

Package ‘wrappedtools’

July 10, 2025

Type Package

Title Useful Wrappers Around Commonly Used Functions

Description The main functionalities of 'wrappedtools' are:
adding backticks to variable names; rounding to desired precision
with special case for p-values;
selecting columns based on pattern and storing their position, name,
and backticked name; computing and formatting of descriptive statistics
(e.g. mean $\pm$ SD), comparing groups and creating publication-ready tables with
descriptive statistics and p-values; creating specialized plots for
correlation matrices. Functions were mainly written for my own daily work or
teaching, but may be of use to others as well.

Version 0.9.8

Date 2025-07-07

Maintainer Andreas Busjahn <andreas@busjahn.net>

License GPL-3

Encoding UTF-8

Imports stats, boot, knitr, coin, utils, dplyr, tidyr, forcats, purrr,
glue, rlang, stringr, ggplot2, tibble, tidyr, kableExtra,
lifecycle, broom, rlist, DescTools, flextable, nortest

Depends R (>= 4.2)

RoxygenNote 7.3.2

LazyData true

VignetteBuilder knitr

Suggests rmarkdown, testthat, ggrepel

URL <https://github.com/abusjahn/wrappedtools>

BugReports <https://github.com/abusjahn/wrappedtools/issues>

NeedsCompilation no

Author Andreas Busjahn [cre, aut] (ORCID:
<<https://orcid.org/0000-0001-9650-6919>>),
Franziska Eidloth [aut],
Bilal Asser [aut]

Repository CRAN

Date/Publication 2025-07-10 16:10:02 UTC

Contents

bt	3
cat_desc_stats	3
cat_desc_table	5
cn	6
ColSeeker	6
compare2numvars	7
compare2qualvars	9
compare_n_numvars	10
compare_n_qualvars	11
cortestR	13
detect_outliers	14
eGFR	14
faketrial	15
FindVars	16
flex2rmd	17
formatP	17
ggcormat	18
glmCI	19
identical_cols	20
ksnormal	21
label_outliers	22
logrange_1	23
markSign	24
meansd	24
meanse	26
mean_cl_boot	26
medianse	27
median_cl_boot	28
median_cl_boot_gg	29
median_quart	29
pairwise_fisher_test	31
pairwise_ordcat_test	32
pairwise_t_test	33
pairwise_wilcox_test	34
pdf_kable	35
plot_LB	36
plot_MM	37
print_kable	38
roundR	39
SEM	40
se_median	40
surprisal	41

Usage

```
cat_desc_stats(
  source = NULL,
  separator = " ",
  return_level = TRUE,
  ndigit = 0,
  groupvar = NULL,
  singleline = FALSE,
  percent = TRUE,
  prettynum = FALSE,
  .german = FALSE,
  quelle = NULL
)
```

Arguments

source	Data for computation. Previously "quelle".
separator	delimiter between results per level, preset as ' '.
return_level	Should levels be reported?
ndigit	Digits for rounding of relative frequencies.
groupvar	Optional grouping factor.
singleline	Put all group levels in a single line?
percent	Logical, add percent-symbol after relative frequencies?
prettynum	logical, apply prettyNum to results?
.german	logical, should "." and "," be used as bigmark and decimal? Sets prettynum to TRUE.
quelle	deprecated, retained for compatibility, use 'source' instead.

Value

Structure depends on parameter `return_level`: if `FALSE` than a tibble with descriptives, otherwise a list with two tibbles with levels of factor and descriptives. If parameter `singleline` is `FALSE` (default), results for each factor level is reported in a separate line, otherwise they are pasted. Number of columns for result tibbles is one or number of levels of the additional grouping variable.

Examples

```
cat_desc_stats(mtcars$gear)
cat_desc_stats(mtcars$gear, return_level = FALSE)
cat_desc_stats(mtcars$gear, groupvar = mtcars$am)
cat_desc_stats(mtcars$gear, groupvar = mtcars$am, singleline = TRUE)
```

cat_desc_table	<i>Compute absolute and relative frequencies for a table.</i>
----------------	---

Description

cat_desc_table computes absolute and relative frequencies for categorical data with a number of formatting options.

Usage

```
cat_desc_table(  
  data,  
  desc_vars,  
  round_desc = 2,  
  singleline = FALSE,  
  spacer = " ",  
  indenter = ""  
)
```

Arguments

data	name of data set (tibble/data.frame) to analyze.
desc_vars	vector of column names for dependent variables.
round_desc	number of significant digits for rounding of descriptive stats.
singleline	Put all group levels in a single line?
spacer	Text element to indent levels and fill empty cells, defaults to " ".
indenter	Optional text to indent factor levels

Value

A tibble with variable names and descriptive statistics.

Examples

```
cat_desc_table(  
  data = mtcars, desc_vars = c("gear", "cyl", "carb")  
)  
  
cat_desc_table(  
  data = mtcars, desc_vars = c("gear", "cyl", "carb"), singleline = TRUE  
)
```

cn	<i>Shortcut for colnames()</i>
----	--------------------------------

Description

cn lists column names, by default for variable rawdata.

Usage

```
cn(data = rawdata)
```

Arguments

data	Data structure to read column names from.
------	---

Value

Character vector with column names.

Examples

```
cn(mtcars)
```

ColSeeker	<i>Find numeric index and names of columns based on class(es) and patterns</i>
-----------	--

Description

ColSeeker looks up colnames (by default for tibble rawdata) based on type and parts of names, using regular expressions. Be warned that special characters as e.g. [(need to be escaped or replaced by \. Exclusion rules may be specified as well.

