Package: RplotterPkg (via r-universe)

March 6, 2025

Type Package

Title R Plotting Functions Using 'ggplot2'

Version 0.1.3

Description Makes it easy to produce everyday 'ggplot2' charts in a functional way without an extensive ``tree" implementation. The package includes over 15 functions for the production and arrangement of basic graphing.

License MIT + file LICENSE

URL https://github.com/deandevl/RplotterPkg

BugReports https://github.com/deandevl/RplotterPkg/issues

Depends R (>= 4.4.0)

Imports ggplot2 (>= 3.5.1), data.table (>= 1.16.4), rlang (>= 1.1.4), gt (>= 0.11.1), gtable (>= 0.3.6), glue (>= 1.8.0), purrr (>= 1.0.2), ggplotify (>= 0.1.2), aplpack (>= 1.3.5), methods

Suggests knitr (>= 1.49), RColorBrewer (>= 1.1-3), rmarkdown (>= 2.29), testthat (>= 3.0.0), vdiffr (>= 1.0.8), here (>= 1.0.1), socviz (>= 1.2), tsbox (>= 0.4.1), gapminder (>= 0.3.0), palmerpenguins (>= 0.1.1), spData (>= 2.3.3), sf (>= 1.0-19), raster (>= 3.6-30), elevatr (>= 0.99.0), maps (>= 3.4.1)

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

Config/testthat/edition 3

Config/pak/sysreqs make libxml2-dev libssl-dev libnode-dev

Repository https://deandevl.r-universe.dev

RemoteUrl https://github.com/deandevl/RplotterPkg

RemoteRef HEAD

RemoteSha 734cf4b6ca0d5f78e02da64d8bcd59ca6c0ac01e

Contents

air_passengers *air_passengers data*

Description

Data where the base `datasets::AirPassengers` monthly time series of international airline passenger numbers 1949-1960 was converted to a data frame using `tsbox::ts_df(datasets::AirPassengers)`.

time	Date with format 1949-01-01
value	numeric number of passengers in thousands

Usage

```
air_passengers
```

Format

An object of class `data.frame` with 144 rows and 2 columns.

Source

`datasets::AirPassengers`

boston_marathon *boston_marathon data*

Description

Boston Marathon completion times of women of different ages. The raw tab delimited data (`boston_marathon.txt`) is read by `data.table::fread()` to produce a data frame with columns for women's "age" and marathon "time" in minutes.

age	numeric age from 20 to 60
time	numeric marathon time

Usage

```
boston_marathon
```

Format

An object of class `data.table` (inherits from `data.frame`) with 109 rows and 2 columns.

Source

LearnEDAfunctions

car_stats *car_stats data*

Description

From `datasets::mtcars` produces statistics for N and the means of "mpg", "hp", "wt", "disp" variables grouped by "cyl".

N	numeric of group's total observations
mean_mpg	numeric of group's mean mpg
mean_hp	numeric of group's mean horsepower
mean_wt	numeric of group's mean weight
mean_disp	numeric of group's mean displacment

Usage

```
car_stats
```

Format

An object of class `data.table` (inherits from `data.frame`) with 3 rows and 6 columns.

Source

`datasets::mtcars`

chick_weights *chick_weights data*

Description

A dataframe with 4 observations of average chick weights.

Diet	numeric diet index
Weight	numeric of average chick's weight
Label	string label for the diet in percent of protein

Usage

```
chick_weights
```

Format

An object of class `data.table` (inherits from `data.frame`) with 4 rows and 3 columns.

Source

`datasets::ChickWeight`

create_bar_plot *create_bar_plot*

Description

Function wraps ggplot2 geom_bar and geom_col to creates a bar plot.

Usage

```
create_bar_plot(
  df = NULL,
  aes_x = NULL,
  aes_y = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  position = "stack",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  bar_fill = NA,
  bar_color = "black",
  bar_alpha = 1,
  bar_lwd = 0.7,
  bar_width = NULL,
  x_major_breaks = NULL,
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  axis_text_size = 11,
  do_coord_flip = FALSE,
  order_bars = NULL,
  bar_labels = FALSE,
  bar_label_sz = 4,
  bar_label_color = "black",
  bar_label_fontface = "plain",
  bar_label_alpha = 1,
  show_major_grids = TRUE,
  show_minor_grids = TRUE,
  panel_color = "white",
```

```
  panel_border_color = "black",
  show_legend = TRUE,
  legend_pos = "right",
  legend_key_width = 0.7,
  legend_key_height = 0.7,
  legend_key_backgrd = "white",
  silent_NA_warning = FALSE,
  png_file_path = NULL,
  png_width_height = c(480, 480)
)
```

Arguments

df	The target data frame for the bar chart.
aes_x	A required string that sets the x axis discrete variable name from 'df'.
aes_y	Sets the y axis numeric variable name from 'df'. If this variable is NULL the bar heights will be proportional to the number of cases in the levels of 'aes_x'. If 'aes_y' is not NULL, the bar heights represent this parameter's values across the 'aes_x' levels.
aes_color	Sets the variable name from 'df' for the aesthetic mapping for color.
aes_fill	Sets the variable name from 'df' for the aesthetic mapping for fill.
position	A string that sets the bar positions. Acceptable values are "dodge"(side by side), "identity"(overlap) or "stack".
title	A string that sets the overall title.
subtitle	A string that sets the overall subtitle.
caption	A string that sets the caption.
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic label. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
bar_fill	A string that sets the fill color attribute for the bars.
bar_color	A string that sets the outline color attribute for the bars.
bar_alpha	A numeric that sets the alpha component attribute to 'bar_color'.
bar_lwd	A numeric that sets the bar's outline line width attribute.

`bar_width` A numeric that sets the width attribute of the bars.

`x_major_breaks` If 'aes_x' is numeric then this parameter is a numeric vector that defines the major breaks/intervals for the x axis. Interval labels are created and their respective counts are displayed.

`y_limits` A numeric 2 element vector that sets the minimum and maximum for the y axis. Use `NA` to refer to the existing minimum and maximum.

`y_major_breaks` A numeric vector or function that defines the exact major tic locations along the y axis.

`y_minor_breaks` A numeric vector or function that defines the exact minor tic locations along the y axis.

`y_labels` A character vector with the same length as 'y_major_breaks', that labels the major tics.

`axis_text_size` A numeric that sets the font size along the axis'. Default is 11.

`do_coord_flip` A logical which if `TRUE` will flip the x and y axis'.

`order_bars` A string which will order the bars in a specific direction. Acceptable values are "asc" or "desc"

`bar_labels` A logical which if `TRUE` will label each bar with its value.

`bar_label_sz` A numeric that sets the size of the label.

`bar_label_color`
A string that sets the label's color.

`bar_label_fontface`
A string that sets the label's font face. Acceptable values are "plain", "bold", "italic", "bold.italic".

`bar_label_alpha`
A numeric that sets the label's alpha value.

`show_major_grids`
A logical that controls the appearance of major grids.

`show_minor_grids`
A logical that controls the appearance of minor grids.

`panel_color` A string in hexidecimal or color name that sets the plot panel's color. The default is "white".

`panel_border_color`
A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

`show_legend` A logical that controls the appearance of the legend.

`legend_pos` A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

`legend_key_width`
A numeric that sets the legend width in cm.

`legend_key_height`
A numeric that sets the legend height in cm.

