Package ‘descriptr’

March 20, 2018

Type Package

Title Generate Descriptive Statistics & Explore Statistical Distributions

Version 0.4.1

Description Generate descriptive statistics such as measures of location, dispersion, frequency tables, cross tables, group summaries and multiple one/two way tables. Visualize and compute percentiles/probabilities of normal, t, f, chi square and binomial distributions.

Depends R(>= 3.2.4)

Imports dplyr, forcats, ggplot2, graphics, magrittr, rlang, scales, shiny, tibble, tidyr

Suggests covr, haven, jsonlite, knitr, lubridate, readr, readxl, rmarkdown, shinyBS, shinythemes, stringr, testthat, tools, vdiffr

License MIT + file LICENSE

URL https://descriptr.rsquaredacademy.com/, https://github.com/rsquaredacademy/descriptr

BugReports https://github.com/rsquaredacademy/descriptr/issues

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Aravind Hebbali [aut, cre]

Maintainer Aravind Hebbali <hebbali.aravind@gmail.com>

Repository CRAN

Date/Publication 2018-03-20 09:41:04 UTC

R topics documented:

descriptr 2
dist_binom_plot 3
dist_chi_plot 4
dist_f_plot 5
dist_norm_plot 6
dist_t 7
ds_cross_table 9
ds_css 10
ds_cvar 11
ds_extreme_obs 11
ds_freq_cont 12
ds_freq_table 13
ds_gmean 14
ds_group_summary 14
ds_hmean 16
ds_kurtosis 16
ds_launch_shiny_app 17
ds_mdev 18
ds_measures_location 19
ds_measures_symmetry 19
ds_measures_variation 20
ds_mode 20
ds_multi_stats 21
ds_oway_tables 22
ds_percentiles 23
ds_range 23
ds_rindex 24
ds_screener 25
ds_skewness 26
ds_std_error 27
ds_summary_stats 27
ds_tailobs 28
hsb 29
mtcarz 29

Index **30**

`descriptr` `descriptr` *package*

Description

Generate descriptive statistics and explore statistical distributions

`dist_binom_plot` *Visualize binomial distribution*

Description

Visualize how changes in number of trials and the probability of success affect the shape of the binomial distribution. Compute & visualize probability from a given quantile and quantiles out of given probability.

Usage

```
dist_binom_plot(n, p)

dist_binom_prob(n, p, s, type = c("lower", "upper", "exact", "interval"))

dist_binom_perc(n, p, tp, type = c("lower", "upper"))
```

Arguments

`n`	Number of trials.
`p`	Aggregate probability.
`s`	Number of success.
`type`	Lower/upper/exact/interval.
`tp`	Probability of success in a trial.

Value

A list containing the following components:

`avg`	Mean of the binomial distribution,
`stdev`	Standard deviation of the binomial distribution.
`prob`	Probability of success.

Deprecated functions

`binom_plot()`, `binom_prob()`, `binom_perc()` have been deprecated. Instead use `dist_binom_plot()`, `dist_binom_prob()` and `dist_binom_perc()`.

See Also

Binomial

Examples

```
# visualize binomial distribution
dist_binom_plot(10, 0.3)

# visualize probability from a given quantile
dist_binom_prob(10, 0.3, 4, type = 'exact')
dist_binom_prob(10, 0.3, 4, type = 'lower')
dist_binom_prob(10, 0.3, 4, type = 'upper')
dist_binom_prob(10, 0.3, c(4, 6), type = 'interval')

# visualize quantiles out of given probability
dist_binom_perc(10, 0.5, 0.05)
dist_binom_perc(10, 0.5, 0.05, "upper")
```

`dist_chi_plot` *Visualize chi square distribution*

Description

Visualize how changes in degrees of freedom affect the shape of the chi square distribution. Compute & visualize quantiles out of given probability and probability from a given quantile.

Usage

```
dist_chi_plot(df = 3, normal = FALSE)

dist_chi_perc(probs = 0.95, df = 3, type = c("lower", "upper"))

dist_chi_prob(perc, df, type = c("lower", "upper"))
```

Arguments

`df`	Degrees of freedom.
`normal`	If `TRUE`, normal curve with same `mean` and `sd` as the chi square distribution is drawn.
`probs`	Probability value.
`type`	Lower tail or upper tail.
`perc`	Quantile value.

