Package 'psyntur'

September 15, 2021

Title Helper Tools for Teaching Statistical Data Analysis

Description Provides functions and data-sets that are helpful for teaching statistics and data analysis. It was originally designed for use when teaching students in the Psychology Department at Nottingham Trent University.

Version 0.1.0

License GPL-3

Maintainer Mark Andrews <mark.andrews@ntu.ac.uk>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports ggplot2 (>= 3.3.5), dplyr (>= 1.0.7), tidyr (>= 1.1.3), magrittr (>= 2.0.1), formula.tools (>= 1.7.1), GGally (>= 2.1.2), purrr (>= 0.3.4), ggthemes (>= 4.2.4), psych (>= 2.1.6), plyr (>= 1.8.6), effsize (>= 0.8.1), rlang (>= 0.4.11), fastDummies (>= 1.6.3), ez (>= 4.4), tidyselect (>= 1.1.1)

Depends R (>= 2.10)

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Mark Andrews [aut, cre],
Jens Roeser [aut]

Repository CRAN

Date/Publication 2021-09-15 09:20:05 UTC

R topics documented:

`ansur` *Anthropometric data from US Army Personnel*

Description

Data on the height, weight, handedness from men and women of different ages and different races.

Usage

```
ansur
```

Format

A data frame with 6068 observations from 9 variables.

subjectid Unique ID of the person

gender Binary variable indicating the subject's sex: `male` or `female`.

height Height in centimeters.

weight Weight in kilograms.

handedness Categorical variable indicating if the person is left, or right handed, or both.

age Age in years

race Race, with categories like `white`, `black`, `hispanic`.

height_tercile The tercile of the person's height.

age_tercile The tercile of the person's weight.

Source

This data is a transformed version of data sets obtained the Anthropometric Survey of US Army Personnel (ANSUR 2 or ANSUR II).

`cohen_d` *Cohen's d and Hedges g effect size*

Description

This is wrapper to the `effsize::cohen.d()` function.

Usage

```
cohen_d(...)
```

Arguments

`...`	A comma separated list of arguments. See `effsize::cohen.d()`.

Value

A list of class `effsize` as returned by `effsize::cohen.d()`.

Examples

```
cohen_d(weight ~ gender, data = ansur)
cohen_d(age ~ gender, data = schizophrenia)
```

cronbach *Calculate Cronbach's alpha for sets of psychometric scale items*

Description

This function calculates the Cronbach alpha for one or more sets of psychometric scale items. Each item is a variable in a data frame. Each set of items is defined by a tidy selection of a set of items.

Usage

```
cronbach(.data, ..., .ci = 0.95)
```

Arguments

`.data`	A data frame with columns that are psychometric items.
`...`	A set of comma separated tidy selectors that selects sets of columns from `.data`. For each set of columns, the Cronbach's alpha is computed.
`.ci`	The value of the confidence interval to calculate.

Value

A data frame whose rows are psychometric scales and for each scale, we have the Cronbach's alpha, and the lower and upper bound of the confidence interval on alpha.

Examples

```
# Return the Cronbach alpha and 95% ci for two scales.
# The first scale, named `x`, is identified by all items beginning with `x_`.
# The second scale, named `y`, is identified by the consecutive items from `y_1` to `y_10`.
cronbach(test_psychometrics,
         x = starts_with('x'),
         y = y_1:y_10)
```

densityplot *A density plot*

Description

This is a wrapper to the typical `ggplot` based density plot, i.e., using `geom_density`. A continuous variable, `x`, is required as an input. Optionally, a by categorical variable can be provided.