`legend_key_backgrd`
A string that sets the legend's background color.

silent_NA_warning
: A logical that controls the appearance of a console warning when Na's are removed.

png_file_path
: A character string with the directory and file name to produce a png image of the plot.

png_width_height
: A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(data.table)
library(RColorBrewer)
library(RplotterPkg)

RplotterPkg::create_bar_plot(
  df = RplotterPkg::religion,
  aes_x = "happy",
  aes_fill = "religion",
  position = "dodge",
  title = "Happy Religions",
  center_titles = TRUE,
  rot_y_tic_label = TRUE,
  bar_width = 0.8,
  order_bars = "desc",
  x_title = "Happiness",
  y_title = "Count",
  axis_text_size = 16
) +
ggplot2::scale_fill_discrete(
  type = RColorBrewer::brewer.pal(n = 9, name = "Set1")
)
```

---

create_box_plot *create_box_plot*

Description

Function wraps around ggplot2 geom_boxplot to create a box plot.

Usage

```
create_box_plot(
  df = NULL,
  aes_x = NULL,
  aes_y = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  aes_label = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  box_fill = NA,
  box_color = "black",
  box_line_width = 0.5,
  box_alpha = 1,
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_decimals = 0,
  y_scientific = FALSE,
  axis_text_size = 11,
  do_coord_flip = FALSE,
  show_legend = TRUE,
  panel_color = "white",
  panel_border_color = "black",
  legend_pos = "top",
  legend_key_width = 0.7,
  legend_key_height = 0.7,
  legend_key_backgrd = "white",
  show_major_grids = TRUE,
  show_minor_grids = TRUE,
  ol_color = "black",
  ol_fill = "black",
  ol_size = 1.5,
  ol_shape = 19,
  ol_stroke = 0.5,
  ol_alpha = NULL,
  order_by_median = NULL,
  silent_NA_warning = FALSE,
  png_file_path = NULL,
  png_width_height = c(480, 480)
```

```
)
```

Arguments

df	The target data frame for the density chart.
aes_x	An optional factor variable name from 'df' that sets the x axis variable. Multiple box plots will be displayed along the x axis if this argument is set.
aes_y	A required variable name from 'df' that sets the y axis variable.
aes_color	Sets the variable name from 'df' for the aesthetic mapping for color.
aes_fill	Sets the variable name from 'df' for the aesthetic mapping for fill.
aes_label	Sets the variable name from 'df' whose value will to be displayed corresponding to the 'aes_y' outliers.
title	A string that sets the overall title.
subtitle	A string that sets the overall subtitle
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
box_fill	A string that sets the fill color attribute for the box plot.
box_color	A string that sets the color attribute for the box plot.
box_line_width	A numeric that sets the size attribute of the box line width.
box_alpha	A numeric that set the alpha component attribute to 'box_color'.
y_limits	A numeric 2 element vector that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.
y_major_breaks	A numeric vector or function that defines the exact major tic locations along the y axis.
y_minor_breaks	A numeric vector or function that defines the exact minor tic locations along the y axis.
y_decimals	A numeric that sets the number of decimal places for y-tic labels.
y_scientific	A logical which if TRUE will put the y-tic labels in scientific notation.
axis_text_size	A numeric that sets the font size along the axis'. Default is 11.

`do_coord_flip` A logical which if `TRUE` will flip the x and y axis'.

`show_legend` A logical that controls the appearance of the legend.

`panel_color` A string in hexidecimal or color name that sets the plot panel's color. The default is "white".

`panel_border_color`
A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

`legend_pos` A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

`legend_key_width`
A numeric that sets the legend width in cm.

`legend_key_height`
A numeric that sets the legend height in cm.

`legend_key_backgrd`
A string that sets the legend's background color. The default is "white".

`show_major_grids`
A logical that controls the appearance of major grids.

`show_minor_grids`
A logical that controls the appearance of minor grids.

`ol_color` A string that sets the outlier color.

`ol_fill` A string that sets the outlier fill.

`ol_size` A numeric that set the outlier size.

`ol_shape` A string that set the outlier shape.

`ol_stroke` A numeric that sets the outlier shape line width.

`ol_alpha` A numeric that sets the outlier alpha for color.

`order_by_median`
A string which will order the plot of 'aes_x' factor levels/categories by the 'aes_y' medians. Acceptable values are "asc" or "desc". Note that the optional argument 'aes_x' must not be `NULL`.

`silent_NA_warning`
A logical that controls the appearance of a console warning when Na's are removed.

`png_file_path` A character string with the directory and file name to produce a png image of the plot.

`png_width_height`
A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(data.table)
library(RplotterPkg)

RplotterPkg::create_box_plot(
  df = RplotterPkg::organdata,
  aes_x = "country",
  aes_y = "donors",
  aes_label = "donors",
  order_by_median = "desc",
  y_limits = c(5,35),
  y_major_breaks = seq(5, 35, 5),
  title = "Organ Donation Rate per Million",
  subtitle = "Showing outlier rates",
  x_title = "Country",
  y_title = "Donor Rate",
  do_coord_flip = TRUE,
  box_color = "purple",
  box_line_width = 0.8,
  rot_y_tic_label = TRUE,
  ol_color = "red",
  ol_size = 1.5
)
```

---

create_density_plot *create_density_plot*

Description

Function creates a ggplot2 based density plot with options for scaling, shading probability areas, and plotting observation locations. The function's density arguments mirror most of the arguments available from `density` for the Kernel Density Estimation (KDE). See the `density` help page for more information.

Usage

```
create_density_plot(
  df = NULL,
  aes_x = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  bw = "nrd0",
  adjust = 1,
  kernel = "gaussian",
  n = 512,
  position = "identity",
```

```
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  density_linewdth = 1,
  density_color = "black",
  density_fill = "gray",
  density_alpha = 0.4,
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  axis_text_size = 11,
  show_major_grids = TRUE,
  show_minor_grids = TRUE,
  plot_obs = FALSE,
  plot_obs_len = 0.02,
  plot_obs_color = "black",
  bold_x = NULL,
  bold_x_color = "black",
  bold_x_linetype = "dashed",
  panel_color = "white",
  panel_border_color = "black",
  show_legend = TRUE,
  legend_pos = "right",
  legend_key_width = 0.7,
  legend_key_height = 0.7,
  legend_key_backgrd = "white",
  silent_NA_warning = FALSE,
  cum_prob = NULL,
  area_colors = c("gray", "green", "gray"),
  area_quantile_line_color = "red",
  png_file_path = NULL,
  png_width_height = c(480, 480)
)
```

Arguments

df	A required data frame that contains a numeric column from which to estimate the KDE along with possible variables for setting 'aes_color' and 'aes_fill' (see below).
aes_x	A required string that sets the variable of 'df' from which to estimate the KDE.
aes_color	Sets the variable name from 'df' for the aesthetic mapping for color.
aes_fill	Sets the variable name from 'df' for the aesthetic mapping for fill.
bw	A string or numeric that sets the smoothing bandwidth to be used with the KDE function.
adjust	A numeric that adjusts 'bw' since the actual bandwidth is computed as adjust*bw.
kernel	A string that set the type of Kernel Density Estimation (KDE). Acceptable values are "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine".
n	The number of equally spaced points at which the density is to be estimated. This should be a power of two.
position	A string that sets the position. Acceptable values are "identity" which overlays or "stack" which stacks.
title	A string that sets the overall title.
subtitle	A string that sets the overall subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
density_linewdth	A numeric that sets the width attribute for the density line .
density_color	A string that sets the color attribute for the density line .
density_fill	A string that sets the fill color attribute for the area under the density line.
density_alpha	A numeric that sets the alpha attribute value for 'density_fill'.
x_limits	A numeric 2 element vector or function that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.
x_major_breaks	A numeric vector or function that sets the major tic locations along the x axis.

