

Package ‘crimeutils’

February 3, 2021

Title A Comprehensive Set of Functions to Clean, Analyze, and Present Crime Data

Version 0.3.0

Description A collection of functions that make it easier to understand crime (or other) data, and assist others in understanding it. The package helps you read data from various sources, clean it, fix column names, and graph the data.

Depends R (>= 2.10)

Imports dplyr, ggplot2, readr, gridExtra, scales, magrittr, gt, grDevices, tidyr, stats, methods, rlang

License MIT + file LICENSE

LazyData TRUE

URL <https://github.com/jacobkap/crimeutils/>

BugReports <https://github.com/jacobkap/crimeutils/issues/>

RoxygenNote 7.1.1

Suggests spelling, testthat (>= 2.1.0), covr

Language en-US

NeedsCompilation no

Author Jacob Kaplan [aut, cre] (<<https://orcid.org/0000-0002-0601-0387>>)

Maintainer Jacob Kaplan <jkkaplan6@gmail.com>

Repository CRAN

Date/Publication 2021-02-03 14:40:02 UTC

R topics documented:

capitalize_words	2
indicate_outliers	3
make_average_graph	4
make_barplots	5
make_desc_stats_table	5
make_latex_tables	6

make_mean_std_dev_by_group_table	7
make_n_and_percent_table	8
make_regression_graph	9
make_regression_table	9
make_state_abb	10
make_stat_count_plots	10
pad_decimals	11
scale_color_crim	12
scale_fill_crim	12
scale_linetype_crim	13
scatterplot_data_graph	13
theme_crim	14
time_series_data_graph	15
ucr_constant_reporter_oris	15

Index	17
--------------	-----------

capitalize_words	<i>Capitalizes the first letter of every word</i>
------------------	---

Description

Capitalizes the first letter of every word

Usage

```
capitalize_words(words, lowercase_of = TRUE)
```

Arguments

words	A string or vector of strings with words you want capitalized
lowercase_of	If TRUE (default), keeps the string " of " to be lowercased as is custom in English writing (e.g. District of Columbia).

Value

The original string with the first letter of each word capitalized

Examples

```
capitalize_words("district of columbia")
```

indicate_outliers	<i>Creates new columns to indicate which values are outliers based on the average value.</i>
-------------------	--

Description

Creates new columns to indicate which values are outliers based on the average value.

Usage

```
indicate_outliers(  
  data,  
  select_columns = NULL,  
  group_variable,  
  std_dev_value = 1.96,  
  zero_is_outlier = FALSE  
)
```

Arguments

data	A data.frame
select_columns	A string or vector of strings with the name(s) of the numeric columns to check for outliers. If NULL (default), will use all numeric columns in the data.
group_variable	A string with the name of the column with the grouping variable.
std_dev_value	A number indicating how many standard deviations away from the mean to determine if a value is an outlier.
zero_is_outlier	If TRUE (not default), reports any zero value as an outlier.

Value

The initial data.frame with new columns for each numeric variable included with a value of 0 if not an outlier and 1 if that row is an outlier.

Examples

```
indicate_outliers(mtcars, "drat", group_variable = "am")  
indicate_outliers(mtcars, "drat", group_variable = "am", zero_is_outlier = TRUE)
```

make_average_graph *Create a line graph with 95% confidence interval bars*

Description

Create a line graph with 95% confidence interval bars

Usage

```
make_average_graph(  
  data,  
  x_col,  
  y_col,  
  confidence_interval_errorBars = TRUE,  
  mean_line = TRUE,  
  type = c("line", "bar")  
)
```

Arguments

data	A data.frame with the data you want to graph
x_col	A string with the name of the x-axis column
y_col	A string with the name of the y-axis column
confidence_interval_errorBars	A boolean (default TRUE) for whether to include 95% confidence intervals or not.
mean_line	If TRUE (default) will add a dashed line with the overall mean.
type	A string for whether it should make a line graph ("line", default) or a bar graph ("bar")

Value

A ggplot object. Also prints the graph to the Plots panel.

Examples

```
data = data.frame(x = sample(15:25, size = 200, replace = TRUE),  
y = sample(1:100, size = 200, replace = TRUE))  
make_average_graph(data, "x", "y")  
make_average_graph(data, "x", "y", confidence_interval_errorBars = FALSE)  
make_average_graph(data, "x", "y", type = "bar", mean_line = FALSE)  
make_average_graph(data, "x", "y", confidence_interval_errorBars = FALSE, type = "bar")
```

make_barplots	<i>Make a nice-looking barplot.</i>
---------------	-------------------------------------

Description

Make a nice-looking barplot.