Usage

```
ColSeeker(
  data = rawdata,
  namepattern = ".",
  varclass = NULL,
  exclude = NULL,
  excludeclass = NULL,
  casesensitive = TRUE,
  returnclass = FALSE
)
```

Arguments

data	tibble or data.frame, where columns are to be found; by default rawdata
namepattern	Vector of pattern to look for.
varclass	Vector, only columns of defined class(es) are returned
exclude	Vector of pattern to exclude from found names.
excludeclass	Vector, exclude columns of specified class(es)
casesensitive	Logical if case is respected in matching (default FALSE: a<>A)
returnclass	Logical if classes should be included in output

Value

A list with index, names, backticked names, and count; optionally the classes as well

Examples

```
ColSeeker(data = mtcars, namepattern = c("^c", "g"))
ColSeeker(data = mtcars, namepattern = c("^c", "g"), exclude = "r")
assign("rawdata", mtcars)
ColSeeker(namepattern = c("^c", "g"), varclass="numeric")
num_int_data <- data.frame(num1=rnorm(10), num2=runif(10), int1=1:10, int2=11:20)
ColSeeker(num_int_data, varclass="numeric") # integers are not found
ColSeeker(num_int_data, varclass=c("numeric","integer"))
```

compare2numvars

Comparison for columns of numbers for 2 groups

Description

compare2numvars computes either [t_var_test](#) or [wilcox.test](#), depending on parameter gaussian. Descriptive statistics, depending on distribution, are reported as well.

Usage

```
compare2numvars(
  data,
  dep_vars,
  indep_var,
  gaussian,
  round_p = 3,
  round_desc = 2,
  range = FALSE,
  rangesep = " ",
  pretext = FALSE,
  mark = FALSE,
  n = FALSE,
  add_n = FALSE,
```

```

    singleline = TRUE,
    indenter = "  ",
    ci = FALSE,
    n_boot = 10^3
  )

```

Arguments

<code>data</code>	name of dataset (tibble/data.frame) to analyze.
<code>dep_vars</code>	vector of column names for independent variables.
<code>indep_var</code>	name of grouping variable, has to translate to 2 groups. If more levels are encountered, an error is produced.
<code>gaussian</code>	logical specifying normal or ordinal values.
<code>round_p</code>	level for rounding p-value.
<code>round_desc</code>	number of significant digits for rounding of descriptive stats.
<code>range</code>	include min/max?
<code>rangesep</code>	text between statistics and range or other elements.
<code>pretext</code>	for function formatP .
<code>mark</code>	for function formatP .
<code>n</code>	create columns for n per group?
<code>add_n</code>	add n to descriptive statistics. Will automatically be set to TRUE, if <code>singleline = FALSE</code> and <code>n = TRUE</code> to keep it for the long table format.
<code>singleline</code>	Put all computed stats in a single line (default) or below each other.
<code>indenter</code>	Optional text element to indent descriptivestats when using <code>singleline = FALSE</code> . Defaults to " ".
<code>ci</code>	Computes lower and upper confidence limits for the estimated mean/median, based on bootstrapping.
<code>n_boot</code>	Number of bootstrap samples for confidence limits.

Value

A tibble with variable names, descriptive statistics, and p-value, number of rows is number of `dep_vars`.

Examples

```

# Assuming Normal distribution:
compare2numvars(
  data = mtcars, dep_vars = c("wt", "mpg", "qsec"), indep_var = "am",
  gaussian = TRUE
)
compare2numvars(
  data = mtcars, dep_vars = c("wt", "mpg", "qsec"), indep_var = "am",
  gaussian = TRUE, singleline = FALSE
)

```

```
# Ordinal scale:
compare2numvars(
  data = mtcars, dep_vars = c("wt", "mpg", "qsec"), indep_var = "am",
  gaussian = FALSE, add_n = TRUE, range = TRUE
)
# If dependent variable has more than 2 levels, consider fct_lump:
mtcars |>
  dplyr::mutate(gear = factor(gear) |> forcats::fct_lump_n(n = 1)) |>
  compare2numvars(dep_vars = "wt", indep_var = "gear", gaussian = TRUE)
```

compare2qualvars	<i>Comparison for columns of factors for 2 groups</i>
------------------	---

Description

compare2qualvars computes [fisher.test](#) with simulated p-value and descriptive statistics for a group of categorical dependent variables.

Usage

```
compare2qualvars(
  data,
  dep_vars,
  indep_var,
  round_p = 3,
  round_desc = 2,
  pretext = FALSE,
  mark = FALSE,
  singleline = FALSE,
  spacer = " ",
  linebreak = "\n",
  p_subgroups = FALSE
)
```

Arguments

data	name of data set (tibble/data.frame) to analyze.
dep_vars	vector of column names for dependent variables.
indep_var	name of grouping variable, has to translate to 2 groups.
round_p	level for rounding p-value.
round_desc	number of significant digits for rounding of descriptive stats.
pretext	for function formatP .
mark	for function formatP .
singleline	Put all group levels in a single line?
spacer	Text element to indent levels and fill empty cells, defaults to " ".
linebreak	place holder for newline.
p_subgroups	test subgroups by recoding other levels into other, default is not to do this.

Value

A tibble with variable names, descriptive statistics, and p-value, number of rows is number of dep_vars.

Examples

```
compare2qualvars(
  data = mtcars, dep_vars = c("gear", "cyl", "carb"), indep_var = "am",
  spacer = " "
)
compare2qualvars(
  data = mtcars, dep_vars = c("gear", "cyl", "carb"), indep_var = "am",
  spacer = " ", singleline = TRUE
)
compare2qualvars(
  data = mtcars, dep_vars = c("gear", "cyl", "carb"), indep_var = "am",
  spacer = " ", p_subgroups = TRUE
)
```

compare_n_numvars	<i>Comparison for columns of Gaussian or ordinal measures for n groups</i>
-------------------	--

Description

Some names were changed in August 2022, to reflect the update of the function to handle ordinal data using non-parametric equivalents.

Usage

```
compare_n_numvars(
  .data = rawdata,
  dep_vars,
  indep_var,
  gaussian,
  round_desc = 2,
  range = FALSE,
  rangeseq = " ",
  pretext = FALSE,
  mark = FALSE,
  round_p = 3,
  add_n = FALSE
)
```

Arguments

.data	name of dataset (tibble/data.frame) to analyze, defaults to rawdata.
dep_vars	vector of column names.

indep_var	name of grouping variable.
gaussian	Logical specifying normal or ordinal indep_var (and chooses comparison tests accordingly)
round_desc	number of significant digits for rounding of descriptive stats.
range	include min/max?
rangesep	text between statistics and range or other elements.
pretext, mark	for function formatP.
round_p	level for rounding p-value.
add_n	add n to descriptive statistics?