Usage

```
densityplot(
  x,
  data,
  by = NULL,
  position = "stack",
  facet = NULL,
  facet_type = "wrap",
  alpha = 1,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

`x` The numeric variable that is to be density plotted.

`data` A data frame with at least one numeric variable (the `x` variable).

`by` A categorical variable by which to group the `x` values. If provided there will be one density plot for each set of `x` values grouped by the values of the `by` variable.

`position` If the `by` variable is provided, there are three ways these multiple density plots can be positioned: stacked (`position = 'stack'`), superimposed (position = identity').

`facet` A character string or character vector. If provided, we `facet_wrap` (by default) the histogram by the variables. This is equivalent to the `facet_wrap(variables)` in `ggplot2`.

`facet_type` By default, this takes the value of `wrap`, and `facet` leads to a facet wrap. If `facet_type` is `grid`, then `facet` gives us a `facet_grid`.

`alpha` The transparency to for the filled histogram bars. This is probably only required when using `position = 'identity'`.

`xlab` The label of the x-axis (defaults to the `x` variable name).

`ylab` The label of the y-axis (defaults to the `y` variable name).

Value

A `ggplot2::ggplot` object, which may be modified with further `ggplot2` commands.

Examples

```
densityplot(x = age, data = schizophrenia, by = gender)
```

describe *Calculate descriptive statistics*

Description

This function is a lightweight wrapper to dplyr's summarize function. It can be used to calculate any descriptive or summary statistic for any variable in the data set. Optionally, a by grouping variable can be used, and then the summary statistics are calculated for each subgroup defined by the different values of the by variable.

Usage

```
describe(data, by = NULL, ...)
```

Arguments

data	A data frame
by	A grouping variable. If included, the data will be grouped by the values of the by variable before the summary statistics are applied.
...	Arguments of functions applied to variables, e.g. avg = mean(x).

Value

A tibble data frame with each row providing descriptive statistics for selected variables for each value of the grouping by variable.

Examples

```
describe(faithfulfaces, avg = mean(faithful), stdev = sd(faithful))
describe(faithfulfaces, by = face_sex, avg = mean(faithful), stdev = sd(faithful))
```

describe_across *Apply multiple descriptive functions to multiple variables*

Description

This function is a wrapper to dplyr's summarize used with the across function. For each variable in a set of variables, calculate each summary statistic from a list of summary statistic functions. Optionally, group the variables by a grouping variable, and then calculate the statistics. Optionally, the tibble that is returned by default, which is in a wide format, can be pivoted to a long format.

Usage

```
describe_across(data, variables, functions, by = NULL, pivot = FALSE)
```

Arguments

`data`	A data frame
`variables`	A vector of variables in `data`
`functions`	A list of summary statistic function. If it is named list, which is recommended, the names of the functions will be used to make the names of the returned data frame.
`by`	A grouping variable. If included, the `data` will be grouped by the values of the by variable before the summary statistics are applied.
`pivot`	A logical variable indicating if the wide format da

Value

A tibble data frame. If `pivot = F`, which is the default, the data frames contains one row per value of the `by` variable, or just one row overall if there is no `by` variable. If `pivot = T`, there will be `k` + 1 columns if there is no `by` variable, or `k` + 2 columns if there is a `by` variable, where `k` is the number of functions.

Examples

```
describe_across(faithfulfaces,
                variables = c(trustworthy, faithful),
                functions = list(avg = mean, stdev = sd),
                pivot = TRUE)
describe_across(faithfulfaces,
                variables = c(trustworthy, faithful),
                functions = list(avg = mean, stdev = sd),
                by = face_sex)
describe_across(faithfulfaces,
                variables = c(trustworthy, faithful),
                functions = list(avg = mean, stdev = sd),
                by = face_sex,
                pivot = TRUE)
```

`ez_anova` *Analysis of variance*

Description

This is wrapper to the `ez::ezANOVA()` function.

Usage

