

Package: tjmisc (via r-universe)

October 29, 2024

Type Package

Title TJ's Miscellany

Version 0.0.0.9000

Description A collection of helper functions.

License GPL-3

URL <https://github.com/tjmahr/tjmisc>

BugReports <https://github.com/tjmahr/tjmisc/issues>

Imports broom, dplyr (>= 1.1.0), ggplot2, glue, Hmisc, magrittr, nlme, purrr, readr, rlang (>= 0.1.6), stringi, stringr, testthat, tibble, tidyr, utils, stats, reshape2, grDevices, rstudioapi, whisker, ids

ByteCompile true

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 4.1)

Repository <https://staffanbetner.r-universe.dev>

RemoteUrl <https://github.com/tjmahr/tjmisc>

RemoteRef HEAD

RemoteSha f7e09bd374ee4740c825d98acce5a216897d0798

Contents

annotate_label_grey	2
compare_pairs	3
compare_sets	3
count_words_in_rmd_file	4
fct_glue_labels	5

ggmatplot	6
ggpreview	6
is_same_as_last	7
jekyll_create_rmd_draft	7
sample_n_of	8
seq_along_rows	9
str_which_between	9
tidy_correlation	10
tidy_quantile	11
tjm_colors	11

Index	12
--------------	-----------

annotate_label_grey *Annotating plots with a grey background*

Description

Annotating plots with a grey background

Usage

```
annotate_label_grey(
  label,
  x,
  y,
  size = 4,
  fill = "#EBEBEB99",
  hjust = 0,
  vjust = 0,
  label.size = 0,
  ...
)
```

Arguments

label	Text to write on the plot.
x, y	x and y positions.
size, fill, hjust, vjust, label.size	Plotting aesthetics that this function handles. They can be overridden.
...	Other parameters to pass onto <code>ggplot2::annotate()</code> .

Value

An annotation layer for a ggplot2 plot.

compare_pairs	<i>Compare pairs of categorical variables</i>
---------------	---

Description

Compare pairs of categorical variables

Usage

```
compare_pairs(data, levels, values, f = `-`)
```

Arguments

data	a dataframe
levels	a column with a categorical variable. All pairs of values in levels will be compared.
values	a column with values to compare.
f	comparison function to apply to values in each pair. Defaults to - to compute the pairwise differences.

Value

a dataframe with pairwise comparisons

Examples

```
to_compare <- nlme::Machines %>%
  dplyr::group_by(Worker) %>%
  dplyr::summarise(avg_score = mean(score)) %>%
  print()

to_compare %>%
  compare_pairs(Worker, avg_score) %>%
  dplyr::rename(difference = value) %>%
  dplyr::mutate_if(is.numeric, round, 1)
```

compare_sets	<i>Compare two vectors using R's set operations</i>
--------------	---

Description

Compare two vectors using R's set operations

Usage

```
compare_sets(x, y)
```

Arguments

`x, y` vectors to compare

Value

a list with lengths (the lengths of the other elements), `x`, `y`, `unique(x)`, `unique(y)`, `setequal(x, y)`, `setdiff(x, y)`, `setdiff(y, x)`, `intersect(x, y)`, `union(x, y)`.

Examples

```
yours <- c(1, 2, 3, 4, 4)
mine <- c(3, 5, 6, 4)
compare_sets(yours, mine)
```

count_words_in_rmd_file

Count words in an Rmarkdown file

Description

These functions strips away code and non-prose elements before counting words.

Usage

```
count_words_in_rmd_file(path)

count_words_in_rmd_lines(lines)

simplify_rmd_lines(lines)
```

Arguments

`path` path to an Rmarkdown file
`lines` a character vector of text (from an Rmarkdown file)

Details

The helper function `simplify_rmd_lines()` strips down an Rmarkdown file so that dubious things do not contribute to the word count. It does the following.

1. Remove all lines that fall between a pair of `````` lines. (These are used sometimes to show verbatim text from blocks with three tick marks).
2. Remove all lines that fall between a pair of ````` lines.
3. Lines that end with `\r` are merged with the following line.
4. Inline code spans are replaced with a single word (``code``).
5. Single-line HTML comments are deleted.