Usage

```
make_barplots(data, column, count = TRUE, title = NULL, ylab = NULL)
```

Arguments

data	A data.frame with the data you want to graph.
column	A string with the name of the column you want to make the plot from.
count	A boolean (default TRUE) indicating if you want the barplot to show a count of the column values or a percent.
title	A string with the text you want as the title.
ylab	A string with the text you want as the y-axis label.

Value

A barplot object.

Examples

```
make_barplots(mtcars, "cyl")  
  
make_barplots(mtcars, "cyl", count = FALSE, title = "hello", ylab = "YLAB Label")
```

make_desc_stats_table	<i>Create a descriptive statistics table from numeric variables</i>
-----------------------	---

Description

Create a descriptive statistics table from numeric variables

Usage

```
make_desc_stats_table(  
  data,  
  columns,  
  output = c("min", "median", "mean", "sd", "max", "sum", "NAs"),  
  decimals = 2,  
  title = NULL,  
  subtitle = NULL,  
  footnote = NULL  
)
```

Arguments

data	A data.frame with the data you want to make the table from.
columns	A string or vector of strings with the names of the columns you want to use.
output	A string or vector of strings indicating which math functions you want to perform on the columns and present in the table. Options are: 'min', 'median', 'mean', 'sd', 'max', and 'N'. Default is to use all of these math functions. The order you put in these values is the order the table will present the columns.
decimals	A positive integer for how many decimal places you want to round to.
title	A string with the text you want as the title
subtitle	A string with the text you want as the subtitle.
footnote	A string with the text you want as the footnote.

Value

A data.frame with the data that generates the table, which is outputted in the Viewer tab.

Examples

```
make_desc_stats_table(mtcars, columns = c("mpg", "disp", "wt", "cyl"))

make_desc_stats_table(mtcars, c("mpg", "disp", "wt"), output = c("mean", "min"),
decimals = 4, title = "hello", subtitle = "world")
```

make_latex_tables	<i>Creates a .tex file with LaTeX code to create a table from an R data.frame.</i>
-------------------	--

Description

Creates a .tex file with LaTeX code to create a table from an R data.frame.

Usage

```
make_latex_tables(
  data,
  file,
  caption = "",
  label = "",
  multi_column = NULL,
  footnote = "",
  sideways = FALSE
)
```

Arguments

data	A data.frame or a list of data.frames. If a data.frame, the table is created with the values in that data.frame. If a list of data.frames, the table gets one panel for each data.frame. If the list is named, will use the names to create panel labels.
file	A string with the name of the file to save the .tex as.
caption	(Optional) A string with the caption for the table (i.e. the table title).
label	(Optional) A string with the reference for the table - to be used when referencing the table in the text. If NULL,
multi_column	(Optional) A named vector with the names being the names of the multi-column and the values being the width of the multi-column.
footnote	(Optional) A string with text for the footnote of the table.
sideways	(Optional) If TRUE, will make a sideways table (useful for large tables), otherwise (default) will make a normal table.

Value

Nothing. It will create a .tex file in the current working directory.

Examples

```
## Not run:
make_latex_tables(mtcars, file = "text.tex", caption = "This is a description of the table",
label = "internal_table_label", footnote = "Here is some info you should know to read this table")

## End(Not run)
```

```
make_mean_std_dev_by_group_table
```

Get mean and standard deviation of variables by group

Description

Get mean and standard deviation of variables by group

Usage

```
make_mean_std_dev_by_group_table(data, group_column, columns, total_row = TRUE)
```

Arguments

data	A data.frame with the data you want to make the table from.
group_column	A string with the name of the variable you are grouping by
columns	A string or vector of strings for the variables you want to get the mean and standard deviation for.
total_row	A boolean (default TRUE) for whether to include a row at the bottom for the overall mean and standard deviation (i.e. not by group).

Value

A data.frame with the first column showing the category grouped by. Then one column for each variable you want the mean and standard deviation for. Will give the mean and standard deviation as a single string with the standard deviation in parentheses.