Value

A list with elements "results": tibble with descriptive statistics, p-value from ANOVA/Kruskal-Wallis test, p-values for pairwise comparisons, significance indicators, and descriptives pasted with significance. "raw": nested list with output from all underlying analyses.

Examples

```
# Usually, only the result table is relevant:
compare_n_numvars(
  .data = mtcars, dep_vars = c("wt", "mpg", "hp"),
  indep_var = "cyl",
  gaussian = TRUE
)$results
# For a report, result columns may be filtered as needed:
compare_n_numvars(
  .data = mtcars, dep_vars = c("wt", "mpg", "hp"),
  indep_var = "cyl",
  gaussian = FALSE
)$results |>
  dplyr::select(Variable, `cyl 4 fn`:`cyl 8 fn`, multivar_p)
```

compare_n_qualvars	<i>Comparison for columns of factors for more than 2 groups with post-hoc</i>
--------------------	---

Description

Comparison for columns of factors for more than 2 groups with post-hoc

Usage

```
compare_n_qualvars(
  data,
  dep_vars,
  indep_var,
```

```

round_p = 3,
round_desc = 2,
pretext = FALSE,
mark = FALSE,
singleline = FALSE,
spacer = " ",
linebreak = "\n",
prettynum = FALSE
)

```

Arguments

data	name of data set (tibble/data.frame) to analyze.
dep_vars	vector of column names.
indep_var	name of grouping variable.
round_p	level for rounding p-value.
round_desc	number of significant digits for rounding of descriptive stats
pretext	for function formatP
mark	for function formatP
singleline	Put all group levels in a single line?
spacer	Text element to indent levels, defaults to " ".
linebreak	place holder for newline.
prettynum	Apply prettyNum to results?

Value

A tibble with variable names, descriptive statistics, and p-value of [fisher.test](#) and [pairwise_fisher_test](#), number of rows is number of dep_vars.

Examples

```

# Separate lines for each factor level:
compare_n_qualvars(
  data = mtcars, dep_vars = c("am", "cyl", "carb"), indep_var = "gear",
  spacer = " "
)
# All levels in one row but with linebreaks:
compare_n_qualvars(
  data = mtcars, dep_vars = c("am", "cyl", "carb"), indep_var = "gear",
  singleline = TRUE
)
# All levels in one row, separated by ";":
compare_n_qualvars(
  data = mtcars, dep_vars = c("am", "cyl", "carb"), indep_var = "gear",
  singleline = TRUE, linebreak = "; "
)

```

cortestR	<i>Correlations with significance</i>
----------	---------------------------------------

Description

cortestR computes correlations and their significance level based on [cor.test](#). Coefficients and p-values may be combined or reported separately.

Usage

```
cortestR(  
  cordata,  
  method = "pearson",  
  digits = 3,  
  digits_p = 3,  
  sign_symbol = TRUE,  
  split = FALSE,  
  space = ""  
)
```

Arguments

cordata	data frame or matrix with rawdata.
method	as in cor.test.
digits	rounding level for estimate.
digits_p	rounding level for p value.
sign_symbol	If true, use significance indicator instead of p-value.
split	logical, report correlation and p combined (default) or split in list.
space	character to fill empty upper triangle.

Value

Depending on parameters split and sign_symbol, either a single data frame with coefficient and p-values or significance symbols or a list with two data frames.

Examples

```
# with defaults  
cortestR(mtcars[, c("wt", "mpg", "qsec")], split = FALSE, sign_symbol = TRUE)  
# separate coefficients and p-values  
cortestR(mtcars[, c("wt", "mpg", "qsec")], split = TRUE, sign_symbol = FALSE)
```

detect_outliers	<i>Find outliers based on IQR</i>
-----------------	-----------------------------------

Description**[Experimental]**

detect_outliers computes IQR and finds outliers. It gives the same results as geom_boxplot and thus differs slightly from boxplot.stats.

Usage

```
detect_outliers(x, coef = 1.5)
```

Arguments

x	numeric vector.
coef	coefficient for boxplot.stats, defaults to 1.5.

Value

A list with elements positions and outliers as numeric vectors.

Examples

```
detect_outliers(rnorm(100))
```

eGFR	<i>Estimation of glomerular filtration rate (eGFR) based on sex, age, and either serum creatinine and/or cystatin C</i>
------	---

Description**[Experimental]**

eGFR computes eGFR according to different rules (see references).

Usage

```
eGFR(data, age_var = "age", sex_var = "sex", crea_var = NULL, cys_var = NULL)
```

Arguments

data	name of data set (tibble/data.frame) to analyze.
age_var	name of column with patient age in years, default=age.
sex_var	name of column with sex, assumed as female and male.
crea_var	name of column with creatinine in mg/dl. If not available, leave as NULL.
cys_var	name of column with cystatin C in mg/l. If not available, leave as NULL.

Value

A list with 3 elements:

eGFR_crea

eGFR_cystatin

eGFR_creatinine_cystatin

References

<https://www.kidney.org/content/ckd-epi-creatinine-cystatin-equation-2021>

<https://www.kidney.org/content/ckd-epi-creatinine-equation-2021>

<https://www.kidney.org/content/ckd-epi-cystatin-c-equation-2012>

faketrial

Results from a simulated clinical trial with interaction effects.

Description

A dataset containing physiological data, biomarkers, and categorical data.

Usage

faketrial

Format

A tibble with 300 rows and 24 variables:

Sex Sex of animal, factor with levels 'female', 'male'

Agegroup Factor with levels 'young', 'middle', 'old'

Treatment Factor with levels 'sham', 'OP'

HR Heart rate

sysRR, diaRR Systolic and diastolic blood pressure

Med xxx Pseudo-medications, factors with levels 'y', 'n'

Biomarker x units Biomarkers with log-normal distribution

Responder factor yes/no, systolic blood pressure ≥ 120 ?

FindVars

*Find numeric index and names of columns based on patterns***Description****[Superseded]**

Function [ColSeeker](#) extends this by adding class-checks.

FindVars looks up colnames (by default for data-frame rawdata) based on parts of names, using regular expressions. Be warned that special characters as e.g. [(need to be escaped or replaced by \. Exclusion rules may be specified as well. New function [ColSeeker\(\)](#) extends this by adding class-checks.