```
ez_anova(
  data,
  dv,
  wid,
  within = NULL,
  within_full = NULL,
  within_covariates = NULL,
  between = NULL,
  between_covariates = NULL,
  observed = NULL,
  diff = NULL,
  reverse_diff = FALSE,
  type = 2,
  white.adjust = FALSE,
  detailed = FALSE,
  return_aov = FALSE
)
```

Arguments

data Data frame containing the data to be analyzed.

dv Name of the column in data that contains the dependent variable. Values in this column must be numeric.

wid Name of the column in data that contains the variable specifying the case/Ss identifier. This should be a unique value per case/Ss.

within Names of columns in data that contain predictor variables that are manipulated (or observed) within-Ss.

within_full Same as within, but intended to specify the full within-Ss design in cases where the data have not already been collapsed to means per condition specified by within and when within only specifies a subset of the full design.

within_covariates

Names of columns in data that contain predictor variables that are manipulated (or observed) within-Ss and are to serve as covariates in the analysis.

between Names of columns in data that contain predictor variables that are manipulated (or observed) between-Ss.

between_covariates

Names of columns in data that contain predictor variables that are manipulated (or observed) between-Ss and are to serve as covariates in the analysis.

observed Names of columns in data that are already specified in either within or between that contain predictor variables that are observed variables (i.e. not manipulated).

diff Names of any variables to collapse to a difference score. If a single value, may be specified by name alone; if multiple values, must be specified as a .() list.

reverse_diff Logical. If TRUE, triggers reversal of the difference collapse requested by diff. Take care with variables with more than 2 levels.

type Numeric value (either 1, 2 or 3) specifying the Sums of Squares type to employ when data are unbalanced (eg. when group sizes differ).

white.adjust Only affects behaviour if the design contains only between-Ss predictor variables. If not FALSE, the value is passed as the white.adjust argument to Anova, which provides heteroscedasticity correction.

`detailed` Logical. If TRUE, returns extra information (sums of squares columns, intercept row, etc.) in the ANOVA table.

`return_aov` Logical. If TRUE, computes and returns an aov object corresponding to the requested ANOVA (useful for computing post-hoc contrasts).

Value

A list containing one or more components as returned by `ez::ezANOVA()`.

Examples

```
ez_anova(data = selfesteem2_long,
            dv = score,
            wid = id,
            within = c(time, treatment),
            detailed = TRUE,
            return_aov = TRUE)
```

`faithfulfaces` *Faithfulness from a Photo?*

Description

Ratings from a facial photo and actual faithfulness.

Usage

```
faithfulfaces
```

Format

A data frame with 170 observations on the following 7 variables.

sex_dimorph Rating of sexual dimorphism (masculinity for males, femininity for females)

attractive Rating of attractiveness

cheater Was the face subject unfaithful to a partner?

trustworthy Rating of trustworthiness

faithful Rating of faithfulness

face_sex Sex of face (female or male)

rater_sex Sex of rater (female or male)

Details

College students were asked to look at a photograph of an opposite-sex adult face and to rate the person, on a scale from 1 (low) to 10 (high), for attractiveness. They were also asked to rate trustworthiness, faithfulness, and sexual dimorphism (i.e., how masculine a male face is and how feminine a female face is). Overall, 68 students (34 males and 34 females) rated 170 faces (88 men and 82 women).

Source

This data set was taken from the Stats2Data R package. From the description in that package, the original is based on G. Rhodes et al. (2012), "Women can judge sexual unfaithfulness from unfamiliar men's faces," Biology Letters, November 2012. All of the 68 raters were heterosexual Caucasians, as were the 170 persons who were rated. (We have deleted 3 subjects with missing values and 16 subjects who were over age 35.)

get_dummy_code *Show the dummy code of a categorical variable*

Description

For each value of a categorical variables, show the binary code used in a regression model to represent its value. This is wrapper to the `fastDummies::dummy_cols()` function.

Usage

```
get_dummy_code(Df, variable)
```

Arguments

`Df`	A data frame
`variable`	A categorical variable (e.g. character vector or factor).

Value

A data frame whose rows provide the dummy code for each distinct value of `variable`.

Examples

```
get_dummy_code(PlantGrowth, group)
```

histogram *A histogram*

Description

This is a wrapper to the typical `ggplot` based histogram, i.e., using `geom_histogram`. A continuous variable, `x`, is required as an input. Optionally, a by categorical variable can be provided.

Usage

```
histogram(
  x,
  data,
  by = NULL,
  position = "stack",
  facet = NULL,
  facet_type = "wrap",
  bins = 10,
  alpha = 1,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

`x` The numeric variable that is to be histogrammed.

`data` A data frame with at least one numeric variable (the `x` variable).