x_minor_breaks A numeric vector or function that sets the minor tic locations along the x axis.

x_labels A character vector or function giving x axis tic labels. Must be the same length as 'x_breaks'.

y_limits A numeric 2 element vector or function that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.

y_major_breaks A numeric vector or function that sets the major tic locations along the y axis.

y_minor_breaks A numeric vector or function that sets the minor tic locations along the y axis.

y_labels A character vector or function giving y axis tic labels. Must be the same length as 'y_breaks'.

axis_text_size A numeric that sets the font size along the axis'. Default is 11.

show_major_grids
A logical that controls the appearance of major grids.

show_minor_grids
A logical that controls the appearance of minor grids.

plot_obs A logical which if TRUE plots a line for each observation along the axis margin.

plot_obs_len A numeric that sets the length of the 'plot_obs' lines.

plot_obs_color A string that sets the color of the 'plot_obs' lines.

bold_x A numeric that sets the x-intercept for plotting a bold vertical line.

bold_x_color A string that sets the color of 'bold_x'.

bold_x_linetype
A string that set the linetype of 'bold_x'.

panel_color A string in hexidecimal or color name that sets the plot panel's color. The default is "white".

panel_border_color
A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

show_legend A logical that controls the appearance of the legend.

legend_pos A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

legend_key_width
A numeric that sets the legend width in cm.

legend_key_height
A numeric that sets the legend height in cm.

legend_key_backgrd
A string that sets the legend's background color.

silent_NA_warning
A logical that controls the appearance of a console warning when Na's are removed.

cum_prob A one or two element numeric vector for defining cumulative probabilities which divide the density plot area. The values are probabilities with values within 0.0 to 1.0. If for example with a cumulative probability 'cum_prob' of .95, then the density is divided into two probability areas of 95 then three areas are defined.

area_colors A string vector that sets the color of each area defined by 'cum_prob'.

area_quantile_line_color
A string that sets the vertical line color at quantile locations that divide the areas defined by cum_prob.

png_file_path A character string with the directory and file name to produce a png image of the plot.

png_width_height
A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(rlang)
library(data.table)
library(RplotterPkg)

RplotterPkg::create_density_plot(
  df = datasets::airquality,
  aes_x = "Ozone",
  rot_y_tic_label = TRUE,
  x_limits = c(-40,200),
  x_major_breaks = seq(-40,200,20),
  title = "Ozone Air Quality",
  x_title = "Ozone",
  y_title = "Density",
  plot_obs = TRUE,
  density_fill = "green",
  density_alpha = 0.5
)
```

---

create_density_ridge_plot

create_density_ridge_plot

Description

Function creates ggplot2 based plots stacked vertically (also known as ridge or raincloud plots). The function's density arguments mirror most of the arguments available from density for the Kernel Density Estimation (KDE). See the density help page for more information.

Usage

```
create_density_ridge_plot(
  df = NULL,
  variables = NULL,
  plot_heights = 3.5,
  plot_widths = 24,
  bw = "nrd0",
  adjust = 1,
  kernel = "gaussian",
  n = 512,
  na.rm = TRUE,
  title = NULL,
  title_fontsz = 14,
  x_title = NULL,
  rot_x_tic_angle = 0,
  density_linewdth = 1,
  density_color = "black",
  density_fill = "gray",
  density_alpha = 0.4,
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_show_axis = FALSE,
  axis_text_size = 11
)
```

Arguments

df	The source data frame from which the densities are plotted.
variables	A required string vector that names the x axis variables from 'df' for plotting their densities.
plot_heights	A numeric that sets the plot height in centimeters for each variable in 'variables'. The default is 3.5 centimeters in height for each plot.
plot_widths	A numeric that sets the plot width in centimeters for each variable in 'variables'. The default is 24 centimeters in width for each plot.
bw	A string or numeric that sets the smoothing bandwidth to be used with the KDE function.
adjust	A numeric that adjusts 'bw' since the actual bandwidth is computed as `adjust*bw`.
kernel	A string that set the type of Kernel Density Estimation (KDE). Acceptable values are "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine".
n	The number of equally spaced points at which the density is to be estimated. This should be a power of two.

na.rm	A logical which if TRUE, missing values are removed from 'df'. If FALSE any missing values cause an error.
title	A string that sets the overall title.
title_fontsz	A numeric that sets the title's font size. The default is 14.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
density_linewdth	A numeric that sets the density line width.
density_color	A string that sets the color for the density line.
density_fill	A string that sets the color for the density fill.
density_alpha	A numeric that sets the alpha value for 'density_fill'.
x_limits	A numeric 2 element vector or function that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.
x_major_breaks	A numeric vector or function that sets the major tic locations along the x axis.
x_minor_breaks	A numeric vector or function that sets the minor tic locations along the x axis.
x_labels	A character vector or function giving x axis tic labels. Must be the same length as 'x_major_breaks'.
y_limits	A numeric 2 element vector or function that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.
y_major_breaks	A numeric vector or function that sets the major tic locations along the y axis.
y_minor_breaks	A numeric vector or function that sets the minor tic locations along the y axis.
y_show_axis	A logical which if TRUE will display the y-axis density for each variable.
axis_text_size	A numeric that sets the font size along the axis'. Default is 11.

Value

A ggplot class object.

References

Micah Allen,Davide Poggiali,Kirstie Whitaker,Tom Rhys Marshall, Jordy van Langen,Rogier A. Kievit (2021) Raincloud plots: a multi-platform tool for robust data visualization: version 2

Joachim Schork (2021) Ridgeline Plots in R (3 Examples)

Examples

```
library(ggplot2)
library(data.table)
library(grid)
library(gtable)
library(ggplotify)
```

```
library(RplotterPkg)

RplotterPkg::create_density_ridge_plot(
  df = RplotterPkg::midwest,
  bw = "sj",
  variables = c("HS_Diploma", "College_Edu", "Prof_Deg", "White", "Black", "Asian"),
  title = "Percent Distribution Among Midwest Counties",
  x_limits = c(0, 100),
  x_major_breaks = seq(0, 100, 10),
  density_fill = "blue",
  density_alpha = 0.5,
  plot_heights = 2.5,
  plot_widths = 18
)
```

`create_heatmap` *create_heatmap*

Description

Function returns a ggplot2 x/y plot of a pair of discrete variables. Instead of a point, a rectangle/cell/tile is located for each x/y value with the cell itself being colored or sized depending on the value of a third associated discrete or continuous variable. Function wraps ggplot2 geom_tile to produce a heatmap that shows magnitude as an array of cells in two dimensions.

Usage

```
create_heatmap(
  df,
  aes_x = NULL,
  aes_y = NULL,
  aes_label = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  aes_size = NULL,
  aes_width = NULL,
  aes_height = NULL,
  tile_sz = 1.1,
  tile_color = "white",
  tile_fill = "white",
  label_sz = 6,
  label_color = "black",
  label_fontface = "plain",
  label_alpha = 1,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
```