Value

Percentile for the `probs` based on `df` and `type` or probability value for `perc` based on `df` and `type`.

Deprecated functions

`chi_plot()`, `chi_prob()` and `chi_per()` have been deprecated. Instead use `dist_chi_plot()`, `dist_chi_prob()` and `dist_chi_perc()`.

See Also

`Chisquare`

Examples

```
# visualize chi square distribution
dist_chi_plot()
dist_chi_plot(df = 5)
dist_chi_plot(df = 5, normal = TRUE)

# visualize quantiles out of given probability
dist_chi_perc(0.165, 8, 'upper')
dist_chi_perc(0.22, 13, 'upper')

# visualize probability from a given quantile.
dist_chi_prob(13.58, 11, 'lower')
dist_chi_prob(15.72, 13, 'upper')
```

`dist_f_plot` *Visualize f distribution*

Description

Visualize how changes in degrees of freedom affect the shape of the F distribution. Compute & visualize quantiles out of given probability and probability from a given quantile.

Usage

```
dist_f_plot(num_df = 4, den_df = 30, normal = FALSE)

dist_f_perc(probs = 0.95, num_df = 3, den_df = 30, type = c("lower",
  "upper"))

dist_f_prob(perc, num_df, den_df, type = c("lower", "upper"))
```

Arguments

`num_df`	Degrees of freedom associated with the numerator of f statistic.
`den_df`	Degrees of freedom associated with the denominator of f statistic.
`normal`	If `TRUE`, normal curve with same `mean` and `sd` as the F distribution is drawn.
`probs`	Probability value.
`type`	Lower tail or upper tail.
`perc`	Quantile value.

Value

Percentile for the `probs` based on `num_df`, `den_df` and `type` or probability value for `perc` based on `num_df`, `den_df` and `type`.

Deprecated functions

`f_plot()`, `f_prob()` and `f_per()` have been deprecated. Instead use `dist_f_plot()`, `dist_f_prob()` and `dist_f_perc()`.

See Also

`FDist`

Examples

```
# visualize F distribution
dist_f_plot()
dist_f_plot(6, 10, normal = TRUE)

# visualize probability from a given quantile
dist_f_perc(0.95, 3, 30, 'lower')
dist_f_perc(0.125, 9, 35, 'upper')

# visualize quantiles out of given probability
dist_f_prob(2.35, 5, 32)
dist_f_prob(1.5222, 9, 35, type = "upper")
```

`dist_norm_plot` *Visualize normal distribution*

Description

Visualize how changes in mean and standard deviation affect the shape of the normal distribution. Compute & visualize quantiles out of given probability and probability from a given quantile.

Usage

```
dist_norm_plot(mean = 0, sd = 1)

dist_norm_perc(probs = 0.95, mean = 0, sd = 1, type = c("lower",
  "upper", "both"))

dist_norm_prob(perc, mean = 0, sd = 1, type = c("lower", "upper", "both"))
```

Arguments

`mean`	Mean of the normal distribution.
`sd`	Standard deviation of the normal distribution.
`probs`	Probability value.
`type`	Lower tail, upper tail or both.
`perc`	Quantile value.

Value

Percentile for the `probs` based on `mean`, `sd` and `type` or probability value for `perc` based on `mean`, `sd` and `type`.

Deprecated functions

`norm_plot()`, `norm_prob()` and `norm_per()` have been deprecated. Instead use `dist_norm_plot()`, `dist_norm_prob()` and `dist_norm_per()`.

See Also

`Normal`

Examples

```
# visualize normal distribution
dist_norm_plot()
dist_norm_plot(mean = 2, sd = 0.6)

# visualize probability from a given quantile
dist_norm_prob(3.78, mean = 2, sd = 1.36)
dist_norm_prob(3.43, mean = 2, sd = 1.36, type = 'upper')
dist_norm_prob(c(-1.74, 1.83), type = 'both')

# visualize quantiles out of given probability
dist_norm_perc(0.95, mean = 2, sd = 1.36)
dist_norm_perc(0.3, mean = 2, sd = 1.36, type = 'upper')
dist_norm_perc(0.95, mean = 2, sd = 1.36, type = 'both')
```

`dist_t` *Visualize t distribution*

Description

Visualize how degrees of freedom affect the shape of t distribution, visualize quantiles out of given probability and probability from a given quantile.

