Package ‘descriptr’

February 1, 2020

Type   Package
Title  Generate Descriptive Statistics
Version 0.5.1
Description Generate descriptive statistics such as measures of location, dispersion, frequency tables, cross tables, group summaries and multiple one/two way tables.
Depends R(>= 3.2)
Imports cli, dplyr, ggplot2, graphics, gridExtra, magrittr, purrr, rlang, scales, stats, tibble, tidyr, utils, vistributions, xplorerr
Suggests covr, knitr, rmarkdown, testthat, vdiffr
License MIT + file LICENSE
BugReports https://github.com/rsquaredacademy/descriptr/issues
Encoding UTF-8
LazyData true
VignetteBuilder knitr
RoxygenNote 6.1.1
NeedsCompilation no
Author Aravind Hebbali [aut, cre] (<https://orcid.org/0000-0001-9220-9669>)
Maintainer Aravind Hebbali <hebbali.aravind@gmail.com>
Repository CRAN
Date/Publication 2020-02-01 08:20:02 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>descriptr</td>
<td>3</td>
</tr>
<tr>
<td>dist_binom_plot</td>
<td>3</td>
</tr>
<tr>
<td>dist_chi_plot</td>
<td>4</td>
</tr>
<tr>
<td>dist_f_plot</td>
<td>5</td>
</tr>
<tr>
<td>dist_norm_plot</td>
<td>6</td>
</tr>
<tr>
<td>dist_t</td>
<td>7</td>
</tr>
<tr>
<td>ds_auto_freq_table</td>
<td>8</td>
</tr>
<tr>
<td>ds_auto_group_summary</td>
<td>9</td>
</tr>
<tr>
<td>ds_auto_summary_stats</td>
<td>10</td>
</tr>
<tr>
<td>ds_cross_table</td>
<td>10</td>
</tr>
<tr>
<td>ds_css</td>
<td>11</td>
</tr>
<tr>
<td>ds_cvar</td>
<td>12</td>
</tr>
<tr>
<td>ds_extreme_obs</td>
<td>12</td>
</tr>
<tr>
<td>ds_freq_table</td>
<td>13</td>
</tr>
<tr>
<td>ds_gmean</td>
<td>14</td>
</tr>
<tr>
<td>ds_group_summary</td>
<td>14</td>
</tr>
<tr>
<td>ds_group_summary_interact</td>
<td>15</td>
</tr>
<tr>
<td>ds_hmean</td>
<td>16</td>
</tr>
<tr>
<td>ds_kurtosis</td>
<td>17</td>
</tr>
<tr>
<td>ds_launch_shiny_app</td>
<td>18</td>
</tr>
<tr>
<td>ds_mdev</td>
<td>18</td>
</tr>
<tr>
<td>ds_measures_location</td>
<td>19</td>
</tr>
<tr>
<td>ds_measures_symmetry</td>
<td>20</td>
</tr>
<tr>
<td>ds_measures_variation</td>
<td>20</td>
</tr>
<tr>
<td>ds_mode</td>
<td>21</td>
</tr>
<tr>
<td>ds_percentiles</td>
<td>21</td>
</tr>
<tr>
<td>ds_plot_bar</td>
<td>22</td>
</tr>
<tr>
<td>ds_plot_bar_grouped</td>
<td>23</td>
</tr>
<tr>
<td>ds_plot_bar_stacked</td>
<td>23</td>
</tr>
<tr>
<td>ds_plot_box_group</td>
<td>24</td>
</tr>
<tr>
<td>ds_plot_box_single</td>
<td>24</td>
</tr>
<tr>
<td>ds_plot_density</td>
<td>25</td>
</tr>
<tr>
<td>ds_plot_histogram</td>
<td>25</td>
</tr>
<tr>
<td>ds_plot_scatter</td>
<td>26</td>
</tr>
<tr>
<td>ds_range</td>
<td>26</td>
</tr>
<tr>
<td>ds_rindex</td>
<td>27</td>
</tr>
<tr>
<td>ds_screener</td>
<td>28</td>
</tr>
<tr>
<td>ds_skewness</td>
<td>29</td>
</tr>
<tr>
<td>ds_std_error</td>
<td>30</td>
</tr>
<tr>
<td>ds_summary_stats</td>
<td>30</td>
</tr>
<tr>
<td>ds_tailobs</td>
<td>31</td>
</tr>
<tr>
<td>ds_tidy_stats</td>
<td>31</td>
</tr>
<tr>
<td>hsb</td>
<td>32</td>
</tr>
<tr>
<td>mtcarz</td>
<td>33</td>
</tr>
</tbody>
</table>

**Index** 34
descriptr

**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**

```r
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.

**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.

See Also

Chisquare
dist_f_plot

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.
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.
See Also

Normal

Examples