`by` A categorical variable by which to group the `x` values. If provided there will be one histogram for each set of `x` values grouped by the values of the `by` variable.

`position` If the `by` variable is provided, there are three ways these multiple histograms can be positioned: stacked (`position = 'stack'`), side by side (`position = 'dodge'`), superimposed (position = identity').

`facet` A character string or character vector. If provided, we `facet_wrap` (by default) the histogram by the variables. This is equivalent to the `facet_wrap(variables)` in `ggplot2`.

`facet_type` By default, this takes the value of `wrap`, and `facet` leads to a facet wrap. If `facet_type` is `grid`, then `facet` gives us a `facet_grid`.

`bins` The number of bins to use in the histogram.

`alpha` The transparency to for the filled histogram bars. This is probably only required when using `position = 'identity'`.

`xlab` The label of the x-axis (defaults to the `x` variable name).

`ylab` The label of the y-axis (defaults to the `y` variable name).

Value

A `ggplot2::ggplot` object, which may be modified with further `ggplot2` commands.

Examples

```
histogram(x= age, data = schizophrenia, by = gender, bins = 20)
histogram(x= age, data = schizophrenia, by = gender, position = 'identity', bins = 20, alpha = 0.7)
histogram(x= age, data = schizophrenia, by = gender, position = 'dodge', bins = 20)
histogram(x = weight, bins = 20, data = ansur, facet = height_tercile)
histogram(x = weight, bins = 20, data = ansur,
         facet = c(height_tercile, age_tercile), facet_type = 'grid')
```

`interaction_line_plot` *Make a interaction line plot*

Description

Make a interaction line plot

Usage

```
interaction_line_plot(y, x, by, data, ylim = NULL, xlab = NULL, ylab = NULL)
```

Arguments

`y`	A continuous variable to be plotted along the y-axis
`x`	A continuous variable to be plotted along the x-axis
`by`	A categorical variable by which we split the data and create one line plot for each resulting group
`data`	A data frame with the `x`, `y`, `by` variables
`ylim`	A vector of limits for the y-axis
`xlab`	The label of the x-axis (defaults to the `x` variable name).
`ylab`	The label of the y-axis (defaults to the `y` variable name).

Value

A `ggplot2::ggplot` object, which may be modified with further `ggplot2` commands.

Examples

```
interaction_line_plot(y = score, x = time, by = treatment,
                      data = selfesteem2_long, ylim = c(70, 100))
interaction_line_plot(y = score, x = time, by = treatment,
                      data = selfesteem2_long,
                      xlab = 'measurement time',
                      ylab = 'self esteem score',
                      ylim = c(70, 100))
```

`jobsatisfaction` *Job Satisfaction Data for Two-Way ANOVA*

Description

Contains the job satisfaction score organized by gender and education level. This data set was taken from the `datarium` R package.

Usage

```
data("jobsatisfaction")
```

Format

A data frame with 58 rows and 3 columns.

Examples

```
data(jobsatisfaction)
jobsatisfaction
```

`pairedsleep` *Paired sleep data*

Description

Data which show the effect of two soporific drugs (increase in hours of sleep compared to control) on 10 patients.

Usage

```
pairedsleep
```

Format

A data frame with 10 observations on the following 3 variables.

ID The patient ID.

y1 The increase in hours, relative to control, for drug 1.

y2 The increase in hours, relative to control, for drug 2.

Source

This data is a transformed version of datasets::sleep.

paired_t_test *Paired samples t-test*

Description

A wrapper to `stats::t.test()` with `paired = TRUE`.

Usage

```
paired_t_test(y1, y2, data, ...)
```

Arguments

`y1` A numeric vector of observations

`y2` A numeric vector of observations, with each value of y2 is assumed to be paired, such as by repeated measures, the corresponding value of y1.

`data` A data frame with `y1` and `y2` as values.

`...` Additional arguments passed to `stats::t.test()`.

Value

A list with class "htest" as returned by `stats::t.test()`.