Examples

```
make_mean_std_dev_by_group_table(mtcars, "gear", c("mpg", "disp"))
```

```
make_n_and_percent_table
```

Make a table showing the number (n) and percent of the population (e.g. % of nrow()) for each value in a variable(s).

Description

Make a table showing the number (n) and percent of the population (e.g. % of nrow()) for each value in a variable(s).

Usage

```
make_n_and_percent_table(data, columns)
```

Arguments

data	A data.frame with the data you want to make the table from.
columns	A string or vector of strings with the column names to make the N and % from.

Value

A data.frame with one row for each value in the inputted variable(s) and columns showing the N and % for that value.

Examples

```
make_n_and_percent_table(mtcars, c("cyl", "gear"))
```

`make_regression_graph` *Make a graph of coefficient values and 95 percent confidence interval for regression.*

Description

Make a graph of coefficient values and 95 percent confidence interval for regression.

Usage

```
make_regression_graph(model, coefficients = NULL)
```

Arguments

`model` A 'lm' object made from making a model using 'lm()'.
`coefficients` A string or vector of strings with the coefficient names. Will then make the graph only with those coefficients.

Value

Outputs a 'ggplot2' graph

Examples

```
make_regression_graph(model = lm(mpg ~ cyl + disp + hp + drat, data = mtcars))
make_regression_graph(model = lm(mpg ~ cyl + disp + hp + drat, data = mtcars),
  coefficients = c("cyl", "disp"))
make_regression_graph(model = lm(mpg ~ cyl + disp, data = mtcars))
```

`make_regression_table` *Turns regression results in a data.frame for easy conversion to a table*

Description

Turns regression results in a data.frame for easy conversion to a table

Usage

```
make_regression_table(model, coefficients_only = TRUE)
```

Arguments

`model` A 'lm' object made from making a model using 'lm()'.
`coefficients_only` If TRUE (default), returns only the coefficients, standard error, t-value, p-value, and confidence intervals. Else also returns the r-squared, the adjusted r-squared, f-stat, p-value for the f-stat, and the degrees of freedom.

Value

A data.frame with the regression results

Examples

```
make_regression_table(lm(mpg ~ cyl, data = mtcars))
make_regression_table(lm(mpg ~ cyl, data = mtcars), coefficients_only = FALSE)
```

make_state_abb	<i>Returns abbreviations of state name input.</i>
----------------	---

Description

Returns abbreviations of state name input.

Usage

```
make_state_abb(state)
```

Arguments

state A vector of strings with the names of US states.

Value

A vector of strings with the abbreviations of the inputted state names.

Examples

```
make_state_abb("california")
```

make_stat_count_plots	<i>Make a nice-looking stat_count (similar to barplot) plot.</i>
-----------------------	--

Description

Make a nice-looking stat_count (similar to barplot) plot.

Usage

```
make_stat_count_plots(
  data,
  column,
  count = TRUE,
  title = NULL,
  ylab = NULL,
  xlab = NULL
)
```

Arguments

data	A data.frame with the data you want to graph.
column	A string with the name of the column you want to make the plot from.
count	A boolean (default TRUE) indicating if you want the barplot to show a count of the column values or a percent.
title	A string with the text you want as the title.
ylab	A string with the text you want as the y-axis label.
xlab	A string with the text you want as the x-axis label.

Value

A stat_count object

Examples

```
make_stat_count_plots(mtcars, "mpg")
make_stat_count_plots(mtcars, "mpg", count = FALSE, title = "hello", ylab = "YLAB Label")
```

pad_decimals	<i>Pad decimal places with trailing zeros.</i>
--------------	--

Description

Pad decimal places with trailing zeros.

Usage

```
pad_decimals(numbers, digits = NULL)
```

Arguments

numbers	A number or vector of numbers.
digits	Number of decimal places to pad. If NULL (default), uses the maximum number of decimal places in the numbers input. If digits is less than the number of decimal places in the data, rounds the data to the decimal place specified. If rounding at a 5, follows R's rules to round to the nearest even number.

Value

The original numbers, now as strings with trailing zeros added to the decimal places.

Examples

```
pad_decimals(c(2, 3.4, 8.808))
```

scale_color_crim *A set of colorblind friendly colors for graphs.*

Description

A set of colorblind friendly colors for graphs.

Usage

```
scale_color_crim(...)
```

Arguments

... Arguments passed to discrete_scale()

Value

The ggplot graph with colors set.