```r
# 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**

```r
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`. 
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_auto_freq_table  Multiple One & Two Way Tables

Description

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

Usage

ds_auto_freq_table(data, ...)
ds_auto_cross_table(data, ...)

Arguments

data A data.frame or tibble.
...

Details

ds_auto_freq_table is a extension of the ds_freq_table function. It creates a frequency table for each categorical variable in the dataframe. ds_auto_cross_table 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**

`ds_oway_tables()` and `ds_tway_tables()` have been deprecated. Instead use `ds_auto_freq_table()` and `ds_auto_cross_table()`.

**See Also**

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

**Examples**

```r
# multiple one way tables
ds_auto_freq_table(mtcarz)
ds_auto_freq_table(mtcarz, cyl, gear)

# multiple two way tables
ds_auto_cross_table(mtcarz)
ds_auto_cross_table(mtcarz, cyl, gear, am)
```

---

### ds_auto_group_summary Tabulation

**Description**

Generate summary statistics for all continuous variables in data.

**Usage**

```r
ds_auto_group_summary(data, ...)
```

**Arguments**

- `data` A `data.frame` or `tibble`.
- `...` Column(s) in `data`.

**Examples**

```r
ds_auto_group_summary(mtcarz, cyl, gear, mpg, disp)
```
**ds_auto_summary_stats**  
*Descriptive statistics and frequency tables*

**Description**

Generate summary statistics & frequency table for all continuous variables in data.

**Usage**

```
ds_auto_summary_stats(data, ...)
```

**Arguments**

- `data`  
  A `data.frame` or `tibble`.
- `...`  
  Column(s) in `data`.

**Examples**

```
ds_auto_summary_stats(mtcarz)
ds_auto_summary_stats(mtcarz, disp, hp)
```

---

**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, 
     print_plot = TRUE, ...)
```

```
ds_twoway_table(data, var1, var2)
```
**ds Css**

**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.
- **print_plot**
  logical; if TRUE, prints the plot else returns a plot object.
- **...**
  Further arguments to be passed to or from methods.

**Examples**

```r
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**

```r
dc_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.
ds_extreme_obs

Value
Corrected sum of squares of x

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.

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.
**Description**

Frequency table for categorical and continuous data and returns the frequency, cumulative frequency, frequency percent and cumulative frequency percent. plot(ds_freq_table()) creates bar plot for the categorical data and histogram for continuous data.

**Usage**

```r
ds_freq_table(data, variable, bins = 5)
## S3 method for class 'ds_freq_table'
plot(x, print_plot = TRUE, ...)
```

**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_table.
- `print_plot`: logical; if TRUE, prints the plot else returns a plot object.
- `...`: Further arguments to be passed to or from methods.

**See Also**

- `ds_cross_table`

**Examples**

```r
# categorical data
ds_freq_table(mtcarz, cyl)

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

# continuous data
ds_freq_table(mtcarz, mpg)

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

*Geometric Mean*

**Description**
Compute the geometric mean

**Usage**

```r
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`

**See Also**

- `ds_hmean`
- `mean`

**Examples**

```r
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**

```r
ds_group_summary(data, gvar, cvar)
```

```r
## S3 method for class 'ds_group_summary'
plot(x, print_plot = TRUE, ...)
```
ds_group_summary_interact

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`.
- **print_plot**: logical; if TRUE, prints the plot else returns a plot object.
- **...**: 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.

See Also

ds_summary_stats

Examples

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

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

# tibble
k$tidy_stats
```

---

ds_group_summary_interact

Category wise descriptive statistics

Description

Descriptive statistics of a continuous variable for the combination of levels of two or more categorical variables.
ds_hmean

Usage

ds_group_summary_interact(data, cvar, ...)

Arguments

data A data.frame or a tibble.
cvar Column in data; continuous variable.
... Columns in data; categorical variables.

See Also

ds_group_summary

Examples

ds_group_summary_interact(mtcarz, mpg, cyl, gear)

dhmean

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

Value

Returns the harmonic mean of x

See Also

ds_gmean mean

Examples

ds_hmean(mtcars$mpg)
ds_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

References


See Also

ds_skewness

Examples

ds_kurtosis(mtcars$mpg)
ds_mdev

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

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
ds_measures_location

Value
Mean absolute deviation of x

See Also
mad

Examples
ds_mdev(mtcars$mpg)

ds_measures_location

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

Usage
ds_measures_location(data, ..., trim = 0.05)

Arguments
data A data.frame or tibble.
... Column(s) in data.
trim The fraction of values to be trimmed before computing the mean.

Examples
ds_measures_location(mtcars)
ds_measures_location(mtcars, mpg)
ds_measures_location(mtcars, mpg, disp)
ds_measures_symmetry  Measures of symmetry

Description

Returns the measures of symmetry such as skewness and kurtosis.