Examples

```
paired_t_test(y1, y2, data = pairedsleep)
```

pairs_plot *A pairs plot*

Description

This is a wrapper to the `GGally` based pairs plot of a list of variables displayed as scatterplots for pairs of continuous variables, density functions in the diagonal, and boxplots for pairs of continuous and categorical variables. Optionally, a by categorical variable can be provided.

Usage

```
pairs_plot(variables, data, by = NULL)
```

Arguments

`variables` A vector of variable names

`data` The data frame.

`by` An optional variable, usually categorical (factor or character), by which the data are grouped and coloured.

Value

A GGally::ggpairs plot.

Examples

```
# A simple pairs plot
pairs_plot(variables = c("sex_dimorph", "attractive"),
data = faithfulfaces)
# A pairs plot with grouping variable
pairs_plot(variables = c("sex_dimorph", "attractive"),
by = face_sex,
data = faithfulfaces)
```

pairwise_t_test *Pairwise t-test*

Description

This is wrapper to the pairwise.t.test function. The p-value adjustment is "bonferroni" by default. Other possible values are "holm", "hochberg", "hommel", "BH", "BY", "fdr", "none". See stats::p.adjust().

Usage

```
pairwise_t_test(formula, data, p_adj = "bonferroni")
```

Arguments

formula	A two sided formula with one variable on either side, e.g. y ~ x, where the left hand side, dependent, variable is a numeric variable in data and the right hand side, independent, variable is a categorical or factor variable in data.
data	A data frame that contains the dependent and independent variables.
p_adj	The p-value adjustment method (see Description).

Value

An object of class pairwise.htest as returned by stats::pairwise.t.test().

Examples

```
data_df <- dplyr::mutate(vizverb, IV = interaction(task, response))
pairwise_t_test(time ~ IV, data = data_df)
```

re_code *Recode specified values by new values*

Description

Recode specified values by new values

Usage

```
re_code(x, from, to)
```

Arguments

x	A vector, including column of data frame
from	The set of old values to be replaced by new ones
to	The set of new values to replace the old ones

Value

A vector that is the input vector but with old values replaced by new ones.

Examples

```
# Replace any occurrence of 1 and 2 with 101 and 201, respectively
x <- c(1, 2, 3, 4, 5, 1, 2)
re_code(x, from = c(1, 2), to = c(101, 201))
```

scatterplot *A two dimensional scatterplot*

Description

This function is a wrapper around the typical ggplot command to create two dimensional scatterplots, i.e. using geom_point. It provides the option of colouring point by a third variable, one that is usually, though not necessarily categorical. Also, it provides the option of placing the line of best fit on the scatterplot. If points are coloured by a categorical variable, the a different line of best for each value of the categorical variable is provided.

Usage

```
scatterplot(
  x,
  y,
  data,
  by = NULL,
  best_fit_line = FALSE,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

x	A numeric variable in data. Its values are plotted on the x axis.
y	A numeric variable in data. Its values are plotted on the y axis.
data	A data frame with the x and y variables.
by	An optional variable, usually categorical (factor or character), by which the points in the scatterplot are byed and coloured.
best_fit_line	A logical variable indicating if the line of best fit should shown or not.
xlab	The label of the x-axis (defaults to the x variable name).
ylab	The label of the y-axis (defaults to the y variable name).

Value

A ggplot2::ggplot object, which may be modified with further ggplot2 commands.

Examples

```
scatterplot(x = attractive, y = trustworthy, data = faithfulfaces)
scatterplot(x = attractive, y = trustworthy, data = faithfulfaces,
            xlab = 'attractiveness', ylab = 'trustworthiness')
scatterplot(x = attractive, y = trustworthy, data = faithfulfaces,
            by = face_sex)
scatterplot(x = trustworthy, y = faithful, data = faithfulfaces,
            by = face_sex, best_fit_line = TRUE)
```

scatterplot_matrix *Make a scatterplot matrix*

Description

Make a scatterplot matrix

Usage

```
scatterplot_matrix(.data, ..., .by = NULL, .bins = 10)
```

Arguments

`.data`	A data frame
`...`	A comma separated list of tidyselections of columns. This can be as simple as a set of column names.
`.by`	An optional categorical variable by which to group and colour the points.
`.bins`	The number of bins in the histograms on diagonal of matrix.

Value

A `GGally::ggpairs` plot.

Examples