Usage

```
FindVars(
  varnames,
  allnames = colnames(rawdata),
  exact = FALSE,
  exclude = NA,
  casesensitive = TRUE,
  fixed = FALSE,
  return_symbols = FALSE
)
```

Arguments

varnames	Vector of pattern to look for.
allnames	Vector of values to detect pattern in; by default: colnames(rawdata).
exact	Partial matching or exact only (adding ^ and \$)?
exclude	Vector of pattern to exclude from found names.
casesensitive	Logical if case is respected in matching (default FALSE: a<>A)
fixed	Logical, match as is, argument is passed to grep() .
return_symbols	Should names be reported as symbols additionally? (Default FALSE)

Value

A list with index, names, backticked names, and symbols

Examples

```
FindVars(varnames = c("^c", "g"), allnames = colnames(mtcars))
FindVars(varnames = c("^c", "g"), allnames = colnames(mtcars), exclude = "r")
```

flex2rmd	<i>Transform flextable to rmd if non-interactive</i>
----------	--

Description

flex2rmd takes a flextable and returns a markdown table if not in an interactive session

Usage

```
flex2rmd(ft)
```

Arguments

ft a flextable

Value

either a markdown table or the flextable

formatP	<i>Re-format p-values, avoiding rounding to 0 and adding surprisal if requested</i>
---------	---

Description

formatP simplifies p-values by rounding to the maximum of p or a predefined level. Optionally < or = can be added, as well as symbols according to significance level.

Usage

```
formatP(
  pIn,
  ndigits = 3,
  textout = TRUE,
  pretext = FALSE,
  mark = FALSE,
  german_num = FALSE,
  add.surprisal = FALSE,
  sprecision = 1
)
```

Arguments

pIn	A numeric vector or matrix with p-values.
ndigits	Number of digits (default=3).
textout	Cast output to character (default=TRUE)?
pretext	Should = or < be added before p (default=FALSE)?
mark	Should significance level be added after p (default=FALSE)?
german_num	change dot (default) to comma?
add.surprisal	Add surprisal aka Shannon information to p-value (default=FALSE)?
sprecision	Rounding level for surprisal (default=1).

Value

vector or matrix (depending on type of pIn) with type character (default) or numeric, depending on parameter textout

Examples

```
formatP(0.012345)
formatP(0.012345, add.surprisal = TRUE)
formatP(0.012345, ndigits = 4)
formatP(0.000122345, ndigits = 3, pretext = TRUE)
```

ggcormat

Print graphical representation of a correlation matrix.

Description

ggcormat makes the same correlation matrix as [cortestR](#) and graphically represents it in a plot

Usage

```
ggcormat(
  cor_mat,
  p_mat = NULL,
  method = "Correlation",
  title = "",
  maxpoint = 2.1,
  textsize = 5,
  axistextsize = 2,
  titlesize = 3,
  breaklabels = NULL,
  lower_only = TRUE,
  .low = "blue3",
  .high = "red2",
  .legendtitle = NULL
)
```

Arguments

<code>cor_mat</code>	correlation matrix as produced by <code>cor</code> .
<code>p_mat</code>	Optional matrix of p-values; if provided, this is used to define size of dots rather than absolute correlation.
<code>method</code>	text specifying type of correlation.
<code>title</code>	plot title.
<code>maxpoint</code>	maximum for <code>scale_size_manual</code> , may need adjustment depending on plot size.
<code>textsize</code>	for theme text.
<code>axistextsize</code>	relative text size for axes.
<code>titlesize</code>	as you already guessed, relative text size for title.
<code>breaklabels</code>	currently not used, intended for <code>str_wrap</code> .
<code>lower_only</code>	should only lower triangle be plotted?
<code>.low</code>	Color for heatmap.
<code>.high</code>	Color for heatmap.
<code>.legendtitle</code>	Optional name for color legend.

Value

A ggplot object, allowing further styling.

Examples

```
coeff_pvalues <- cortestR(mtcars[, c("wt", "mpg", "qsec", "hp")],
  split = TRUE, sign_symbol = FALSE
)
# focus on coefficients:
ggcormat(cor_mat = coeff_pvalues$corout, maxpoint = 5)
# size taken from p-value:
ggcormat(
  cor_mat = coeff_pvalues$corout,
  p_mat = coeff_pvalues$pout, maxpoint = 5)
```

 glmCI

Confidence interval for generalized linear models

Description

`glm_CI` computes and formats CIs for `glm`.

Usage

```
glmCI(model, min = .01, max = 100, ciseq = '\U000022ef', ndigit=2)
```

Arguments

model	Output from glm .
min, max	Lower and upper limits for CIs, useful for extremely wide CIs.
cisep	Separator between CI values.
ndigit	rounding level.

Value

A list with coefficient, CIs, and pasted coef([CIs]).

Examples

```
glm_out <- glm(am ~ mpg, family = binomial, data = mtcars)
glmCI(glm_out)
```

identical_cols	<i>Find and optionally remove identical columns in a data frame.</i>
----------------	--

Description

This function identifies columns with identical values in a data frame and provides options to remove them, clean column names, and print the duplicated groups. It also includes an interactive mode where the user can choose to remove all, some, or none of the duplicated columns.

Usage

```
identical_cols(
  df,
  interactive = TRUE,
  remove_duplicates = TRUE,
  clean_names = TRUE,
  print_duplicates = TRUE
)
```

Arguments

df	A data frame or tibble.
interactive	Logical. If TRUE, the function prompts the user to choose how to handle duplicated columns. Defaults to TRUE.
remove_duplicates	Logical. If TRUE, removes duplicated columns. Defaults to TRUE.
clean_names	Logical. If TRUE, cleans column names by removing trailing "..." followed by digits. Defaults to TRUE.
print_duplicates	Logical. If TRUE, prints the groups of duplicated columns. Defaults to TRUE.

Value

A data frame with optionally removed and renamed columns.