Usage

```
dist_t_plot(df = 3)

dist_t_perc(probs = 0.95, df = 4, type = c("lower", "upper", "both"))

dist_t_prob(perc, df, type = c("lower", "upper", "interval", "both"))
```

Arguments

`df`	Degrees of freedom.
`probs`	Probability value.
`type`	Lower tail, upper tail, interval or both.
`perc`	Quantile value.

Value

Percentile for the `probs` based on `df` and `type` or probability value for the `perc` based on `df` and `type`.

Deprecated functions

`t_plot()`, `t_prob()` and `t_per()` have been deprecated. Instead use `dist_t_plot()`, `dist_t_prob()` and `dist_t_perc()`.

See Also

`TDist`

Examples

```
# visualize t distribution
dist_t_plot()
dist_t_plot(6)
dist_t_plot(df = 8)

# visualize quantiles out of given probability
dist_t_perc(probs = 0.95, df = 4, type = 'lower')
dist_t_perc(probs = 0.35, df = 4, type = 'upper')
dist_t_perc(probs = 0.69, df = 7, type = 'both')

# visualize probability from a given quantile
dist_t_prob(2.045, 7, 'lower')
dist_t_prob(0.945, 7, 'upper')
dist_t_prob(1.445, 7, 'interval')
dist_t_prob(1.6, 7, 'both')
```

`ds_cross_table` *Two way table*

Description

Creates two way tables of categorical variables. The tables created can be visualized as barplots and mosaicplots.

Usage

```
ds_cross_table(data, var1, var2)

## S3 method for class 'ds_cross_table'
plot(x, stacked = FALSE, proportional = FALSE, ...)

ds_twoway_table(data, var1, var2)
```

Arguments

`data`	A `data.frame` or a `tibble`.
`var1`	First categorical variable.
`var2`	Second categorical variable.
`x`	An object of class `cross_table`.
`stacked`	If `FALSE`, the columns of height are portrayed as stacked bars, and if `TRUE` the columns are portrayed as juxtaposed bars.
`proportional`	If `TRUE`, the height of the bars is proportional.
`...`	Further arguments to be passed to or from methods.

Deprecated function

`ds_cross_table()` has been deprecated. Instead use `ds_cross_table()`.

Examples

```
k <- ds_cross_table(mtcarz, cyl, gear)
k

# bar plots
plot(k)
plot(k, stacked = TRUE)
plot(k, proportional = TRUE)

# alternate
ds_twoway_table(mtcarz, cyl, gear)
```

ds_css *Corrected Sum of Squares*

Description

Compute the corrected sum of squares

Usage

```
ds_css(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector containing the values whose mode is to be computed
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Any NA values are stripped from `x` before computation takes place.

Value

Corrected sum of squares of `x`

Deprecated Function

`stat_css()` has been deprecated. Instead use `ds_css()`.

References

NIST/SEMATECH e-Handbook of Statistical Methods

Examples

```
ds_css(mtcars$mpg)
```

`ds_cvar` *Coefficient of Variation*

Description

Compute the coefficient of variation

Usage

```
ds_cvar(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector containing the values whose mode is to be computed
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Any NA values are stripped from `x` before computation takes place.

Deprecated Function

`stat_cvar()` has been deprecated. Instead use `ds_cvar()`.

Examples

```
ds_cvar(mtcars$mpg)
```

`ds_extreme_obs` *Extreme observations*

Description

Returns the most extreme observations.

Usage

```
ds_extreme_obs(data, column)
```

Arguments

`data`	A `data.frame` or `tibble`.
`column`	Column in `data`.

Examples

```
ds_extreme_obs(mtcarz, mpg)
```

`ds_freq_cont` *Frequency distribution of continuous data*

Description

Frequency distribution of continuous data by splitting into equidistant intervals created based on the number of bins specified. `hist.ds_freq_cont()` creates histogram for the frequency table created using `ds_freq_cont()`.