```
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  x_major_date_breaks = waiver(),
  x_minor_date_breaks = waiver(),
  x_date_labels = waiver(),
  x_log10 = FALSE,
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  y_log10 = FALSE,
  x_y_decimals = NULL,
  x_y_scientific = NULL,
  axis_text_size = 11,
  panel_color = "white",
  panel_border_color = "black",
  show_legend = TRUE,
  legend_pos = "right",
  legend_key_width = 0.5,
  legend_key_height = 0.7,
  legend_key_backgrd = "white",
  silent_NA_warning = FALSE,
  png_file_path = NULL,
  png_width_height = c(480, 480)
)
```

Arguments

df	The target data frame from which the heatmap is plotted.
aes_x	The x axis variable name from 'df'. This is a required discrete numeric, Date/POSIXct variable.
aes_y	The y axis variable name from 'df'. This is a required discrete numeric.
aes_label	Sets the variable name from 'df' as the source for the tile's text.
aes_color	The discrete/continuous variable name from 'df' for the aesthetic mapping for color.
aes_fill	The discrete/continuous variable name from 'df' for the aesthetic mapping for fill.
aes_size	The variable name from 'df' for the aesthetic mapping for size.

aes_width	The variable name from 'df' for the aesthetic mapping for width.
aes_height	The variable name from 'df' for the aesthetic mapping for height.
tile_sz	A numeric that sets the tile's size width attribute in millimeters.
tile_color	A numeric that sets the tile's attribute border color.
tile_fill	A string that sets the fill color attribute for the tile.
label_sz	A numeric that sets the size of the label.
label_color	A string that sets the label's color.
label_fontface	A string that sets the label's font face. Acceptable values are "plain", "bold", "italic", "bold.italic".
label_alpha	A numeric that sets the label's alpha value.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
x_limits	Depending on the class of 'aes_x', a numeric/Date/POSIXct 2 element vector that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.
x_major_breaks	Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact major tic locations along the x axis.
x_minor_breaks	Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact minor tic locations along the x axis.
x_labels	A character vector with the same length as 'x_major_breaks', that labels the major tics.
x_major_date_breaks	If the class of 'aes_x' is Date/POSIXct, a string containing the number and date unit for major breaks. Examples: "1 year", "4 sec", "3 month", "2 week".
x_minor_date_breaks	If the class of 'aes_x' is Date/POSIXct, a string containing the number and date unit for minor breaks.

x_date_labels If the class of 'aes_x' is Date/POSIXct, a string containing the format codes, the strftime format, for the date. Examples: %Y-%m, %Y/%b/%d, %H-%M-%S

x_log10 A logical which if TRUE will use a log10 scale for the x axis.

y_limits A numeric 2 element vector that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.

y_major_breaks A numeric vector or function that defines the exact major tic locations along the y axis.

y_minor_breaks A numeric vector or function that defines the exact minor tic locations along the y axis.

y_labels A character vector with the same length as 'y_major_breaks', that labels the major tics.

y_log10 A logical which if TRUE will use a log10 scale for the y axis.

x_y_decimals A two element numeric vector that set the number of decimals for the x and y tic labels.

x_y_scientific A two element logical vector that if TRUE uses scientific notation for the x and y tic labels.

axis_text_size A numeric that sets the font size along the axis'. Default is 11.

panel_color A string in hexidecimal or color name that sets the plot panel's color. The default is "white".

panel_border_color
A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

show_legend A logical that controls the appearance of the legend.

legend_pos A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

legend_key_width
A numeric that sets the legend width in cm.

legend_key_height
A numeric that sets the legend height in cm.

legend_key_backgrd
A string that sets the legend's background color.

silent_NA_warning
A logical that controls the appearance of a console warning when Na's are removed.

png_file_path A character string with the directory and file name to produce a png image of the plot.

png_width_height
A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(data.table)
library(RplotterPkg)

RplotterPkg::create_heatmap(
  df = RplotterPkg::spinrates,
  aes_x = "velocity",
  aes_y = "spinrate",
  aes_fill = "swing_miss",
  aes_label = "swing_miss",
  label_fontface = "bold",
  title = "Likelihood of swinging and missing on a fastball",
  x_title = "Velocity",
  y_title = "Spinrate",
  label_sz = 5,
  rot_y_tic_label = TRUE
) +
 ggplot2::scale_fill_gradientn(
   colors = RColorBrewer::brewer.pal(n = 9, name = "YlOrRd"),
   n.breaks = 8
 ) +
 ggplot2::guides(
   fill = ggplot2::guide_colorbar(
     ticks.colour = "black"
   )
 )
```

---

create_histogram_plot *create_histogram_plot*

Description

Function plots a ggplot2 based histogram with options for scaling and viewing observation locations. The function offers one of four ways of setting the number of bins.

Usage

```
create_histogram_plot(
  df,
  aes_x = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  position = "stack",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
```

```
x_title = NULL,
y_title = NULL,
hide_x_tics = FALSE,
hide_y_tics = FALSE,
rot_x_tic_angle = 0,
rot_y_tic_label = FALSE,
bins = 20,
binwidth = NULL,
bin_breaks = NULL,
bin_class = NULL,
bar_fill = NA,
bar_color = "black",
bar_alpha = 1,
bar_lwd = 1,
x_limits = NULL,
x_major_breaks = waiver(),
x_minor_breaks = waiver(),
x_labels = waiver(),
x_decimals = NULL,
x_scientific = NULL,
x_log10 = FALSE,
y_limits = NULL,
y_major_breaks = waiver(),
y_minor_breaks = waiver(),
y_labels = waiver(),
axis_text_size = 11,
do_coord_flip = FALSE,
bar_labels = FALSE,
bar_label_size = 6,
bar_label_color = "black",
plot_obs = FALSE,
plot_obs_len = 0.02,
plot_obs_color = "black",
bold_x = NULL,
bold_x_color = "black",
bold_x_linetype = "dashed",
show_major_grids = TRUE,
show_minor_grids = TRUE,
show_legend = TRUE,
panel_color = "white",
panel_border_color = "black",
legend_pos = "top",
legend_key_width = 0.7,
legend_key_height = 0.7,
legend_key_backgrd = "white",
silent_NA_warning = FALSE,
png_file_path = NULL,
png_width_height = c(480, 480)
```

)

Arguments

df	The target data frame for the bar chart.
aes_x	A required string that sets the x axis continuous variable name from 'df'.
aes_color	Sets the variable name from 'df' for the aesthetic mapping for color.
aes_fill	Sets the variable name from 'df' for the aesthetic mapping for fill.
position	A string that sets the bar positions. Acceptable values are "dodge", "dodge2"(side by side), "identity"(overlap) or "stack".
title	A string that sets the overall title.
subtitle	A string that sets the overall subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
bins	An integer that sets the number of bins of the histogram. Is overridden by a non-null value for either 'binwidth', 'bin_breaks' or 'bins_class'. Default is 20.
binwidth	A numeric that sets the number of bins based on this value. If 'aes_x' is a date variable then 'binwidth' is the number of days and if a time variable then 'binwidth' is the number of seconds.
bin_breaks	A numeric vector that sets the number of bins by giving the bin boundaries explicitly.
bin_class	A character string that sets the number of bins by selecting one of three types of formulas. Acceptable values are "Sturges", "Scott", or "FD".
bar_fill	A string that sets the fill color attribute for the bars.
bar_color	A string that sets the outline color attribute for the bars.
bar_alpha	A numeric that set the alpha component attribute to 'bar_color'.
bar_lwd	A numeric that sets the bar's outline line width attribute.
x_limits	A numeric 2 element vector or function that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.