Examples

```
library(tibble)

dummy <- tibble(
  A...1 = rnorm(10),
  A...2 = A...1,
  C = sample(letters, 10),
  A...4 = A...1,
  E = sample(1:10, 10),
  `F` = C
)

# Example usage:
identical_cols(dummy) # Interactive removal
identical_cols(dummy, remove_duplicates = FALSE) # Find identical columns only
identical_cols(dummy, print_duplicates = FALSE) # Interactive removal, no print
identical_cols(dummy, clean_names = FALSE) # Interactive removal, no clean names
identical_cols(dummy, interactive = FALSE) #Non interactive removal of all duplicates.
```

ksnormal

*Kolmogorov-Smirnov-Test against Normal distribution***Description**

ksnormal is a convenience function around [ks.test](#), testing against Normal distribution. If less than 2 values are provided, NA is returned.

Usage

```
ksnormal(x, lillie = TRUE)
```

Arguments

x	Vector of data to test.
lillie	Logical, should the Lilliefors test be used? Defaults to TRUE

Value

p.value from [ks.test](#).

Examples

```
# original ks.test:
ks.test(
  x = mtcars$wt, pnorm, mean = mean(mtcars$wt, na.rm = TRUE),
  sd = sd(mtcars$wt, na.rm = TRUE)
)
# wrapped version:
ksnormal(x = mtcars$wt, lillie = FALSE)
```

label_outliers	<i>Add labels to outliers in boxplot/beeswarm.</i>
----------------	--

Description

[Experimental]

label_outliers adds a text_repel layer to an existing ggplot object. It is intended to be used with boxplots or beeswarm plots. Faceting will result in separate computations for outliers. It requires the ggrepel package.

Usage

```
label_outliers(
  plotbase,
  labelvar = NULL,
  coef = 1.5,
  nudge_x = 0,
  nudge_y = 0,
  color = "darkred",
  size = 3,
  hjust = 0,
  face = "bold"
)
```

Arguments

plotbase	ggplot object to add labels to.
labelvar	variable to use as label. If NULL, rownames or rownumbers are used.
coef	coefficient for boxplot.stats, defaults to 1.5.
nudge_x	nudge in x direction, defaults to 0.
nudge_y	nudge in y direction, defaults to 0.
color	color of labels, defaults to darkred.
size	size of labels, defaults to 3.
hjust	horizontal justification of labels, defaults to 0.
face	font face of labels, defaults to bold.

Value

A ggplot object, allowing further styling.

logrange_1

Predefined sets of labels for plots with log-scaled axes

Description

logrange_1 returns a vector for log-labels at .1, 1, 100, 1000 ...

Usage

```
logrange_1
```

```
logrange_5
```

```
logrange_123456789
```

```
logrange_12357
```

```
logrange_15
```

Format

An object of class `numeric` of length 41.

An object of class `numeric` of length 738.

An object of class `numeric` of length 369.

An object of class `numeric` of length 205.

An object of class `numeric` of length 82.

Value

numeric vector

numeric vector

Functions

- logrange_5: vector for log-labels at 1.0, 1.5, 2.0, 2.5 ... 10, 15, 20, 25 ...
- logrange_123456789: vector for log-labels at 1, 2, 3 ... 9, 10, 20, 30 ... 90, 100 ...
- logrange_12357: vector for log-labels at 1, 2, 3, 5, 7, 10, 20, 30, 50, 70 ...
- logrange_15: vector for log-labels at 1, 5, 10, 50 ...

Examples

```
ggplot2::ggplot(mtcars) +
  ggplot2::aes(wt, mpg) +
  ggplot2::geom_point() +
  ggplot2::scale_y_log10(breaks = logrange_5)
ggplot2::ggplot(mtcars) +
  ggplot2::aes(wt, mpg) +
  ggplot2::geom_point() +
  ggplot2::scale_y_log10(breaks = logrange_123456789)
```

markSign	<i>Convert significance levels to symbols</i>
----------	---

Description

markSign returns the symbol associated with a significance level.

Usage

```
markSign(SignIn, plabel = c("n.s.", "+", "*", "**", "***"))
```

Arguments

- SignIn A single p-value.
- plabel A translation table, predefined with the usual symbols.

Value

factor with label as defined in plabel.

Examples

```
markSign(0.012)
```

meansd	<i>Compute mean and sd and put together with the $\pm$ symbol.</i>
--------	---

Description

Compute mean and sd and put together with the $\pm$ symbol.

Usage

```
meansd(
  x,
  roundDig = 2,
  drop0 = FALSE,
  groupvar = NULL,
  range = FALSE,
  rangeseq = " ",
  add_n = FALSE,
  .german = FALSE,
  ci = FALSE,
  nrepl = 10^3,
  singleline = TRUE
)
```

Arguments

<code>x</code>	Data for computation.
<code>roundDig</code>	Number of relevant digits for roundR.
<code>drop0</code>	Should trailing zeros be dropped?
<code>groupvar</code>	Optional grouping variable for subgroups.
<code>range</code>	Should min and max be included in output?
<code>rangeseq</code>	How should min/max be separated from mean+-sd?
<code>add_n</code>	Should n be included in output?
<code>.german</code>	Logical, should "." and "," be used as bigmark and decimal?
<code>ci</code>	Should bootstrap based 95% confidence interval be computed?
<code>nrepl</code>	Number of bootstrap replications, defaults to 1000.
<code>singleline</code>	Put all descriptive stats in a single element (default) or below each other. <code>singleline = FALSE</code> sets <code>ci</code> and <code>add_n</code> as TRUE

Value

Either character vector with mean $\pm$ SD and additional results (`singleline`), or matrix with rows for n, mean with CI, and SD, and optionally range.

Examples

```
# basic usage of meansd
meansd(x = mtcars$wt)
# with additional options
meansd(x = mtcars$wt, groupvar = mtcars$am, add_n = TRUE)
meansd(x = mtcars$wt, groupvar = mtcars$am, add_n = TRUE, ci = TRUE, singleline = FALSE)
```

meanse	<i>Compute mean and standard error of mean and put together with the $\pm$ symbol.</i>
--------	---

Description

meanse computes SEM based on Standard Deviation/square root(n)

Usage

```
meanse(x, mult = 1, roundDig = 2, drop0 = FALSE)
```

Arguments

x	Data for computation.
mult	multiplier for SEM, default 1, can be set to e.g. 2 or 1.96 to create confidence intervals
roundDig	Number of relevant digits for roundR.
drop0	Should trailing zeros be dropped?

