Package ‘scipub’

May 8, 2020

Title Summarize Data For Scientific Publication

Version 0.0.0.9000

Description This package contains functions for summarizing data for scientific publication. This includes making a 'Table 1' to summarize demographics across groups, correlation tables with significance indicated by stars, and extracting formatted statistical summarizes from simple tests for in-text notation. The package also includes functions for Winsorizing data based on a Z-statistic cutoff.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

Language en-US

VignetteBuilder knitr

URL `https://github.com/dpagliaccio/scipub`

BugReports `https://github.com/dpagliaccio/scipub/issues`

Depends R (>= 3.6.3)

Imports dplyr,
forcats,
purrr,
stats,
stringr,
tibble,
tidyr,
tidyselect

Suggests ggplot2,
htmlTable,
knitr,
rmarkdown

R topics documented:

---

apastat *Format simple statistic test results for scientific publication*

Description

The apastat function summarizes statistic test results scientific publication. This currently will take stats::t.test, stats::cor.test, or stats::lm results as input. The output is intended to be included as in-text parenthetical statistics in publication.

Usage

```
apastat(test, roundN = 2, es = c(TRUE, FALSE), ci = c(TRUE, FALSE), var = NULL)
```

Arguments

test	The stats::t.test, stats::cor.test, or stats::lm object to be formatted.
roundN	The number of decimal places to round all output to (default=2).
es	Include effect side (Cohen's d for t-test or 2-level factor lm variable), default to TRUE.
ci	Include confidence interval of estimate, default to TRUE.
var	Only for lm object, select name of variable to summarize (default=NULL), if NULL, will summarize overall model fit.

Value

Output formatted statistics

Examples

```
apastat(stats::cor.test(psydat$Age, psydat$Height))
apastat(stats::t.test(Height ~ Sex, data = psydat))
apastat(stats::lm(data = psydat, Height ~ Age + Sex))
apastat(stats::lm(data = psydat, Height ~ Age + Sex), var = "Age")
```

correltable *Create correlation table (with stars for significance) for scientific publication*

Description

The correltable function can be used to create correlation table (with stars for significance) for scientific publication This is intended to summarize correlations between (vars) from an input dataset (data). Correlations are based on stats::cor, use and method follow from that function. Stars indicate significance: *p<.05, **p<.01, ***p<.001 For formatting, variables can be renamed, numbers can be rounded, upper or lower triangle only can be selected (or whole matrix), and empty columns/rows can be dropped if using triangles. For more compact columns, variable names can be numbered in the rows and column names will be corresponding numbers. If only cross-correlation between two sets of variables is desired (no correlations within a set of variables), vars2 and var_names can be used. This function will drop any non-numeric variables by default. Requires tidyverse and stats libraries.

Usage

```
correltable(
  data,
  vars = NULL,
  var_names = vars,
  vars2 = NULL,
  var_names2 = vars2,
  method = c("pearson", "spearman"),
  use = c("pairwise", "complete"),
  round_n = 2,
  tri = c("upper", "lower", "all"),
  cutempty = c(FALSE, TRUE),
  colnum = c(FALSE, TRUE),
  html = c(FALSE, TRUE)
)
```

Arguments

data	The input dataset.
vars	A list of the names of variables to correlate, e.g. c("Age","height","WASI"), if NULL, all variables in data will be used.
var_names	An optional list to rename the vars colnames in the output table, e.g. c("Age (years)","Height (inches)","IQ"). Must match vars in length. If not supplied, vars will be printed as is.
vars2	If cross-correlation between two sets of variables is desired, add a second list of variables to correlate with vars; Overrides tri, cutempty, and colnum.
var_names2	An optional list to rename the vars2 colnames in the output table If not supplied, vars2 will be printed as is.
method	Type of correlation to calculate c("pearson", "spearman"), based on stats::cor, default = "pearson".
use	Use pairwise.complete.obs or restrict to complete cases c("pairwise", "complete"), based on stats::cor, default = "pairwise".

`round_n` The number of decimal places to round all output to (default=2).

`tri` Select output formatting c("upper", "lower","all"); KEEP the upper triangle, lower triangle, or all values, default ="upper.

`cutempty` If keeping only upper/lower triangle with `tri`, cut empty row/column, default=FALSE.

`colnum` For more concise column names, number row names and just use corresponding numbers as column names, default=FALSE, if TRUE overrides cutempty.

`html` Format as html in viewer or not (default=F, print in console), needs library(htmlTable) installed.