x_major_breaks	A numeric vector or function that sets the major tic locations along the x axis.
x_minor_breaks	A numeric vector or function that sets the minor tic locations along the x axis.
x_labels	A character vector or function giving x axis tic labels. Must be the same length as 'x_major_breaks'.
x_decimals	A numeric that sets the number of decimal places for x-tic labels.
x_scientific	A logical which if TRUE will put the x-tic labels in scientific notation.
x_log10	A logical which if TRUE will use a log10 scale for the x axis.
y_limits	A numeric 2 element vector or function that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.
y_major_breaks	A numeric vector or function that sets the major tic locations along the y axis.
y_minor_breaks	A numeric vector or function that sets the minor tic locations along the y axis.
y_labels	A character vector or function giving y axis tic labels. Must be the same length as 'y_major_breaks'.
axis_text_size	A numeric that sets the font size along the axis'. Default is 11.
do_coord_flip	A logical which if TRUE will flip the x and y axis'.
bar_labels	A logical which if TRUE will label each bar with its value.
bar_label_size	A numeric that sets the size of the bar labels
bar_label_color	A string that sets the color of the bar labels
plot_obs	A logical which if TRUE plots a line for each observation along the axis margin.
plot_obs_len	A numeric that sets the length of the 'plot_obs' lines.
plot_obs_color	A string that sets the color of the 'plot_obs' lines.
bold_x	A numeric that sets the x-intercept for plotting a bold vertical line.
bold_x_color	A string that sets the color of 'bold_x'.
bold_x_linetype	A string that set the linetype of 'bold_x'.
show_major_grids	A logical that controls the appearance of major grids.
show_minor_grids	A logical that controls the appearance of minor grids.
show_legend	A logical that controls the appearance of the legend.
panel_color	A string in hexidecimal or color name that sets the plot panel's color. The default is "white".
panel_border_color	A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".
legend_pos	A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".
legend_key_width	A numeric that sets the legend width in cm.

legend_key_height
: A numeric that sets the legend height in cm.

legend_key_backgrd
: A string that sets the legend's background color.

silent_NA_warning
: A logical that controls the appearance of a console warning when Na's are removed.

png_file_path
: A character string with the directory and file name to produce a png image of the plot.

png_width_height
: A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(data.table)
library(rlang)
library(RplotterPkg)

RplotterPkg::create_histogram_plot(
  df = RplotterPkg::midwest,
  aes_x = "Area",
  binwidth = 0.01,
  x_limits = c(0.0, 0.11),
  x_major_breaks = seq(0.0, 0.11, 0.01),
  title = "Distribution of area",
  subtitle = "437 counties from midwest dataset",
  x_title = "Area",
  y_title = "Count",
  bar_color = "white",
  bar_lwd = 2.0,
  bar_fill = "brown",
  do_coord_flip = TRUE,
  bar_labels = TRUE,
  bar_label_size = 4,
  bar_label_color = "blue",
  rot_y_tic_label = TRUE,
  silent_NA_warning = TRUE,
  plot_obs = TRUE,
  plot_obs_color = "darkorange"
)
```

create_range_plot *create_range_plot*

Description

Function wraps ggplot2::geom_pointrange to produce x/y plot of numeric ranges. Function returns a ggplot2 plot object displaying the individual spread or vertical interval/range for a collection of x/y pairs of points.

Usage

```
create_range_plot(
  df = NULL,
  orientation = "x",
  aes_x = NULL,
  aes_y = NULL,
  aes_min = NULL,
  aes_max = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  x_log10 = FALSE,
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  y_log10 = FALSE,
  axis_text_size = 11,
  pts_fill = "white",
  pts_shape = 21,
  pts_stroke = 1,
  pts_size = 2,
  line_type = "solid",
  line_width = 1,
  line_pts_color = "black",
  line_pts_alpha = 1,
```

```
  panel_color = "white",
  panel_border_color = "black",
  show_major_grids = TRUE,
  show_minor_grids = TRUE,
  bold_x = NULL,
  bold_x_color = "black",
  bold_x_linetype = "dashed",
  bold_y = NULL,
  bold_y_color = "black",
  bold_y_linetype = "dashed",
  silent_NA_warning = FALSE,
  png_file_path = NULL,
  png_width_height = c(480, 480)
)
```

Arguments

df	The target data frame from which the point ranges are plotted.
orientation	A string that sets the axis on which the range should run along. Acceptable values are either "x" (the default) or "y".
aes_x	Sets the x axis numeric variable name from 'df'.
aes_y	Sets a y axis variable name from 'df'.
aes_min	A string that sets a numeric variable from 'df' that defines the minimum values for the range.
aes_max	A string that sets a numeric variable from 'df' that defines the maximum values for the range.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
x_limits	A numeric 2 element vector that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.

x_major_breaks	A numeric vector or function that defines the exact major tic locations along the x axis.
x_minor_breaks	A numeric vector or function that defines the exact minor tic locations along the x axis.
x_labels	A character vector with the same length as 'x_major_breaks', that labels the major tics.
x_log10	A logical which if TRUE will use a log10 scale for the x axis.
y_limits	A numeric 2 element vector that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.
y_major_breaks	A numeric vector or function that defines the exact major tic locations along the y axis.
y_minor_breaks	A numeric vector or function that defines the exact minor tic locations along the y axis.
y_labels	A character vector with the same length as 'y_major_breaks', that labels the major tics.
y_log10	A logical which if TRUE will use a log10 scale for the y axis.
axis_text_size	A numeric that sets the font size along the axis'. Default is 11.
pts_fill	A string that sets the fill color of the points.
pts_shape	A numeric integer that sets the shape of the points. Typical values are 21 "circle", 22 "square", 23 "diamond", 24 "up triangle", 25 "down triangle".
pts_stroke	A numeric that sets the drawing width for a point shape.
pts_size	A numeric that sets the point size.
line_type	A string that sets range line type "twodash", "solid", "longdash", "dotted", "dotdash", "dashed", "blank".
line_width	A numeric that sets the width of the lines.
line_pts_color	A string that sets the color of the range lines and outlines of the points.
line_pts_alpha	A numeric value that sets the alpha level of points.
panel_color	A string in hexidecimal or color name that sets the plot panel's color. The default is "white".
panel_border_color	A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".
show_major_grids	A logical that controls the appearance of major grids.
show_minor_grids	A logical that controls the appearance of minor grids.
bold_x	A numeric that sets the x-intercept for plotting a bold vertical line.
bold_x_color	A string that sets the color of 'bold_x'.
bold_x_linetype	A string that set the linetype of 'bold_x'.
bold_y	A numeric that sets the y-intercept for plotting a bold horizontal line.

bold_y_color A string that sets the color of 'bold_y'.

bold_y_linetype
A string that set the linetype of 'bold_y'.

silent_NA_warning
A logical that controls the appearance of a console warning when Na's are removed.

png_file_path A character string with the directory and file name to produce a png image of the plot.

png_width_height
A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(RplotterPkg)

RplotterPkg::create_range_plot(
  df = RplotterPkg::penguins_stats,
  orientation = "x",
  aes_x = "avg_body_mass",
  aes_y = "species",,
  aes_min = "min_body_mass",
  aes_max = "max_body_mass",
  title = "Average and Range of Penguins Body Mass(g) by Species",
  subtitle = "Source: palmerpenguins",
  center_titles = TRUE,
  x_title = "Body Mass(g)",
  y_title = "Species",
  pts_fill = "blue",
  pts_shape = 22,
  pts_stroke = 1.7,
  line_width = 1.5,
  line_type = "solid",
  line_pts_color = "red",
  line_pts_alpha = 0.5,
  x_limits = c(2500, 7000),
  x_major_breaks = seq(2500,7000,500),
  show_major_grids = TRUE,
  show_minor_grids = FALSE
)
```