```
data_df <- test_psychometrics %>%
             total_scores(x = starts_with('x_'),
                          y = starts_with('y_'),
                          z = starts_with('z_'))
scatterplot_matrix(data_df, x, y, z)
```

`schizophrenia` *Age of Onset of Schizophrenia Data*

Description

Data on sex differences in the age of onset of schizophrenia.

Usage

```
schizophrenia
```

Format

A data frame with 251 observations on the following 2 variables.

age Age at the time of diagnosis.

gender A categorical variable with values `female` and `male`

Details

A sex difference in the age of onset of schizophrenia was noted by Kraepelin (1919). Subsequently epidemiological studies of the disorder have consistently shown an earlier onset in men than in women. One model that has been suggested to explain this observed difference is known as the subtype model which postulates two type of schizophrenia, one characterised by early onset, typical symptoms and poor premorbid competence, and the other by late onset, atypical symptoms, and good premorbid competence. The early onset type is assumed to be largely a disorder of men and the late onset largely a disorder of women.

Source

This data set was taken from the HSAUR R package. From the description in that package, the original is E. Kraepelin (1919), *Dementia Praecox and Paraphrenia*. Livingstone, Edinburgh.

`selfesteem` *Self-Esteem Score Data for One-way Repeated Measures ANOVA*

Description

The dataset contains 10 individuals' self-esteem score on three time points during a specific diet to determine whether their self-esteem improved.

One-way repeated measures ANOVA can be performed in order to determine the effect of time on the self-esteem score.

This data set was taken from the datarium R package.

Usage

```
data("selfesteem")
```

Format

A data frame with 10 rows and 4 columns.

Examples

```
data(selfesteem)
selfesteem
```

`selfesteem2` *Self Esteem Score Data for Two-way Repeated Measures ANOVA*

Description

Data are the self esteem score of 12 individuals enrolled in 2 successive short-term trials (4 weeks) - control (placebo) and special diet trials.

The self esteem score was recorded at three time points: at the beginning (t1), midway (t2) and at the end (t3) of the trials.

The same 12 participants are enrolled in the two different trials with enough time between trials.

Two-way repeated measures ANOVA can be performed in order to determine whether there is interaction between time and treatment on the self esteem score.

This data set was taken from the datarium R package.

Usage

```
data("selfesteem2")
```

Format

A data frame with 24 rows and 5 columns.

Examples

```
data(selfesteem2)
selfesteem2
```

---

selfesteem2_long *Self Esteem Score Data for Two-way Repeated Measures ANOVA: Long format*

Description

Data are the self esteem score of 12 individuals enrolled in 2 successive short-term trials (4 weeks) - control (placebo) and special diet trials.

The self esteem score was recorded at three time points: at the beginning (t1), midway (t2) and at the end (t3) of the trials.

The same 12 participants are enrolled in the two different trials with enough time between trials.

Two-way repeated measures ANOVA can be performed in order to determine whether there is interaction between time and treatment on the self esteem score.

This data set was converted from the `selfesteem2` data taken from the `datarium` R package.

Usage

```
data("selfesteem2_long")
```

Format

A data frame with 72 rows and 4 columns.

id Unique ID of the person

treatment Binary variable indicating the treatment condition: `Diet` or `ctr`.

time A categorical variable indicating the time of measurement: beginning (`t1`), midway (`t2`) and at the end (`t3`)

score Self-esteem score

Examples

```
data(selfesteem2_long)
selfesteem2_long
```

shapiro_test *Shapiro-Wilk normality test*

Description

This function is a wrapper around stats::shapiro.test(). It implements the Shapiro-Wilk test that tests the null hypothesis that a sample of values is a sample from a normal distribution. Thie function can be applied to single vectors or groups of vectors.

Usage