Usage

```
ds_freq_cont(data, variable, bins = 5)

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

Arguments

`data`	A `data.frame` or a `tibble`.
`variable`	Column in `data`.
`bins`	Number of intervals into which the data must be split.
`x`	An object of class `ds_freq_cont`.
`...`	Further arguments to be passed to or from methods.

Value

A tibble.

Deprecated functions

`freq_cont()` has been deprecated. Instead use `ds_freq_cont()`.

See Also

`ds_freq_table` `ds_cross_table`

Examples

```
# frequency table
ds_freq_cont(mtcarz, mpg, 4)

# histogram
k <- ds_freq_cont(mtcarz, mpg, 4)
plot(k)
```

`ds_freq_table`	*Frequency table*

Description

Frequency table for factor data and returns the frequency, cumulative frequency, frequency percent and cumulative frequency percent. `barplot.ds_freq_table()` creates bar plot for the frequency table created using `ds_freq_table()`.

Usage

```
ds_freq_table(data, variable)

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

Arguments

`data`	A `data.frame` or a `tibble`.
`variable`	Column in `data`.
`x`	An object of class `ds_freq_table`.
`...`	Further arguments to be passed to or from methods.

Value

`ds_freq_table` returns an object of class `"ds_freq_table"`. An object of class `"ds_freq_table"` is a list containing the following components:

`ftable`	Frequency table.

Deprecated function

`freq_table()` has been deprecated. Instead use `ds_freq_table()`.

See Also

`ds_freq_cont` `ds_cross_table`

Examples

```
# frequency table
ds_freq_table(mtcarz, cyl)

# barplot
k <- ds_freq_table(mtcarz, cyl)
plot(k)
```

`ds_gmean` *Geometric Mean*

Description

Compute the geometric mean

Usage

```
ds_gmean(x, na.rm = FALSE, ...)
```

Arguments

`x`	a numeric vector containing the values whose geometric mean is to be computed
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.
`...`	further arguments passed to or from other methods #' @details Any NA values are stripped from `x` before computation takes place.

Value

Returns the geometric mean of `x`

Deprecated function

`gmean()` has been deprecated. Instead use `ds_gmean()`.

See Also

`ds_hmean` `mean`

Examples

```
ds_gmean(mtcars$mpg)
```

`ds_group_summary` *Groupwise descriptive statistics*

Description

Descriptive statistics of a continuous variable for the different levels of a categorical variable. `boxplot.group_summary()` creates boxplots of the continuous variable for the different levels of the categorical variable.

Usage

```
ds_group_summary(data, gvar, cvar)

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

Arguments

data	A data.frame or a tibble.
gvar	Column in data.
cvar	Column in data.
x	An object of the class ds_group_summary.
...	Further arguments to be passed to or from methods.

Value

ds_group_summary() returns an object of class "ds_group_summary". An object of class "ds_group_summary" is a list containing the following components:

stats	A data frame containing descriptive statistics for the different levels of the factor variable.
tidy_stats	A tibble containing descriptive statistics for the different levels of the factor variable.
plotdata	Data for boxplot method.

Deprecated function

ds_group_summary() has been deprecated. Instead use ds_group_summary().

See Also

ds_summary_stats

Examples

```
# ds_group summary
ds_group_summary(mtcarz, cyl, mpg)

# boxplot
k <- ds_group_summary(mtcarz, cyl, mpg)
plot(k)

# tibble
k$tidy_stats
```

ds_hmean *Harmonic Mean*

Description

Compute the harmonic mean

Usage

```
ds_hmean(x, na.rm = FALSE, ...)
```

Arguments

x	a numeric vector containing the values whose harmonic mean is to be computed
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods #' @details Any NA values are stripped from x before computation takes place.

Value

Returns the harmonic mean of x

Deprecated function

hmean() has been deprecated. Instead use ds_hmean().

See Also

ds_gmean mean

Examples

```
ds_hmean(mtcars$mpg)
```

ds_kurtosis *Kurtosis*

Description

Compute the kurtosis of a probability distribution.

Usage

```
ds_kurtosis(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector containing the values whose kurtosis is to be computed
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Any NA values are stripped from `x` before computation takes place.

Value

Kurtosis of `x`

Deprecated Function

`kurtosis()` has been deprecated. Instead use `ds_kurtosis()`.

References

Sheskin, D.J. (2000) Handbook of Parametric and Nonparametric Statistical Procedures, Second Edition. Boca Raton, Florida: Chapman & Hall/CRC.

See Also

`ds_skewness`

Examples

```
ds_kurtosis(mtcars$mpg)
```

`ds_launch_shiny_app` *Launch Shiny App*

Description

Launches shiny app

Usage

```
ds_launch_shiny_app()
```

Deprecated Function

`launch_descriptr()` has been deprecated. Instead use `ds_launch_shiny_app()`.