create_raster_plot *create_raster_plot*

Description

The function is wrapper around `ggplot2::geom_raster()` that plots from a data.frame with columns for x and y coordinates along with an associated numeric attribute.

The function accepts a data frame with columns for numeric x/y values and an attribute that will be mapped as a color or fill aesthetic. Function provides additional ggplot2 text labeling and axis scaling.

Usage

```
create_raster_plot(
  df,
  aes_x = NULL,
  aes_y = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  interpolate = FALSE,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  panel_color = "white",
  panel_border_color = "black",
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  scale_breaks = waiver(),
  scale_values = NULL,
  scale_limits = NULL,
  scale_labels = waiver(),
  scale_colors = heat.colors(8),
  scale_na_value = "gray50",
  own_scale = FALSE,
  show_legend = TRUE,
```

```
  legend_pos = "right",
  legend_key_width = 0.5,
  legend_key_height = 0.7,
  legend_key_backgrd = "white"
)
```

Arguments

df	The required data.frame derived from a raster file (such as *.tiff) to be plotted.
aes_x	The required name of aesthetic variable from 'df' for the x dimension.
aes_y	The required name of aesthetic variable from 'df' for the y dimension.
aes_color	The variable name from 'df' for the attribute dependent aesthetic mapping for color.
aes_fill	The variable name from 'df' for the attribute dependent aesthetic mapping for fill.
interpolate	A logical which if TRUE interpolate linearly.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
panel_color	A string in hexidecimal or color name that sets the plot panel's color. The default is "white".
panel_border_color	A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".
x_limits	Depending on the class of 'aes_x', a numeric/Date/POSIXct 2 element vector that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.
x_major_breaks	Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact major tic locations along the x axis.
x_minor_breaks	Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact minor tic locations along the x axis.
x_labels	A character vector with the same length as 'x_major_breaks', that labels the major tics.
y_limits	A numeric 2 element vector that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.

y_major_breaks	A numeric vector or function that defines the exact major tic locations along the y axis.
y_minor_breaks	A numeric vector or function that defines the exact minor tic locations along the y axis.
y_labels	A character vector with the same length as 'y_major_breaks', that labels the major tics.
scale_breaks	A string/numeric vector that defines the scale breaks.
scale_values	A string/numeric vector that defines the possible values.
scale_limits	A string/numeric vector that defines the scale limits.
scale_labels	An optional string vector that defines the scale labels. Vector must be the same length as scale_breaks.
scale_colors	Vector of colors to use for n-color gradient.
scale_na_value	A string that sets the color for missing values.
own_scale	A logical which if TRUE, then your own scaling may be appended to the plot without using the above scale_* parameters.
show_legend	A logical that controls the appearance of the legend.
legend_pos	A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".
legend_key_width	A numeric that sets the legend width in cm.
legend_key_height	A numeric that sets the legend height in cm.
legend_key_backgrd	A string that sets the legend's background color.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(RplotterPkg)

RplotterPkg::create_raster_plot(
  df = RplotterPkg::kentucky_elevation,
  title = "County Elevations in Southeast Kentucky",
  aes_x = "x",
  aes_y = "y",
  aes_fill = "elevation"
) +
ggplot2::geom_sf(
  data = RplotterPkg::kentucky_counties,
  aes(x = NULL, y = NULL),
  alpha = 0,
  linewidth = 1.5
```

```
) +
ggplot2::geom_sf_text(
  data = RplotterPkg::kentucky_counties,
  aes(x = NULL, y = NULL, label = ID)
)
```

`create_scatter_plot` *create_scatter_plot*

Description

Function is a wrapper around ggplot2 geom_point and geom_line to produce a scatter plot. Function returns a ggplot2 plot object of x/y scatter points/lines. Options are provided for axis scaling and variable/non-variable dependent aesthetics.

Usage

```
create_scatter_plot(
  df = NULL,
  aes_x = NULL,
  aes_y = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  aes_size = NULL,
  aes_alpha = NULL,
  aes_linetype = NULL,
  aes_label = NULL,
  aes_label_color = "black",
  aes_label_size = 6,
  aes_label_nudge_x = 0,
  aes_label_nudge_y = -0.3,
  aes_CI_lwr = NULL,
  aes_CI_upr = NULL,
  position = position_jitter(width = 0, height = 0),
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  x_limits = NULL,
  x_major_breaks = waiver(),
```

```
x_minor_breaks = waiver(),
x_labels = waiver(),
x_major_date_breaks = waiver(),
x_minor_date_breaks = waiver(),
x_date_labels = waiver(),
x_log10 = FALSE,
y_limits = NULL,
y_major_breaks = waiver(),
y_minor_breaks = waiver(),
y_labels = waiver(),
y_log10 = FALSE,
x_y_decimals = NULL,
x_y_scientific = NULL,
axis_text_size = 11,
show_pts = TRUE,
pts_fill = "white",
pts_shape = 21,
pts_stroke = 1,
pts_color = "black",
pts_size = 1,
pts_line_alpha = 1,
connect = FALSE,
line_width = 0.6,
line_color = "black",
connect_linetype = "solid",
show_major_grids = TRUE,
show_minor_grids = TRUE,
show_legend = TRUE,
legend_pos = "right",
legend_key_width = 0.7,
legend_key_height = 0.7,
legend_key_backgrd = "white",
panel_color = "white",
panel_border_color = "black",
bold_y = NULL,
bold_y_color = "black",
bold_y_linetype = "dashed",
CI_dir = "y",
CI_show_line = FALSE,
CI_show_errorbar = FALSE,
CI_show_ribbon = FALSE,
CI_line_color = "black",
CI_line_width = 1,
CI_linetype = "dashed",
CI_errorbar_color = "black",
CI_errorbar_width = 1,
CI_ribbon_color = "gray70",
silent_NA_warning = FALSE,
```