```
shapiro_test(y, by = NULL, data)
```

Arguments

y	A numeric variable whose normality is being tested.
by	An optional grouping variable
data	A data frame containing y and the by variable

Value

A tibble data frame with one row for each value of the by variable, or one row overall if there is no by variable. For the y variable whose normality is being tested, for each subset of values corresponding to the values of they by variable, or for all values if there is no by variable, return the Shapiro-Wilk statistic, and the corresponding p-value.

Examples

```
shapiro_test(faithful, data = faithfulfaces)
shapiro_test(faithful, by = face_sex, data = faithfulfaces)
```

sum_xna *Descriptive statistics for variables with missing values*

Description

Most descriptive statistic function like base::sum(), base::mean(), stats::median(), etc., do not skip NA values when computing the results and so always return NA if there is at least one NA in the input vector. The NA values can be skipped always by setting the na.rm argument to TRUE. While this is simply to do usually, in some cases, such as when a function is being passed to another function, setting na.rm = TRUE in that function requires creating a new anonymous function. The functions here, which all end in _xna, are wrappers to common statistics functions, but with na.rm = TRUE.

Usage

```
sum_xna(...)

mean_xna(...)

median_xna(...)

iqr_xna(...)

sd_xna(...)

var_xna(...)
```

Arguments

`...`	Arguments to a descriptive statistic function

Value

A numeric vector, usually with one element, that provides the result of a descriptive statistics function applied to a vector after the `NA` values have been removed.

Functions

- `mean_xna`: The arithmetic mean for vectors with missing values.
- `median_xna`: The median for vectors with missing values.
- `iqr_xna`: The interquartile range for vectors with missing values.
- `sd_xna`: The standard deviation for vectors with missing values.
- `var_xna`: The variance for vectors with missing values.

Examples

```
set.seed(10101)
# Make a vector of random numbers
x <- runif(10, min = 10, max = 20)
# Concatenate with a NA value
x1 <- c(NA, x)
sum(x)
sum(x1) # Will be NA
sum_xna(x1) # Will be same as sum(x)
stopifnot(sum_xna(x1) == sum(x))
stopifnot(mean_xna(x1) == mean(x))
stopifnot(median_xna(x1) == median(x))
stopifnot(iqr_xna(x1) == IQR(x))
stopifnot(sd_xna(x1) == sd(x))
stopifnot(var_xna(x1) == var(x))
```

`test_psychometrics` *Psychometrics raw data from testing or demo purposes*

Description

Typical psychometrics raw data files have multiple psychometric variables (scales), each with multiple constituent items. In this data set, there are three psychometric variables, each with 10 constituent items. The variables can be labelled `x`, `y`, and `z`. The constituent items of `x`, `y` and `z` are x_1, x_2 ... x_10, y_1, y_2 ... y_10, z_1, z_2 ... z_10, respectively.

Usage

```
data('test_psychometrics')
```

Format

A data frame with 44 rows and 30 columns

Examples

```
data(test_psychometrics)
test_psychometrics
```

`total_scores` *Calculate the total scores from sets of scores*

Description

Calculate the total scores from sets of scores

Usage

```
total_scores(.data, ..., .method = "mean", .append = FALSE)
```

Arguments

`.data` A data frame with columns to summed or averaged over.

`...` A comma separated set of named tidy selectors, each of which selects a set of columns to which to apply the totalling function.

`.method` The method used to calculate the total. Must be one of "mean", "sum", or "sum_like". The "mean" is the arithmetic mean, skipping missing values. The "sum" is the sum, skipping missing values. The "sum_like" is the arithmetic mean, again skipping missing values, multiplied by the number of elements, including missing values.

`.append` logical If FALSE, just the totals be returned. If TRUE, the totals are appended as new columns to original data frame.

Value

A new data frame with columns representing the total scores.

Examples