Value

character vector with mean $\pm$ SEM, rounded to desired precision

Examples

```
# basic usage of meanse
meanse(x = mtcars$wt)
```

mean_cl_boot	<i>Compute confidence interval of mean by bootstrapping.</i>
--------------	--

Description

mean_cl_boot computes lower and upper confidence limits for the estimated mean, based on bootstrapping.

Usage

```
mean_cl_boot(
  x,
  conf = 0.95,
  type = "basic",
  nrepl = 10^3,
  round = FALSE,
  roundDig = 2
)
```

Arguments

x	Data for computation.
conf	confidence interval with default 95%.
type	type for function boot.ci.
nrepl	number of bootstrap replications, defaults to 1000.
round	logical, applies roundR function to results. Output is character.
roundDig	Number of relevant digits for function roundR .

Value

A tibble with one row and three columns: Mean, CIlow, CIhigh.

Examples

```
# basic usage of mean_cl_boot
mean_cl_boot(x = mtcars$wt)
```

medianse	<i>Compute standard error of median.</i>
----------	--

Description

medianse is based on [mad](#)/square root(n)

Usage

```
medianse(x)
```

Arguments

x	Data for computation.
---	-----------------------

Value

numeric vector with SE Median.

Examples

```
# basic usage of medianse
medianse(x = mtcars$wt)
```

median_cl_boot	<i>Compute confidence interval of median by bootstrapping.</i>
----------------	--

Description

median_cl_boot computes lower and upper confidence limits for the estimated median, based on bootstrapping.

Usage

```
median_cl_boot(  
  x,  
  conf = 0.95,  
  type = "basic",  
  nrepl = 10^3,  
  round = FALSE,  
  roundDig = 2  
)
```

Arguments

x	Data for computation.
conf	confidence interval with default 95%.
type	type for function boot.ci.
nrepl	number of bootstrap replications, defaults to 1000.
round	logical, applies roundR function to results. Output is character.
roundDig	number of relevant digits for function roundR .

Value

A tibble with one row and three columns: Median, CIlow, CIhigh.

Examples

```
# basic usage of median_cl_boot  
median_cl_boot(x = mtcars$wt)
```

median_cl_boot_gg	Rename output from median_cl_boot for use in ggplot.
-------------------	--

Description

median_cl_boot_gg computes lower and upper confidence limits for the estimated median, based on bootstrapping, using default settings.

Usage

```
median_cl_boot_gg(x)
```

Arguments

x	Data for computation.
---	-----------------------

Value

A tibble with one row and three columns: y, ymin, ymax.

Examples

```
# basic usage of median_cl_boot
median_cl_boot_gg(x = mtcars$wt)
```

median_quart	Compute median and quartiles and put together.
--------------	--

Description

Compute median and quartiles and put together.

Usage

```
median_quart(
  x,
  nround = NULL,
  qtype = 8,
  roundDig = 2,
  drop0 = FALSE,
  groupvar = NULL,
  range = FALSE,
  rangesep = " ",
  rangearrow = " -> ",
  prettynum = FALSE,
  .german = FALSE,
```

```

    add_n = FALSE,
    ci = FALSE,
    nrepl = 10^3,
    singleline = TRUE
  )

```

Arguments

x	Data for computation.
nround	Number of digits for fixed round.
qtype	Type of quantiles.
roundDig	Number of relevant digits for roundR.
drop0	Should trailing zeros be dropped?
groupvar	Optional grouping variable for subgroups.
range	Should min and max be included in output?
rangesep	How should min/max be separated from mean+-sd?
rangearrow	What is put between min -> max?
prettynum	logical, apply prettyNum to results?
.german	logical, should "." and "," be used as bigmark and decimal?
add_n	Should n be included in output?
ci	Should bootstrap based 95% confidence interval be computed?
nrepl	Number of bootstrap replications, defaults to 1000.
singleline	Put all descriptive stats in a single element (default) or below each other. singleline = FALSE sets ci and add_n as TRUE

Value

Either character vector with median [1stQuartile/3rdQuartile] and additional results (singleline), or matrix with rows for n, median with CI, and quartiles, and optionally range.

Examples

```

# basic usage of median_quart
median_quart(x = mtcars$wt)
# with additional options
median_quart(x = mtcars$wt, groupvar = mtcars$am, add_n = TRUE)
data(faketrial)
median_quart(x = faketrial$`Biomarker 1 [units]`, groupvar = faketrial$Treatment)

```

pairwise_fisher_test *Pairwise Fisher's exact tests*

Description

pairwise_fisher_test calculates pairwise comparisons between group levels with corrections for multiple testing.

Usage

```
pairwise_fisher_test(  
  dep_var,  
  indep_var,  
  adjmethod = "fdr",  
  plevel = 0.05,  
  symbols = letters[-1],  
  ref = FALSE  
)
```

Arguments

dep_var	dependent variable, containing the data.
indep_var	independent variable, should be factor or coercible.
adjmethod	method for adjusting p values (see p.adjust).
plevel	threshold for significance.
symbols	predefined as b,c, d...; provides footnotes to mark group differences, e.g. b means different from group 2
ref	is the 1st subgroup the reference (like in Dunnett test)?

Value

A list with elements "methods" (character), "p.value" (matrix), "plevel" (numeric), and "sign_colwise" (vector of length number of levels - 1)

Examples

```
# All pairwise comparisons  
pairwise_fisher_test(dep_var = mtcars$cyl, indep_var = mtcars$gear)  
# Only comparison against reference gear=3  
pairwise_fisher_test(dep_var = mtcars$cyl, indep_var = mtcars$gear, ref = TRUE)
```

pairwise_ordcat_test *Pairwise comparison for ordinal categories*

Description

pairwise_ordcat_test calculates pairwise comparisons for ordinal categories between all group levels with corrections for multiple testing.