```
  png_file_path = NULL,
  png_width_height = c(480, 480)
)
```

Arguments

df	The target data frame from which the scatter points are plotted.
aes_x	The x axis variable name from 'df'. This is a required discrete factor, continuous numeric, or Date/POSIXct variable.
aes_y	The y axis variable name from 'df'. This is a required discrete factor or continuous numeric.
aes_color	The variable name from df' for the variable dependent aesthetic mapping for color.
aes_fill	The variable name from 'df' for the variable dependent aesthetic mapping for fill.
aes_size	The variable name from 'df' for the variable dependent aesthetic mapping for size.
aes_alpha	The variable name from 'df' for the variable dependent aesthetic mapping for alpha.
aes_linetype	The variable name from 'df' for the variable dependent aesthetic mapping for linetype.
aes_label	The variable name from 'df' for the variable dependent aesthetic mapping for labeling.
aes_label_color	A string that sets the color of labels.
aes_label_size	A numeric that sets the size of labels.
aes_label_nudge_x	A numeric that nudges the label's horizontal position.
aes_label_nudge_y	A numeric that nudges the label's vertical position.
aes_CI_lwr	Sets the column from 'df' for the lower confidence interval with reference to the x or y axis.
aes_CI_upr	Sets the column from 'df' for the upper confidence interval with reference to the x or y axis.
position	A string or function that does a slight adjustment to overlapping points. Typical values are "jitter" or `position_jitter(width = 0.1, height = 0.1)`.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if `TRUE` centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If `NULL` (the default) then the x axis title does not appear.

y_title A string that sets the y axis title. IfNULL (the default) then the y axis title does not appear.

hide_x_tics A logical that controls the appearance of the x axis tics.

hide_y_tics A logical that controls the appearance of the y axis tics.

rot_x_tic_angle
A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.

rot_y_tic_label
A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.

x_limits Depending on the class of 'aes_x', a numeric/Date/POSIXct 2 element vector that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.

x_major_breaks Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact major tic locations along the x axis.

x_minor_breaks Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact minor tic locations along the x axis.

x_labels A character vector with the same length as 'x_major_breaks', that labels the major tics.

x_major_date_breaks
If the class of 'aes_x' is Date/POSIXct, a string containing the number and date unit for major breaks. Examples: "1 year", "4 sec", "3 month", "2 week".

x_minor_date_breaks
If the class of 'aes_x' is Date/POSIXct, a string containing the number and date unit for minor breaks.

x_date_labels If the class of 'aes_x' is Date/POSIXct, a string containing the format codes, the strftime format, for the date. Examples: %Y-%m, %Y/%b/%d, %H-%M-%S

x_log10 A logical which if TRUE will use a log10 scale for the x axis.

y_limits A numeric 2 element vector that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.

y_major_breaks A numeric vector or function that defines the exact major tic locations along the y axis.

y_minor_breaks A numeric vector or function that defines the exact minor tic locations along the y axis.

y_labels A character vector with the same length as 'y_major_breaks', that labels the major tics.

y_log10 A logical which if TRUE will use a log10 scale for the y axis.

x_y_decimals A two element numeric vector that set the number of decimals for the x and y tic labels.

x_y_scientific A two element logical vector that if TRUE uses scientific notation for the x and y tic labels.

axis_text_size A numeric that sets the font size along the axis'. Default is 11.

show_pts A logical which if FALSE will plot only the lines if 'connect' is TRUE.

`pts_fill` A string that sets the fill color attribute of the points.

`pts_shape` A numeric integer that sets the shape attribute of the points. Typical values are 21 "circle", 22 "square", 23 "diamond", 24 "up triangle", 25 "down triangle".

`pts_stroke` A numeric that sets the drawing stroke width attribute for a point shape.

`pts_color` A string that sets the color attribute of the points.

`pts_size` A numeric value that sets the size attribute of the points.

`pts_line_alpha` A numeric value that sets the alpha level attribute of points and connected lines.

`connect` A logical which if `TRUE` then points will be connected with a line.

`line_width` A numeric value that sets the width of lines if 'connect' is TRUE.

`line_color` A string that sets the color of the lines if 'connect' is TRUE.

`connect_linetype` A string that sets line type "twodash", "solid", "longdash", "dotted", "dotdash", "dashed", "blank" if 'connect' is `TRUE`.

`show_major_grids` A logical that controls the appearance of major grids.

`show_minor_grids` A logical that controls the appearance of minor grids.

`show_legend` A logical that controls the appearance of the legend.

`legend_pos` A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

`legend_key_width` A numeric that sets the legend width in cm.

`legend_key_height` A numeric that sets the legend height in cm.

`legend_key_backgrd` A string that sets the legend's background color.

`panel_color` A string in hexidecimal or color name that sets the plot panel's color. The default is "white".

`panel_border_color` A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

`bold_y` A numeric that sets the y-intercept for plotting a bold horizontal line.

`bold_y_color` A string that sets the color of 'bold_y'.

`bold_y_linetype` A string that set the linetype of 'bold_y'.

`CI_dir` A string that sets the axis orientation of the confidence intervals. Acceptable values are "x" or "y".

`CI_show_line` A logical that if `TRUE` shows lower/upper confidence lines.

`CI_show_errorbar` A logical that if `TRUE` shows error bars between 'aes_y' and lower/upper confidence lines.

CI_show_ribbon A logical that if TRUE shows a filled ribbon between the lower/upper confidence lines.

CI_line_color A string that sets the lower/upper confidence line colors.

CI_line_width A numeric that sets the lower/upper confidence line widths.

CI_linetype A string that sets the linetype for lower/upper confidence lines.

CI_errorbar_color
A string that sets the error bars' color.

CI_errorbar_width
A numeric that sets the error bars' width.

CI_ribbon_color
A string that sets the fill color for the confidence interval ribbon.

silent_NA_warning
A logical that controls the appearance of a console warning when Na's are removed.

png_file_path A character string with the directory and file name to produce a png image of the plot.

png_width_height
A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(RplotterPkg)

RplotterPkg::create_scatter_plot(
  df = ggplot2::economics,
  aes_x = "date",
  aes_y = "unemploy",
  pts_shape = 21,
  pts_fill = "black",
  line_color = "violet",
  connect = TRUE,
  title = "US Monthly Unemployment",
  subtitle = "July, 1967 to April, 2015 (in thousands)",
  x_title = "Date",
  y_title = "Unemployment",
  rot_y_tic_label = TRUE,
  x_date_labels = "%Y-%b",
  x_major_date_breaks = "5 year",
  y_limits = c(0, 16000),
  y_major_breaks = seq(0, 16000, 2000),
  show_minor_grids = FALSE,
  bold_y = 8000,
  bold_y_color = "red",
```

```
  bold_y_linetype = "dashed"
)
```

`create_sf_plot` *create_sf_plot*

Description

The function is a wrapper around `ggplot2::geom_sf()` that plots simple feature objects. See Simple Features for R for more information on simple features.

For a good introduction to using ggplot2::geom_sf see Drawing beautiful maps programmatically with R, sf and ggplot2.

The function accepts objects of class sf for visualizing their points, lines, and polygon geometries. Easy plotting is possible with the function's parameters for scaling, color, size, variable aesthetic mapping, and text labeling. Overlapping of geometries is possible by chaining multiple functions via standard piping.