Usage

ds_measures_symmetry(data, ...)

Arguments

data  A data.frame or tibble.

Examples

ds_measures_symmetry(mtcarz)
ds_measures_symmetry(mtcarz, mpg)
ds_measures_symmetry(mtcarz, mpg, disp)

ds_measures_variation  Measures of variation

Description

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

Usage

ds_measures_variation(data, ...)

Arguments

data  A data.frame or tibble.

Examples

ds_measures_variation(mtcarz)
ds_measures_variation(mtcarz, mpg)
ds_measures_variation(mtcarz, mpg, disp)
### ds_mode

**Mode**

**Description**

Compute the sample mode

**Usage**

```r
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`

**See Also**

`mean`, `median`

**Examples**

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

### ds_percentiles

**Percentiles**

**Description**

Returns the percentiles

**Usage**

```r
ds_percentiles(data, ...)
```
**Arguments**

- `data` A `data.frame` or `tibble`.
- `...` Column(s) in `data`.

**Examples**

```r
ds_percentiles(mtcarz)
ds_percentiles(mtcarz, mpg)
ds_percentiles(mtcarz, mpg, disp)
```

---

**ds_plot_bar**

*Generate bar plots*

**Description**

Creates bar plots if the data has categorical variables.

**Usage**

```r
ds_plot_bar(data, ..., fill = "blue", print_plot = TRUE)
```

**Arguments**

- `data` A `data.frame` or `tibble`.
- `...` Column(s) in `data`.
- `fill` Color of the bars.
- `print_plot` logical; if TRUE, prints the plot else returns a plot object.

**Examples**

```r
ds_plot_bar(mtcarz)
ds_plot_bar(mtcarz, cyl)
ds_plot_bar(mtcarz, cyl, gear)
```
ds_plot_bar_grouped

Generate grouped bar plots

Description

Creates grouped bar plots if the data has categorical variables.

Usage

ds_plot_bar_grouped(data, ..., print_plot = TRUE)

Arguments

data: A data.frame or tibble.
...
print_plot: logical; if TRUE, prints the plot else returns a plot object.

Examples

mt <- dplyr::select(mtcarz, cyl, gear, am)
ds_plot_bar_grouped(mt)
ds_plot_bar_grouped(mtcarz, cyl, gear)

ds_plot_bar_stacked

Generate stacked bar plots

Description

Creates stacked bar plots if the data has categorical variables.

Usage

ds_plot_bar_stacked(data, ..., print_plot = TRUE)

Arguments

data: A data.frame or tibble.
...
print_plot: logical; if TRUE, prints the plot else returns a plot object.

Examples

mt <- dplyr::select(mtcarz, cyl, gear, am)
ds_plot_bar_stacked(mt)
ds_plot_bar_stacked(mtcarz, cyl, gear)
**ds_plot_box_group**  
*Compare distributions*

**Description**

Creates box plots if the data has both categorical & continuous variables.

**Usage**

```r
ds_plot_box_group(data, ..., print_plot = TRUE)
```

**Arguments**

- `data`  
  A data.frame or tibble.

- `...`  
  Column(s) in data.

- `print_plot`  
  logical; if TRUE, prints the plot else returns a plot object.

**Examples**

```r
mt <- dplyr::select(mtcarz, cyl, disp, mpg)  
ds_plot_box_group(mt)  
ds_plot_box_group(mtcarz, cyl, gear, mpg)
```

---

**ds_plot_box_single**  
*Generate box plots*

**Description**

Creates box plots if the data has continuous variables.

**Usage**

```r
ds_plot_box_single(data, ..., print_plot = TRUE)
```

**Arguments**

- `data`  
  A data.frame or tibble.

- `...`  
  Column(s) in data.

- `print_plot`  
  logical; if TRUE, prints the plot else returns a plot object.

**Examples**

```r
ds_plot_box_single(mtcarz)  
ds_plot_box_single(mtcarz, mpg)  
ds_plot_box_single(mtcarz, mpg, disp, hp)
```
**ds_plot_density**

Generate density plots

**Description**

Creates density plots if the data has continuous variables.

**Usage**

```r
ds_plot_density(data, ..., color = "blue", print_plot = TRUE)
```

**Arguments**

- **data**: A `data.frame` or `tibble`.
- **...**: Column(s) in `data`.
- **color**: Color of the plot.
- **print_plot**: logical; if `TRUE`, prints the plot else returns a plot object.

**Examples**

```r
ds_plot_density(mtcarz)
da_plot_density(mtcarz, mpg)
da_plot_density(mtcarz, mpg, disp, hp)
```

---

**ds_plot_histogram**

Generate histograms

**Description**

Creates histograms if the data has continuous variables.

**Usage**