These steps are very ad hoc, updated and expanded as I run into new things that need to be excluded from my word counts. Let's not pretend that this thing is at all comprehensive.

The word-count is computed by `stringi::stri_stats_latex()`.

Value

a data-frame with the counts of word, characters in words, and whitespace characters. `simplify_rmd_lines()` returns a character vector of simplified Rmarkdown lines.

fct_glue_labels	<i>Format the labels of a factor</i>
-----------------	--------------------------------------

Description

Format the labels of a factor

Usage

```
fct_glue_labels(xs, fmt = "{levels}", first_fmt = fmt)
```

```
fct_add_counts(xs, fmt = "{levels} ({counts})", first_fmt = fmt)
```

Arguments

xs	a factor
fmt	glue-style format to use. Defaults to "{levels}" for <code>fct_glue_labels()</code> and "{levels} ({counts})" for <code>fct_add_counts()</code> .
first_fmt	glue-style format to use for very first label. Defaults to value of <code>fmt</code> .

Details

At this point, only the magic variables "{levels}" and "{counts}" are available ". In principle, others could be defined. `fct_add_counts()` is a special case of `fct_glue_labels()`.

Value

a factor with the labels updated

ggmatplot	<i>Plot columns of a matrix</i>
-----------	---------------------------------

Description

Creates plots of matrices like `graphics::matplot()` but uses `ggplot2`, defaults to drawing lines, and can specify a column to use for the *x*-axis.

Usage

```
ggmatplot(x, x_axis_column = NULL, n_colors = 6, unique_rows = TRUE)
```

Arguments

<code>x</code>	A matrix.
<code>x_axis_column</code>	Index (number) of the column to plot for the <i>x</i> -axis. Defaults to <code>NULL</code> in which case it uses row index (number) as the <i>x</i> -axis.
<code>n_colors</code>	Number of colors to cycle through. Defaults to 6.
<code>unique_rows</code>	Whether to work first take the unique rows of the matrix. Defaults to <code>TRUE</code> .

Value

a `ggplot2` plot.

ggpreview	<i>Preview a file that would be created by ggsave()</i>
-----------	---

Description

This function saves a plot to a temporary file with `ggsave()` and opens the temporary file in the system viewer. This function is useful for quickly previewing how a plot will look when it is saved to a file.

Usage

```
ggpreview(..., device = "png")
```

Arguments

<code>...</code>	options passed onto <code>ggplot2::ggsave()</code>
<code>device</code>	the file extension of the device to use. Defaults to <code>"png"</code> .

is_same_as_last	<i>Check for locally repeating values</i>
-----------------	---

Description

Check for locally repeating values

Usage

```
is_same_as_last(xs)
replace_if_same_as_last(xs, replacement = "")
```

Arguments

xs a vector
replacement a value used to replace a repeated value. Defaults to "".

Value

is_same_as_last() returns TRUE when xs[n] the same as xs[n-1].

Examples

```
xs <- c("a", "a", "a", NA, "b", "b", "c", NA, NA)
is_same_as_last(xs)
replace_if_same_as_last(xs, "")
```

jekyll_create_rmd_draft	<i>Create a Jekyll draft post</i>
-------------------------	-----------------------------------

Description

This is the function I use to create new posts for my website.

Usage

```
jekyll_create_rmd_draft(
  slug = NULL,
  date = NULL,
  dir_drafts = "../_R/_drafts",
  open = TRUE
)
```

Arguments

slug	A "slug" to use for the post. Should be a string consisting of "hyphen-separated-content-words". Defaults to NULL in which case a random slug is created.
date	Date string to use for the post. Default to NULL for the current date format (Sys.Date()).
dir_drafts	Relative path to the folder to store the drafts. Defaults to ". /_R/_drafts".
open	Whether to open the file for editing when using RStudio. Defaults to TRUE.