Examples

```
## Not run:
ds_launch_shiny_app()

## End(Not run)
```

---

ds_mdev *Mean Absolute Deviation*

Description

Compute the mean absolute deviation about the mean

Usage

```
ds_mdev(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

The `stat_mdev` function computes the mean absolute deviation about the mean. It is different from `mad` in `stats` package as the statistic used to compute the deviations is not `median` but `mean`. Any NA values are stripped from `x` before computation takes place

Value

Mean absolute deviation of `x`

Deprecated Function

`stat_mdev()` has been deprecated. Instead use `ds_mdev()`.

See Also

`mad`

Examples

```
ds_mdev(mtcars$mpg)
```

`ds_measures_location` *Measures of location*

Description

Returns the measures of location such as mean, median & mode.

Usage

```
ds_measures_location(data, column, trim = 0.05)
```

Arguments

`data`	A `data.frame` or `tibble`.
`column`	Column in `data`.
`trim`	The fraction of values to be trimmed before computing the mean.

Examples

```
ds_measures_location(mtcarz, mpg)
```

`ds_measures_symmetry` *Measures of symmetry*

Description

Returns the measures of symmetry such as skewness and kurtosis.

Usage

```
ds_measures_symmetry(data, column)
```

Arguments

`data`	A `data.frame` or `tibble`.
`column`	Column in `data`.

Examples

```
ds_measures_symmetry(mtcarz, mpg)
```

`ds_measures_variation` *Measures of variation*

Description

Returns the measures of location such as range, variance and standard deviation.

Usage

```
ds_measures_variation(data, column)
```

Arguments

`data`	A `data.frame` or `tibble`.
`column`	Column in `data`.

Examples

```
ds_measures_variation(mtcarz, mpg)
```

`ds_mode` *Mode*

Description

Compute the sample mode

Usage

```
ds_mode(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector containing the values whose mode is to be computed
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Any NA values are stripped from `x` before computation takes place.

Value

Mode of `x`

Deprecated Function

‘stat_mode()‘ has been deprecated. Instead use ‘ds_mode()‘.

See Also

`mean` `median`

Examples

```
ds_mode(mtcars$mpg)
ds_mode(mtcars$cyl)
```

`ds_multi_stats` *Multiple variable statistics*

Description

Descriptive statistics for multiple variables.

Usage

```
ds_multi_stats(x, ...)
```

Arguments

`x`	A `tibble` or a `data.frame`.
`...`	Columns in `x`.

Value

A tibble.

Deprecated function

`multistats()` has been deprecated. Instead use `ds_multi_stats()`

Examples

```
ds_multi_stats(mtcarz, mpg, disp, hp)
```

`ds_oway_tables` *Multiple One & Two Way Tables*

Description

`ds_oway_tables` creates multiple one way tables by creating a frequency table for each categorical variable in a data frame. `ds_tway_tables` creates multiple two way tables by creating a cross table for each unique pair of categorical variables in a data frame.

Usage

```
ds_oway_tables(data)

ds_tway_tables(data)
```

Arguments

`data`	a data frame

Details

`ds_oway_tables` is a extension of the `ds_freq_table` function. It creates a frequency table for each categorical variable in the dataframe. `ds_tway_tables` is a extension of the `ds_cross_table` function. It creates a two way table for each unique pair of categorical variables in the dataframe.

Deprecated Functions

`oway_tables()` and `tway_tables()` have been deprecated. Instead use `ds_oway_tables()` and `ds_tway_tables()`.

See Also

`link{ds_freq_table}` `link{ds_cross_table}`

Examples

```
# multiple one way tables
ds_oway_tables(mtcarz)

# multiple two way tables
ds_tway_tables(mtcarz)
```

`ds_percentiles` *Percentiles*

Description

Returns the percentiles

Usage

```
ds_percentiles(data, column)
```

Arguments

`data`	A `data.frame` or `tibble`.
`column`	Column in `data`.