```
# Calculate the mean of all items beginning with `x_` and separately all items beginning with `y_`
total_scores(test_psychometrics, x = starts_with('x'), y = starts_with('y'))
# Calculate the sum of all items beginning with `z_` and separately all items beginning with `x_`
total_scores(test_psychometrics, .method = 'sum', z = starts_with('z'), x = starts_with('x_'))
# Calculate the mean of all items from `x_1` to `y_10`
total_scores(test_psychometrics, xy = x_1:y_10)
```

---

tukeyboxplot *A Tukey box-and-whisker plot*

Description

This function is a wrapper around a typical ggplot based box-and-whisker plot, i.e. using geom_boxplot, which implements the Tukey variant of the box-and-whisker plot. The y variable is the outcome variable whose distribution is represented by the box-and-whisker plot. If the x variable is missing, then a single box-and-whisker plot using all values of y is shown. If an x variable is used, this is used an the independent variable and one box-and-whisker plot is provided for each set of y values that correspond to each unique value of x. For this reason, x is usually a categorical variable. If x is a continuous numeric variable, it ideally should have relatively few unique values, so that each value of x corresponds to a sufficiently large set of y values.

Usage

```
tukeyboxplot(
  y,
  x,
  data,
  by = NULL,
  jitter = FALSE,
  box_width = 1/3,
  jitter_width = 1/5,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

y	The outcome variable
x	The optional independent/predictor/grouping variable
data	The data frame with the y and (optionally) x values.
by	An optional variable, usually categorical (factor or character), by which the points in the box-and-whisker plots are grouped and coloured.

`jitter` A logical variable, defaulting to `FALSE`, that indicates if all points in each box-and-whisker plot should be shown as jittered points.

`box_width` The width of box in each box-and-whisker plot. The default used, `box_width = 1/3`, means that boxes will be relatively narrow.

`jitter_width` The width of the jitter relative to box width. For example, set `jitter_width = 1` if you want the jitter to be as wide the box.

`xlab` The label of the x-axis (defaults to the `x` variable name).

`ylab` The label of the y-axis (defaults to the `y` variable name).

Value

A `ggplot2::ggplot` object, which may be modified with further `ggplot2` commands.

Examples

```
# A single box-and-whisker plot
tukeyboxplot(y = time, data = vizverb)
# One box-and-whisker plot for each value of a categorical variable
tukeyboxplot(y = time, x = task, data = vizverb)
# Box-and-whisker plots with jitters
tukeyboxplot(y = time, x = task, data = vizverb,  jitter = TRUE)
# `tukeyboxplot` can be used with a continuous numeric variable too
tukeyboxplot(y = len, x = dose, data = ToothGrowth)
tukeyboxplot(y = len, x = dose, data = ToothGrowth,
             by = supp, jitter = TRUE, box_width = 0.5, jitter_width = 1)
```

`t_test` *Independent samples t-test*

Description

A wrapper to `stats::t.test()` with `var.equal = TRUE`.

Usage

```
t_test(formula, data)
```

Arguments

`formula` A two sided formula with one variable on either side, e.g. y ~ x, where the left hand side, dependent, variable is a numeric variable in `data` and the right hand side, independent, variable is a categorical or factor variable in `data`, and which has only two distinct values.

`data` A data frame that contains the dependent and independent variables.

Value

A list with class "htest" as returned by `stats::t.test()`.

Examples

```
t_test(trustworthy ~ face_sex, data = faithfulfaces)
```

vizverb *Visual versus Verbal Perception and Responses*

Description

An experiment studying the interaction between visual versus perception and visual versus verbal responses.

Usage

```
vizverb
```

Format

A data frame with 80 observations on the following 5 variables.

subject Subject identifying number (`s1` to `s20`)

task Describe a diagram (`visual`) or a sentence (`verbal`)

response Point response (`visual`) or say response (`verbal`)

time Response time (in seconds)

Details

Subjects carried out two kinds of tasks. One task was visual (describing a diagram), and the other was classed as verbal (reading and describing a sentence sentences). They reported the results either by pointing (a "visual" response), or speaking (a verbal response). Time to complete each task was recorded in seconds.

Source

This data set was taken from the `Stats2Data` R package. From the description in that package, the original data appear to have been collected in a Mount Holyoke College psychology class based replication of an experiment by Brooks, L., R. (1968) "Spatial and verbal components of the act of recall," Canadian J. Psych. V 22, pp. 349 - 368.

Index