Value

Output Table 1

Examples

```
correltable(data = psydat)
correltable(
  data = psydat, vars = c("Age", "Height", "iq"),
  tri = "lower", html = TRUE
)
correltable(
  data = psydat, vars = c("Age", "Height", "iq"),
  var_names = c("Age (months)", "Height (inches)", "IQ"),
  tri = "upper", colnum = TRUE, html = TRUE
)
correltable(
  data = psydat, vars = c("Age", "Height", "iq"),
  var_names = c("Age (months)", "Height (inches)", "IQ"),
  vars2 = c("depressT", "anxT"),
  var_names2 = c("Depression T", "Anxiety T"), html = TRUE
)
```

`FullTable1` *Create Table1 of group summary with stats for scientific publication*

Description

The `FullTable1` function can be used to create a Table1 for scientific publication. This is intended to summarize demographic and other variables (`vars`) split by a grouping variable (`strata`) from an input dataset (`data`). Continuous variables will be summarized as mean (SD) and tested across groups using t-test or ANOVA (for 3+ level `strata`). Categorical variables will be summarized as N (%) and tested across groups as chi-squared. Effect sizes for group differences will be calculated as Cohen's d, partial eta-squared, Odds Ratio, Cramer's V depending on the test. Requires `tidyverse` and `stats` libraries.

Usage

```
FullTable1(
  data,
  strata = NULL,
  vars = NULL,
  var_names = vars,
  factor_vars = NULL,
  round_n = 2,
  es_col = c(TRUE, FALSE),
  p_col = c(TRUE, FALSE),
  stars = c("col", "name", "stat", "none"),
  html = c(FALSE, TRUE)
)
```

Arguments

`data` The input dataset (will be converted to tibble).

`strata` The grouping variable of interest (converted to factor), if NULL will make one column table.

`vars` A list of variables to summarize, e.g. c("Age","sex","WASI").

`var_names` An optional list to rename the variable colnames in the output table, e.g. c("Age (years)","Sex","IQ"). Must match `vars` in length. If not supplied, `vars` will be printed as is.

`factor_vars` An optional list of variables from `vars` to use as class factor, e.g. c("sex"). Note that any character, factor, or logical class variables will be summarized as categorical by default.

`round_n` The number of decimal places to round output to (default=2).

`es_col` Include a column for effect size of group difference? (default=T).

`p_col` Include a column for p-value of group difference? (default=TRUE).

`stars` Where to include stars indicating significance of group differences. Options: "col"=separate column (default), "name"= append to variable name, "stat"= append to group difference statistic, "none" for no stars.

`html` Format as html in viewer or not (default=FALSE, print in console), needs library(htmlTable) installed.

Value

Output Table 1

Examples

```
FullTable1(
  data = psydat,
  vars = c("Age", "Height", "depressT"), strata = "Sex"
)
FullTable1(
  data = psydat,
  vars = c("Age", "Height", "depressT"), strata = "Sex"
)
FullTable1(
  data = psydat, vars = c("Age", "Sex", "Height", "depressT"),
  var_names = c("Age (months)", "Sex", "Height (inches)", "Depression T"),
  strata = "Income", stars = "name", p_col = FALSE
)
```

psydat — *Sample demographic and clinical data for 5,000 children*

Description

An example dataset containing demographic and clinical data for 5,000 children. The variables are as follows:

Usage

```
data(psydat)
```

Format

A data frame with 5000 rows and 7 variables:

Age age in months (107.2–136.4)

Sex biological sex, 4 missing value (M, F)

Income reported family income, 404 missing values (<50K, >=100K, >=50K&<100K)

Height height in inches, 7 missing values (36.05–84.51)

iq cognition test, 179 missing values (34.86–222.99)

depressT depression symptom severity T-score, 8 missing values (48.53–91.32)

anxT anxiety symptom severity T-score, 8 missing values (48.76–93,67)

winsorZ — *Winsorize outliers based on z-score cutoff to next most extreme non-outlier value*

Description

The `winsorZ` function identifies outliers based on Z-score cutoff and replaces with the next most extreme non-outlier value. This involves z-scoring the variable and identifying/replacing any cases beyond the z-score threshold. The `winsorZ_find` function is an optional companion to flag any Z-score outliers to tally as needed.

Usage

```
winsorZ(x, zbound = 3)
```

Arguments

`x`	The input variable to Winsorize.
`zbound`	The Z-score cutoff (default=3, i.e. outliers are Z>3 | Z<-3).

Value

Output Winsorized variable

Examples

```
winsorZ(psydat$iq)
## Not run:
psydat %>%
  dplyr::select(c(iq, anxT)) %>%
  map(winsorZ)
psydat %>% mutate_at(c("iq", "anxT"), list(~ winsorZ(.)))
psydat %>% mutate_if(is.double, list(~ winsorZ(.)))

## End(Not run)
```

---

winsorZ_find — *Identify outliers based on z-score cutoff that are Winsorized by the* winsorZ *function*

Description

The winsorZ_find function is an optional companion to the winsorZ function. The winsorZ function identifies Z-score outliers and replaces with the next most extreme non-outlier value. The winsorZ_find function finds/identifies these Z-score outliers (outliers=1, non-outliers=0).

Usage

```
winsorZ_find(x, zbound = 3)
```

Arguments

x	The input variable to check for Z-score outliers.
zbound	The Z-score cutoff (default=3, i.e. outliers are Z>3 | Z<-3).

Value

Output logical variable of Z-score outliers

Examples

```
summary(winsorZ_find(psydat$iq))
## Not run:
psydat %>% mutate_at(c("iq", "anxT"), list(out = ~ winsorZ_find(.)))

## End(Not run)
```

Index