#)

barplot(), [19](#)
boxText, [8](#)

digitPreference, [9](#)
digitPreference(), [9](#), [20](#), [21](#), [23](#), [24](#)
dist.ex01, [10](#)
dp.ex01, [10](#)
dp.ex02, [11](#)
dp.ex03, [11](#)

flag.ex01, [12](#)
flag.ex02, [13](#)
flag.ex03, [13](#)
fullTable, [14](#)

greensIndex, [15](#)

greensIndex(), [24](#)

hist(), [16](#)
histNormal, [16](#)

national.SMART, [17](#)

outliersMD, [18](#)
outliersUV, [18](#)

plot(), [20](#), [21](#)
plot.ageChildren, [19](#)
plot.ageHeaping, [20](#)
plot.digitPreference, [20](#)
print(), [21–26](#)
print.ageChildren, [21](#)
print.ageHeaping, [22](#)
print.ageRatioTest, [23](#)
print.digitPreference, [23](#)
print.greensIndex, [24](#)
print.sexRatioTest, [25](#)
print.skewKurt, [25](#)
pyramid.plot, [26](#)

qqNormalPlot, [27](#)

rl.ex01, [28](#)

sexRatioTest, [28](#)
sexRatioTest(), [25](#)
skewKurt, [29](#)
skewKurt(), [25](#), [26](#)
sp.ex01, [30](#)
sp.ex02, [31](#)