Examples

```
ds_percentiles(mtcarz, mpg)
```

`ds_range` *Range*

Description

Compute the range of a numeric vector

Usage

```
ds_range(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Value

Range of `x`

Deprecated Function

`stat_range()` has been deprecated. Instead use `ds_range()`.

See Also

range

Examples

```
ds_range(mtcars$mpg)
```

`ds_rindex` *Index Values*

Description

Returns index of values.

Usage

```
ds_rindex(data, values)
```

Arguments

`data`	a numeric vector
`values`	a numeric vector containing the values whose index is returned

Details

Any NA values are stripped from `data` and `values` before computation takes place.

Value

Index of the `values` in `data`. In case, `data` does not contain `index`, `NULL` is returned.

Deprecated Function

`rindex()` has been deprecated. Instead use `ds_rindex()`.

Examples

```
ds_rindex(mtcars$mpg, 21)
ds_rindex(mtcars$mpg, 22)
```

`ds_screener` *Screen data*

Description

Screen data and return details such as variable names, class, levels and missing values. `plot.ds_screener()` creates bar plots to visualize of missing observations for each variable in a data set.

Usage

```
ds_screener(y)

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

Arguments

`y`	A `tibble` or a `data.frame`.
`x`	An object of class `ds_screener`.
`...`	Further arguments to be passed to or from methods.

Value

`ds_screener()` returns an object of class `"ds_screener"`. An object of class `"ds_screener"` is a list containing the following components:

`Rows`	Number of rows in the data frame.
`Columns`	Number of columns in the data frame.
`Variables`	Names of the variables in the data frame.
`Types`	Class of the variables in the data frame.
`Count`	Length of the variables in the data frame.
`nlevels`	Number of levels of a factor variable.
`levels`	Levels of factor variables in the data frame.
`Missing`	Number of missing observations in each variable.
`MissingPer`	Percent of missing observations in each variable.
`MissingTotal`	Total number of missing observations in the data frame.
`MissingTotPer`	Total percent of missing observations in the data frame.
`MissingRows`	Total number of rows with missing observations in the data frame.
`MissingCols`	Total number of columns with missing observations in the data frame.

Deprecated function

`screener()` has been deprecated. Instead use `ds_screener()`.

Examples

```
# screen data
ds_screener(mtcarz)
```

`ds_skewness` *Skewness*

Description

Compute the skewness of a probability distribution.

Usage

```
ds_skewness(x, na.rm = FALSE)
```

Arguments

`x`	a numeric vector containing the values whose skewness is to be computed
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Any NA values are stripped from `x` before computation takes place.

Value

Skewness of `x`

Deprecated Function

`skewness()` has been deprecated. Instead use `ds_skewness()`.

References

Sheskin, D.J. (2000) Handbook of Parametric and Nonparametric Statistical Procedures, Second Edition. Boca Raton, Florida: Chapman & Hall/CRC.

See Also

`kurtosis`

Examples

```
ds_skewness(mtcars$mpg)
```

`ds_std_error` *Standard error of mean*

Description

Returns the standard error of mean.

Usage

```
ds_std_error(x)
```

Arguments

`x`	A numeric vector.

Examples

```
ds_std_error(mtcars$mpg)
```

`ds_summary_stats` *Descriptive statistics*

Description

Range of descriptive statistics for continuous data.

Usage

```
ds_summary_stats(data, variable)
```

Arguments

`data`	A `data.frame` or `tibble`.
`variable`	Column in `data`.

Deprecated function

`summary_stats()` has been deprecated. Instead use `ds_summary_stats()`.

See Also

`summary` `ds_freq_cont` `ds_freq_table` `ds_cross_table`

Examples

```
ds_summary_stats(mtcarz, mpg)
```

ds_tailobs *Tail Observations*

Description

Returns the n highest/lowest observations from a numeric vector.

Usage

```
ds_tailobs(data, n, type = c("low", "high"))
```

Arguments

data	a numeric vector
n	number of observations to be returned
type	if low, the n lowest observations are returned, else the highest n obervations are returned

Details

Any NA values are stripped from data before computation takes place.

Value

n highest/lowest observations from data

Deprecated function

tailobs() has been deprecated. Instead use ds_tailobs().

See Also

top_n

Examples

```
ds_tailobs(mtcarz$mpg, 5)
ds_tailobs(mtcarz$mpg, 5, type = "high")
```

`hsb` *High School and Beyond Data Set*

Description

A dataset containing demographic information and standardized test scores of high school students.