Examples

```
ggplot2::ggplot(mtcars, ggplot2::aes(x = mpg, y = hp, color = as.character(cyl))) +  
  ggplot2::geom_point(size = 2) +  
  scale_color_crim()
```

scale_fill_crim *A set of colorblind friendly fill colors for graphs.*

Description

A set of colorblind friendly fill colors for graphs.

Usage

```
scale_fill_crim(...)
```

Arguments

... Arguments passed to discrete_scale()

Value

The ggplot graph with fills set.

Examples

```
ggplot2::ggplot(mtcars, ggplot2::aes(x = cyl, fill = as.character(cyl))) +  
  ggplot2::geom_bar() +  
  scale_fill_crim()
```

scale_linetype_crim *A set of linetypes*

Description

A set of linetypes

Usage

```
scale_linetype_crim(...)
```

Arguments

... Arguments passed to `discrete_scale()`

Value

The ggplot graph with linetypes set.

Examples

```
ggplot2::ggplot(mtcars, ggplot2::aes(x = mpg, y = hp, linetype = as.character(cyl))) +  
  ggplot2::geom_line(size = 1) +  
  scale_linetype_crim() +  
  theme_crim()
```

scatterplot_data_graph

Create a PDF with one scatterplot for each group in the data.

Description

Create a PDF with one scatterplot for each group in the data.

Usage

```
scatterplot_data_graph(  
  data,  
  numeric_variable1,  
  numeric_variable2,  
  group_variable,  
  file_name  
)
```

Arguments

`data` A data.frame with the data you want to graph.
`numeric_variable1` A string with the name of the first column with numeric data to graph.
`numeric_variable2` A string with the name of the second column with numeric data to graph.
`group_variable` A string with the name of the column with the grouping variable.
`file_name` A string with the name of the PDF to be made with one page for each graph.

Value

A PDF with one page per graph

Examples

```
## Not run:  
scatterplot_data_graph(mtcars, numeric_variable1 = "mpg", numeric_variable2 = "disp",  
group_variable = "gear", file_name = "test.pdf")  
  
## End(Not run)
```

theme_crim

A minimalist theme designed for graphics in academic research

Description

A minimalist theme designed for graphics in academic research

Usage

```
theme_crim()
```

Value

The graph with the theme changed.

Examples

```
ggplot2::ggplot(mtcars) +  
ggplot2::geom_point(ggplot2::aes(x = wt, y = mpg)) +  
theme_crim()
```

`time_series_data_graph`*Create a PDF with one time-series graph for each group in the data.*

Description

Create a PDF with one time-series graph for each group in the data.

Usage

```
time_series_data_graph(  
  data,  
  numeric_variable,  
  time_variable,  
  group_variable,  
  outlier_std_dev_value = 1.96,  
  file_name  
)
```

Arguments

`data` A data.frame with the data you want to graph.

`numeric_variable` A string with the name of the column with numeric data to graph.

`time_variable` A string with the name of the column that contains the time variable.

`group_variable` A string with the name of the column with the grouping variable.

`outlier_std_dev_value` A number that indicates how many standard deviations from the group mean an outlier is. Outliers will be colored orange in the data.

`file_name` A string with the name of the PDF to be made with one page for each graph.

Value

A PDF with one page per graph

`ucr_constant_reporter_oris`*Get ORIs that consistently report their data every year.*

Description

Get ORIs that consistently report their data every year.

Usage

```
ucr_constant_reporter_oris(data, minimum_months_reported)
```

Arguments

`data` A data.frame with Uniform Crime Report (UCR) data. Requires at least the ORI, year, and number_of_months_reported columns.

`minimum_months_reported` Integer indicating the minimum number of months requesting to keep in data.

Value

A vector with the ORIs that report the minimum number of months for every year in the data.

Index

capitalize_words, 2

indicate_outliers, 3

make_average_graph, 4

make_barplots, 5

make_desc_stats_table, 5

make_latex_tables, 6

make_mean_std_dev_by_group_table, 7

make_n_and_percent_table, 8

make_regression_graph, 9

make_regression_table, 9

make_stat_count_plots, 10

make_state_abb, 10

pad_decimals, 11

scale_color_crim, 12

scale_fill_crim, 12

scale_linetype_crim, 13

scatterplot_data_graph, 13

theme_crim, 14

time_series_data_graph, 15

ucr_constant_reporter_oris, 15