Usage

```
pairwise_ordcat_test(
  dep_var,
  indep_var,
  adjmethod = "fdr",
  plevel = 0.05,
  symbols = letters[-1],
  ref = FALSE,
  cmh = TRUE
)
```

Arguments

dep_var	dependent variable, containing the data
indep_var	independent variable, should be factor
adjmethod	method for adjusting p values (see p.adjust)
plevel	threshold for significance
symbols	predefined as b,c, d...; provides footnotes to mark group differences, e.g. b means different from group 2
ref	is the 1st subgroup the reference (like in Dunnett test)
cmh	Should Cochran-Mantel-Haenszel test (coin::cmh_test) be used for testing? If false, the linear-by-linear association test (coin::lbl_test) is applied.

Value

A list with elements "methods" (character), "p.value" (matrix), "plevel" (numeric), and "sign_colwise" (vector of length number of levels - 1)

Examples

```
# All pairwise comparisons
mtcars2 <- dplyr::mutate(mtcars, cyl = factor(cyl, ordered = TRUE))
pairwise_ordcat_test(dep_var = mtcars2$cyl, indep_var = mtcars2$gear)
# Only comparison against reference gear=3
pairwise_ordcat_test(dep_var = mtcars2$cyl, indep_var = mtcars2$gear, ref = TRUE)
```

pairwise_t_test	<i>Extended pairwise t-test</i>
-----------------	---------------------------------

Description

pairwise_t_test calculate pairwise comparisons between group levels with corrections for multiple testing based on [pairwise.t.test](#)

Usage

```
pairwise_t_test(  
  dep_var,  
  indep_var,  
  adjmethod = "fdr",  
  plevel = 0.05,  
  symbols = letters[-1]  
)
```

Arguments

dep_var	dependent variable, containing the data
indep_var	independent variable, should be factor
adjmethod	method for adjusting p values (see p.adjust)
plevel	threshold for significance
symbols	predefined as b,c, d...; provides footnotes to mark group differences, e.g. b means different from group 2

Value

A list with method output of pairwise.t.test, matrix of p-values, and character vector with significance indicators.

Examples

```
pairwise_t_test(dep_var = mtcars$wt, indep_var = mtcars$cyl)
```

pairwise_wilcox_test *Pairwise Wilcoxon tests*

Description

pairwise_wilcox_test calculates pairwise comparisons on ordinal data between all group levels with corrections for multiple testing based on [coin::wilcox_test](#) from package 'coin'.

Usage

```
pairwise_wilcox_test(  
  dep_var,  
  indep_var,  
  strat_var = NA,  
  adjmethod = "fdr",  
  distr = "exact",  
  plevel = 0.05,  
  symbols = letters[-1],  
  sep = ""  
)
```

Arguments

dep_var	dependent variable, containing the data.
indep_var	independent variable, should be factor.
strat_var	optional factor for stratification.
adjmethod	method for adjusting p values (see p.adjust)
distr	Computation of p-values, see coin::wilcox_test .
plevel	threshold for significance.
symbols	predefined as b,c, d...; provides footnotes to mark group differences, e.g. b means different from group 2.
sep	text between statistics and range or other elements.

Value

A list with matrix of adjusted p-values and character vector with significance indicators.

Examples

```
pairwise_wilcox_test(dep_var = mtcars$wt, indep_var = mtcars$cyl)
```

pdf_kable	Enhanced knitr::kable with latex
-----------	--

Description

pdf_kable formats tibbles/df's for markdown

Usage

```
pdf_kable(  
  .input,  
  width1 = 6,  
  twidth = 14,  
  tposition = "left",  
  innercaption = NULL,  
  caption = "",  
  foot = NULL,  
  escape = TRUE  
)
```

Arguments

.input	table to print
width1	Width of 1st column, default 6.
twidth	Default 14
tposition	Default left
innercaption	subheader
caption	header
foot	footnote
escape	see knitr::kable

Value

A character vector of the table source code.

plot_LB	<i>Lineweaver-Burk diagram</i>
---------	--------------------------------

Description

plot_LB plots a Lineweaver-Burk diagram and computes the linear model

Usage

```
plot_LB(
  data,
  substrate,
  velocity,
  group = NULL,
  title = "Lineweaver-Burk-Plot",
  xlab = "1/substrate",
  ylab = "1/velocity"
)
```

Arguments

data	data structure with columns for model data
substrate	colname for substrate concentration
velocity	colname for reaction velocity
group	colname for optional grouping factor
title	title of the plot
xlab	label of the abscissa
ylab	label of the ordinate

Examples

```
MMdata <- data.frame(subst = c(2.00, 1.00, 0.50, 0.25),
                     velo = c(0.2253, 0.1795, 0.1380, 0.1000))

plot_LB(data=MMdata,
        substrate = 'subst', velocity = 'velo')

MMdata <- data.frame(subst = rep(c(2.00, 1.00, 0.50, 0.25),2),
                     velo = c(0.2253, 0.1795, 0.1380, 0.1000,
                               0.4731333, 0.4089333, 0.3473000, 0.2546667),
                     condition = rep(c('C1', 'C2'),each=4))

plot_LB(data=MMdata, substrate = 'subst',
        velocity = 'velo', group='condition')
```



```
plot_MM(data=MMdata,substrate = 'subst',
         velocity = 'velo',group='condition')
```

print_kable	<i>Enhanced knitr::kable with definable number of rows and/or columns for splitting</i>
-------------	---

Description

[Superseded]

package flextable is a more powerful alternative

print_kable formats and prints tibbles/df's in markdown with splitting into sub-tables with repeated caption and header.

Usage

```
print_kable(t, nrow = 30, caption = "", ncol = 100, ...)
```

Arguments

t	table to print.
nrow	number of rows (30) before splitting.
caption	header.
ncol	number of columns (100) before splitting.
...	Further arguments passed to knitr::kable .

Value

No return value, called for side effects.

Examples

```
## Not run:
print_kable(mtcars, caption = "test")

## End(Not run)
```

roundR	<i>Automatic rounding to a reasonable length, based on largest number</i>
--------	---

Description

roundR takes a vector or matrix of numbers and returns rounded values with selected precision and various formatting options.