Value

The path to the created file is invisibly returned.

sample_n_of	<i>Randomly sample data from n sub-groups of data</i>
-------------	---

Description

Randomly sample data from n sub-groups of data

Usage

```
sample_n_of(data, size, ...)
```

Arguments

data	a dataframe
size	number of groups to sample
...	variables to group by

Value

the data from subgroups

Examples

```
sample_data <- tibble::tibble(
  letter = rep(letters, 5),
  color = rep(c("red", "green", "yellow", "orange", "blue"), 26),
  value = rnorm(26 * 5)
)

# data from two letters
sample_data %>%
  sample_n_of(2, letter)

# data from two colors
sample_data %>%
  sample_n_of(2, color)
```



```
# data from 10 letter-colors pairs
sample_data %>%
  sample_n_of(10, letter, color)
```

seq_along_rows	<i>Create a sequence along the rows of a dataframe</i>
----------------	--

Description

Create a sequence along the rows of a dataframe

Usage

```
seq_along_rows(data)
```

Arguments

data a dataframe

Value

a sequence of integers along the rows of a dataframe

str_which_between	<i>Which lines fall in between a delimiter pattern</i>
-------------------	--

Description

Which lines fall in between a delimiter pattern

Usage

```
str_which_between(string, pattern)
```

Arguments

string a character vector
pattern a regular expression pattern to look for

Value

the lines that are contained between pairs of delimiter patterns

Examples

```
string <- "
```{r}
some code
```
```

Here is more code.

```
```markdown
bold!
```
```

```
lines <- unlist(strsplit(string, "\n"))
str_which_between(lines, "```")
```

| | |
|------------------|-----------------------------------|
| tidy_correlation | <i>Generate tidy correlations</i> |
|------------------|-----------------------------------|

Description

This function respects groupings from `dplyr::group_by()`. When the dataframe contains grouped data, the correlations are computed within each subgroup of data.

Usage

```
tidy_correlation(data, ..., type = c("pearson", "spearman"))
```

Arguments

| | |
|------|--|
| data | a dataframe |
| ... | columns to select, using <code>dplyr::select()</code> semantics. |
| type | type of correlation, either "pearson" (the default) or "spearman". |

Value

a long dataframe (a tibble) with correlations calculated for each pair of columns.

Examples

```
tidy_correlation(ChickWeight, -Chick, -Diet)
```

```
tidy_correlation(ChickWeight, weight, Time)
```

```
ChickWeight %>%
  dplyr::group_by(Diet) %>%
  tidy_correlation(weight, Time)
```

| | |
|---------------|---|
| tidy_quantile | <i>Generate tidy quantiles for a dataframe column</i> |
|---------------|---|

Description

This function respects groupings from `dplyr::group_by()`. When the dataframe contains grouped data, the quantiles are computed within each subgroup of data.

Usage

```
tidy_quantile(data, var, probs = seq(0.1, 0.9, 0.2))
```

Arguments

| | |
|-------|---|
| data | a dataframe |
| var | a column in the dataframe |
| probs | quantiles to return. Defaults to <code>c(.1, .3, .5, .7, .9)</code> |

Value

a long dataframe (a tibble) with quantiles for the variable.

Examples

```
tidy_quantile(sleep, extra)

sleep %>%
  dplyr::group_by(group) %>%
  tidy_quantile(extra)
```

| | |
|------------|----------------------|
| tjm_colors | <i>Colors I like</i> |
|------------|----------------------|

Description

Colors I like

Usage

```
tjm_colors
```

Format

An object of class `list` of length 8.

Index

* datasets

tjm_colors, 11

annotate_label_grey, 2

compare_pairs, 3

compare_sets, 3

count_words_in_rmd_file, 4

count_words_in_rmd_lines
(count_words_in_rmd_file), 4

fct_add_counts(fct_glue_labels), 5

fct_glue_labels, 5

ggmatplot, 6

ggplot2::annotate(), 2

ggplot2::ggsave(), 6

ggpreview, 6

graphics::matplot(), 6

is_same_as_last, 7

jekyll_create_rmd_draft, 7

replace_if_same_as_last
(is_same_as_last), 7

sample_n_of, 8

seq_along_rows, 9

simplify_rmd_lines
(count_words_in_rmd_file), 4

str_which_between, 9

stringi::stri_stats_latex(), 5

tidy_correlation, 10

tidy_quantile, 11

tjm_colors, 11