Usage

```
hsb
```

Format

A data frame with 200 rows and 10 variables:

id id of the student
female gender of the student
race ethnic background of the student
ses socio-economic status of the student
schtyp school type
prog program type
read scores from test of reading
write scores from test of writing
math scores from test of math
science scores from test of science
socst scores from test of social studies

Source

`http://www.ats.ucla.edu/stat/spss/whatstat/whatstat.htm`

`mtcarz` *mtcarz*

Description

Copy of mtcars data set with modified variable types

Usage

```
mtcarz
```

Format

An object of class `data.frame` with 32 rows and 11 columns.

Index

∗Topic **datasets**
    hsb, 29
    mtcarz, 29

binom_perc (dist_binom_plot), 3
binom_plot (dist_binom_plot), 3
binom_prob (dist_binom_plot), 3
Binomial, *3*

chi_per (dist_chi_plot), 4
chi_plot (dist_chi_plot), 4
chi_prob (dist_chi_plot), 4
Chisquare, *5*
cross_table (ds_cross_table), 9

descriptr, 2
descriptr-package (descriptr), 2
dist_binom_perc (dist_binom_plot), 3
dist_binom_plot, 3
dist_binom_prob (dist_binom_plot), 3
dist_chi_perc (dist_chi_plot), 4
dist_chi_plot, 4
dist_chi_prob (dist_chi_plot), 4
dist_f_perc (dist_f_plot), 5
dist_f_plot, 5
dist_f_prob (dist_f_plot), 5
dist_norm_perc (dist_norm_plot), 6
dist_norm_plot, 6
dist_norm_prob (dist_norm_plot), 6
dist_t, 7
dist_t_perc (dist_t), 7
dist_t_plot (dist_t), 7
dist_t_prob (dist_t), 7
ds_cross_table, 9, *12*, *13*, *27*
ds_css, 10
ds_cvar, 11
ds_extreme_obs, 11
ds_freq_cont, 12, *13*, *27*
ds_freq_table, *12*, 13, *27*
ds_gmean, 14, *16*
ds_group_summary, 14
ds_hmean, *14*, 16
ds_kurtosis, 16
ds_launch_shiny_app, 17
ds_mdev, 18
ds_measures_location, 19
ds_measures_symmetry, 19
ds_measures_variation, 20
ds_mode, 20
ds_multi_stats, 21
ds_oway_tables, 22
ds_percentiles, 23
ds_range, 23
ds_rindex, 24
ds_screener, 25
ds_skewness, 26
ds_std_error, 27
ds_summary_stats, *15*, 27
ds_tailobs, 28
ds_tway_tables (ds_oway_tables), 22
ds_twoway_table (ds_cross_table), 9

f_per (dist_f_plot), 5
f_plot (dist_f_plot), 5
f_prob (dist_f_plot), 5
FDist, *6*
freq_cont (ds_freq_cont), 12
freq_table (ds_freq_table), 13

gmean (ds_gmean), 14
group_summary (ds_group_summary), 14

hmean (ds_hmean), 16
hsb, 29

kurtosis (ds_kurtosis), 16

launch_descriptr (ds_launch_shiny_app),
        17

mad, *18*

mean, *14*, *16*, *21*
median, *21*
mtcarz, 29
multistats *(*ds_multi_stats*)*, 21

norm_per *(*dist_norm_plot*)*, 6
norm_plot *(*dist_norm_plot*)*, 6
norm_prob *(*dist_norm_plot*)*, 6
Normal, *7*

oway_tables *(*ds_oway_tables*)*, 22

plot.ds_cross_table *(*ds_cross_table*)*, 9
plot.ds_freq_cont *(*ds_freq_cont*)*, 12
plot.ds_freq_table *(*ds_freq_table*)*, 13
plot.ds_group_summary
    *(*ds_group_summary*)*, 14
plot.ds_screener *(*ds_screener*)*, 25

range, *24*
rindex *(*ds_rindex*)*, 24

screener *(*ds_screener*)*, 25
skewness *(*ds_skewness*)*, 26
stat_css *(*ds_css*)*, 10
stat_cvar *(*ds_cvar*)*, 11
stat_mdev *(*ds_mdev*)*, 18
stat_mode *(*ds_mode*)*, 20
stat_range *(*ds_range*)*, 23
summary, *27*
summary_stats *(*ds_summary_stats*)*, 27

t_per *(*dist_t*)*, 7
t_plot *(*dist_t*)*, 7
t_prob *(*dist_t*)*, 7
tailobs *(*ds_tailobs*)*, 28
TDist, *8*
top_n, *28*
tway_tables *(*ds_oway_tables*)*, 22