Usage

```
roundR(  
  roundin,  
  level = 2,  
  smooth = FALSE,  
  textout = TRUE,  
  drop0 = FALSE,  
  .german = FALSE,  
  .bigmark = FALSE  
)
```

Arguments

roundin	A vector or matrix of numbers.
level	A number specifying number of relevant digits to keep.
smooth	A logical specifying if you want rounding before the dot (e.g. 12345 to 12300).
textout	A logical if output is converted to text.
drop0	A logical if trailing zeros should be dropped.
.german	A logical if german numbers should be reported.
.bigmark	A logical if big.mark is to be shown, mark itself depends on parameter .german.

Value

vector of type character (default) or numeric, depending on parameter textout.

Examples

```
roundR(1.23456, level = 3)  
roundR(1.23456, level = 3, .german = TRUE)  
roundR(1234.56, level = 2, smooth = TRUE)
```

SEM	<i>Standard Error of Mean.</i>
-----	--------------------------------

Description

SEM computes standard error of mean.

Usage

```
SEM(x)
```

Arguments

x	Data for computation.
---	-----------------------

Value

numeric vector with SEM.

Examples

```
SEM(x = mtcars$wt)
```

se_median	<i>Compute standard error of median</i>
-----------	---

Description

se_median is based on [mad](#)/square root(n) (Deprecated, please see [medianse](#), which is the same but named more consistently)

Usage

```
se_median(x)
```

Arguments

x	Data for computation.
---	-----------------------

Value

numeric vector with SE Median.

Examples

```
# basic usage of se_median
## Not run:
se_median(x = mtcars$wt)

## End(Not run)
```

surprisal	<i>Compute surprisal aka Shannon information from p-values</i>
-----------	--

Description

surprisal takes p-values and returns s, a value representing the number of consecutive heads on a fair coin, that would be as surprising as the p-value

Usage

```
surprisal(p, precision = 1)
```

Arguments

p	a vector of p-values
precision	rounding level with default 1

Value

a character vector of s-values

tab.search	<i>Search within data.frame or tibble</i>
------------	---

Description

tab.search searches for pattern within a data-frame or tibble, returning column(s) and row(s)

Usage

```
tab.search(searchdata = rawdata, pattern, find.all = T, names.only = FALSE)
```

Arguments

searchdata	table to search in, predefined as rawdata
pattern	regex, for exact matches add ^findme\$
find.all	return all row indices or only 1st per column,default=TRUE
names.only	return only vector of colnames rather than list with names and rows, default=FALSE

Value

A list with numeric vectors for each column giving row numbers of matched elements

t_var_test	<i>Independent sample t-test with test for equal variance</i>
------------	---

Description

t_var_test tests for equal variance based on [var.test](#) and calls t.test, setting the option var.equal accordingly.

Usage

```
t_var_test(data, formula, cutoff = 0.05)
```

Arguments

data	Tibble or data_frame.
formula	Formula object with dependent and independent variable.
cutoff	is significance threshold for equal variances.

Value

A list from [t.test](#)

Examples

```
t_var_test(mtcars, wt ~ am)
# may be used in pipes:
mtcars |> t_var_test(wt ~ am)
```

var_coeff	<i>Compute coefficient of variance.</i>
-----------	---

Description

var_coeff computes relative variability as standard deviation/mean *100

Usage

```
var_coeff(x)
```

Arguments

x	Data for computation.
---	-----------------------

Value

numeric vector with coefficient of variance.

Examples

```
var_coeff(x = mtcars$wt)
```

WINratio	<i>Comparison for groups in clinical trials based on all possible combinations of subjects</i>
----------	--

Description**[Experimental]**

WINratio computes the ratio of wins and losses for any number of comparison rules.

Usage

```
WINratio(data, groupvar, testvars, rules, idvar = NULL, p_digits = 3)
```

Arguments

data	name of data set (tibble/data.frame) to analyze.
groupvar	name of grouping variable, has to translate to 2 groups.
testvars	names of variables for sequential rules.
rules	list of rules (minimal cut-offs) for sequential comparison, negative if reduction is success, positive if increase is beneficial, must not be 0.
idvar	name of identifier variable. If NULL, rownumber is used.
p_digits	level for rounding p-value.

Value

A list with elements:

WINratio=vector with WINratio and CIs,

WINodds=odds ratio of wins and losses, taking ties into account,

p.value=p.value from prop.test,

WINratioCI=character with merged WINratio, CI, and p

testdata= tibble with testdata from cross-join.

Index

* datasets

faketrial, 15
logrange_1, 23

bt, 3

cat_desc_stats, 3
cat_desc_table, 5
cn, 6
coin::cmh_test, 32
coin::lbl_test, 32
coin::wilcox_test, 34
ColSeeker, 6, 16
ColSeeker(), 16
compare2numvars, 7
compare2qualvars, 9
compare_n_numvars, 10
compare_n_qualvars, 11
cor.test, 13
cortestR, 13, 18

detect_outliers, 14

eGFR, 14

faketrial, 15
FindVars, 16
fisher.test, 9, 12
flex2rmd, 17
formatP, 8, 9, 12, 17

ggcormat, 18
glm, 20
glmCI, 19
grep(), 16

identical_cols, 20

knitr::kable, 35, 38
ks.test, 21
ksnormal, 21

label_outliers, 22
logrange_1, 23
logrange_123456789 (logrange_1), 23
logrange_12357 (logrange_1), 23
logrange_15 (logrange_1), 23
logrange_5 (logrange_1), 23

mad, 27, 40
markSign, 24
mean_cl_boot, 26
meansd, 24
meanse, 26
median_cl_boot, 28, 29
median_cl_boot_gg, 29
median_quart, 29
medianse, 27, 40

p.adjust, 31–34
pairwise.t.test, 33
pairwise_fisher_test, 12, 31
pairwise_ordcat_test, 32
pairwise_t_test, 33
pairwise_wilcox_test, 34
pdf_kable, 35
plot_LB, 36
plot_MM, 37
print_kable, 38

roundR, 27, 28, 39

se_median, 40
SEM, 40
surprisal, 41

t.test, 42
t_var_test, 7, 42
tab.search, 41

units, 15

var.test, 42

var_coeff, [42](#)

wilcox.test, [7](#)

WINratio, [43](#)