```r
ds_plot_histogram(data, ..., bins = 5, fill = "blue", print_plot = TRUE)
```

**Arguments**

- **data**: A `data.frame` or `tibble`.
- **...**: Column(s) in `data`.
- **bins**: Number of bins in the histogram.
- **fill**: Color of the histogram.
- **print_plot**: logical; if `TRUE`, prints the plot else returns a plot object.
Examples

```r
ds_plot_histogram(mtcarz)
ds_plot_histogram(mtcarz, mpg)
ds_plot_histogram(mtcarz, mpg, disp, hp)
```

---

### ds_plot_scatter

*Generate scatter plots*

#### Description

Creates scatter plots if the data has continuous variables.

#### Usage

```r
ds_plot_scatter(data, ..., print_plot = TRUE)
```

#### Arguments

- `data`: A `data.frame` or `tibble`.
- `...`: Column(s) in `data`.
- `print_plot`: logical; if `TRUE`, prints the plot else returns a plot object.

#### Examples

```r
ds_plot_scatter(mtcarz)
ds_plot_scatter(mtcarz, mpg, disp)
```

---

### ds_range

*Range*

#### Description

Compute the range of a numeric vector

#### Usage

```r
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.
**ds_rindex**

**Value**

Range of x

**See Also**

`range`

**Examples**

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

---

**ds_rindex**

**Index Values**

**Description**

Returns index of values.

**Usage**

```r
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.

**Examples**

```r
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(data)

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

Arguments
data 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.

Examples

# screen data
ds_screener(mtcarz)
Description
Compute the skewness of a probability distribution.

Usage
\[
\text{ds\_skewness}(x, \text{na.rm} = \text{FALSE})
\]

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 \( x \)

References

See Also
kurtosis

Examples
\[
\text{ds\_skewness(mtcars$mpg)}
\]
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, ...)

Arguments
data  A data.frame or tibble.
...  Column(s) in data.

See Also
summary ds_freq_table ds_cross_table

Examples
ds_summary_stats(mtcars, mpg)
**ds_tailobs**

**Tail Observations**

**Description**

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

**Usage**

```r
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* observations are returned

**Details**

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

**Value**

*n* highest/lowest observations from `data`

**See Also**

top_n

**Examples**

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

---

**ds_tidy_stats**

**Tidy descriptive statistics**

**Description**

Descriptive statistics for multiple variables.

**Usage**

```r
ds_tidy_stats(data, ...)
```
Arguments

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

Value

A tibble.

Deprecated Functions

ds_multi_stats() have been deprecated. Instead use ds_tidy_stats().

Examples

ds_tidy_stats(mtcarz)
ds_tidy_stats(mtcarz, mpg, disp, hp)

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
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, 32
mtcarz, 33

Binomial, 3

Chisquare, 4

descriptr, 3
descriptr-package (descriptr), 3
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_auto_cross_table
(ds_auto_freq_table), 8
ds_auto_freq_table, 8
ds_auto_group_summary, 9
ds_auto_summary_stats, 10
ds_cross_table, 10, 13, 30
ds_css, 11
ds_cvar, 12
ds_extreme_obs, 12
ds_freq_cont (ds_freq_table), 13
ds_freq_table, 13, 30
ds_gmean, 14, 16
ds_group_summary, 14, 16
ds_group_summary_interact, 15
ds_hmean, 14, 16
ds_kurtosis, 17
ds_launch_shiny_app, 18
ds_mdev, 18
ds_measures_location, 19
ds_measures_symmetry, 20
ds_measures_variation, 20
ds_mode, 21
ds_multi_stats (ds_tidy_stats), 31
ds_oway_tables (ds_auto_freq_table), 8
ds_percentiles, 21
ds_plot_bar, 22
ds_plot_bar_grouped, 23
ds_plot_bar_stacked, 23
ds_plot_box_group, 24
ds_plot_box_single, 24
ds_plot_density, 25
ds_plot_histogram, 25
ds_plot_scatter, 26
ds_range, 26
ds_rindex, 27
ds_screener, 28
ds_skewness, 29
ds_std_error, 30
ds_summary_stats, 15, 30
ds_tailobs, 31
ds_tidy_stats, 31
ds_tway_tables (ds_auto_freq_table), 8
ds_twoway_table (ds_cross_table), 10

FDist, 6

hsb, 32

mad, 19
mean, 14, 16, 21
median, 21
mtcarz, 33

Normal, 7
INDEX

plot.ds_cross_table(ds_cross_table), 10
plot.ds_freq_table(ds_freq_table), 13
plot.ds_group_summary
  (ds_group_summary), 14
plot.ds_screener(ds_screener), 28

range, 27

summary, 30

TDist, 8

top_n, 31