Usage

```
create_sf_plot(
  sf,
  gg = NULL,
  aes_color = NULL,
  aes_fill = NULL,
  aes_size = NULL,
  aes_text = NULL,
  text_size = 3,
  text_color = "black",
  text_fontface = "plain",
  text_check_overlap = FALSE,
  text_nudge_x = 0,
  text_nudge_y = 0,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  xlim = NULL,
  ylim = NULL,
  grid_line_color = NULL,
  grid_line_size = 1,
  panel_color = NULL,
  panel_border_color = "black",
```

```
  panel_expand = FALSE,
  sf_color = "black",
  sf_fill = "gray",
  sf_stroke = 0.1,
  sf_shape = 21,
  sf_size = 0.1,
  sf_linewidth = 0.1,
  sf_alpha = 1,
  inherit_aes = TRUE,
  na_rm = FALSE,
  scale_breaks = waiver(),
  scale_values = NULL,
  scale_limits = NULL,
  scale_labels = waiver(),
  scale_colors = heat.colors(8),
  scale_na_value = "gray50",
  own_scale = FALSE,
  show_legend = TRUE,
  legend_pos = "right",
  legend_key_width = 0.5,
  legend_key_height = 0.7,
  legend_key_backgrd = "white"
)
```

Arguments

sf	A simple features object of class "sf".
gg	A base ggplot2 object wherein this ggplot2's geom_sf object is layered on.
aes_color	The variable name from 'sf' for the dependent aesthetic mapping for color.
aes_fill	The variable name from 'sf' for the dependent aesthetic mapping for fill. If the variable is a discrete factor, see ggplot2::scale_fill_manual() for appropriate scaling values. If the variable is continuous, see ggplot2::scale_fill_gradientn().
aes_size	The variable name from 'sf' for the dependent aesthetic mapping for point size.
aes_text	The variable name from 'sf' for the dependent aesthetic mapping for text labeling.
text_size	A numeric value that sets the size of aesthetic mapping of text (i.e. aes_text)
text_color	A string that sets the color of aesthetic mapping of text color (i.e. aes_text)
text_fontface	A string that sets the fontface of aesthetic mapping of text fontface (i.e. aes_text). Acceptable values: "plain", "bold", "italic", "bold.italic". The default is "plain".
text_check_overlap	A logical which if TRUE will not plot text that overlaps.
text_nudge_x	A numeric that sets the value for nudging the text in the x direction.
text_nudge_y	A numeric that sets the value for nudging the text in the y direction.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.

caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
xlim	A numeric vector pair of longitudinal values for zooming in/out the mapping
ylim	A numeric vector pair of latitudinal values for zooming in/out the mapping
grid_line_color	A string in hexidecimal or color name that sets the plot major grid line color. The default is NULL and takes on ggplot2's default white.
grid_line_size	A numeric that sets the grid line's width. The default is 1.
panel_color	A string in hexidecimal or color name that sets the plot panel's color. The default is NULL and takes on ggplot2's default gray..
panel_border_color	A string in hexidecimal or color name that sets the plot panel's border color. The default is "black". Set it to NA to eliminate the border rectangle entirely.
panel_expand	A logical which if TRUE, expands the plot panel and potentially hides the tics. The default is FALSE.
sf_color	A string that sets the color attribute of the sf.
sf_fill	A string that sets the fill color attribute of the sf.
sf_stroke	A numeric that sets the drawing stroke width attribute for a sf point geometry.
sf_shape	A numeric that sets the non-variable associated shape aesthetic.
sf_size	A numeric value that sets the size attribute for scaling points.
sf_linewidth	A numeric value that sets the line width of POLYGON, LINESTRING geometries.
sf_alpha	A numeric value that sets the alpha level attribute of point and line geometries..
inherit_aes	A logical which if FALSE the aesthetics are not combined with other overlapping geoms.
na_rm	A logical which if TRUE, missing observations are removed. If FALSE, the default, missing observations are removed with a warning.
scale_breaks	A string/numeric vector that defines the scale breaks.
scale_values	A string/numeric vector that defines the possible values.
scale_limits	A string/numeric vector that defines the scale limits.
scale_labels	An optional string vector that defines the scale labels. Vector must be the same length as 'scale_breaks'.
scale_colors	Vector of colors to use for n-color gradient.
scale_na_value	A string that sets the color for missing values.

own_scale A logical which if TRUE, then your own scaling may be appended to the plot without using the above scale_* parameters.

show_legend A logical that controls the appearance of the legend.

legend_pos A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

legend_key_width A numeric that sets the legend width in cm.

legend_key_height A numeric that sets the legend height in cm.

legend_key_backgrd A string that sets the legend's background color.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(RColorBrewer)
library(RplotterPkg)

# map country coffee production over country geometries
RplotterPkg::create_sf_plot(
  sf = spData::world,
  title = "World Coffee Production 2017",
  subtitle = "Source: spData::coffee_data",
  panel_color = "white",
  panel_border_color = "white",
) |>
  RplotterPkg::create_sf_plot(
    sf = RplotterPkg::world_coffee,
    aes_fill = "coffee_production_2017",
    scale_breaks = seq(0,3000,250),
    scale_labels = seq(0,3000,250),
    scale_limits = c(0,3000),
    scale_colors = RColorBrewer::brewer.pal(n = 9, name = "YlOrRd"),
    legend_key_width = 0.7,
    legend_key_height = 1.0
  )
```

---

create_stick_plot *create_stick_plot*

Description

Function wraps a ggplot2 geom_linerange to produce a "stick" plot. Function returns a plot object showing vertical/horizontal lines that run from a base value to a measurement value. Options are provided for scaling.

Usage

```
create_stick_plot(
  df = NULL,
  base_val = 0,
  aes_x = NULL,
  aes_y = NULL,
  aes_color = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = NULL,
  y_title = NULL,
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  x_major_date_breaks = waiver(),
  x_date_labels = waiver(),
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  axis_text_size = 11,
  line_color = "black",
  line_width = 0.8,
  line_type = "solid",
  line_alpha = 1,
  show_major_grids = TRUE,
  show_minor_grids = TRUE,
  panel_color = "white",
  panel_border_color = "black",
  show_legend = TRUE,
  legend_pos = "right",
  legend_key_width = 1.5,
  legend_key_height = 1.5,
  legend_key_backgrd = "white",
  bold_y = NULL,
  bold_y_color = "black",
  silent_NA_warning = FALSE,
  png_file_path = NULL,
  png_width_height = c(480, 480)
)
```

Arguments

df	The required target data frame from which the "stick" lines are drawn.
base_val	A numeric that sets the base value from which the "stick" originates. The default value is 0.
aes_x	The required x axis variable name from 'df'. Can be a continuous numeric/Date/POSIXct or discrete factor variable.
aes_y	The required y axis numeric variable name from 'df' and controls the height of individual "sticks".
aes_color	Sets the variable name from 'df' for the aesthetic mapping for color.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
x_limits	Depending on the class of 'aes_x', a numeric/Date/POSIXct 2 element vector that sets the minimum and maximum for the x axis. Use NA to refer to the existing minimum and maximum.
x_major_breaks	Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact major tic locations along the x axis.
x_minor_breaks	Depending on the class of 'aes_x', a numeric/Date/POSIXct vector or function that defines the exact minor tic locations along the x axis.
x_labels	A character vector with the same length as 'x_major_breaks', that labels the major tics.
x_major_date_breaks	If the class of 'aes_x' is Date/POSIXct, a string containing the number and date unit for major breaks. Examples: "1 year", "4 sec", "3 month", "2 week".
x_date_labels	If the class of 'aes_x' is Date/POSIXct, a string containing the format codes, the strftime format, for the date. Examples: %Y-%m, %Y/%b/%d, %H-%M-%S
y_limits	A numeric 2 element vector that sets the minimum and maximum for the y axis. Use NA to refer to the existing minimum and maximum.

`y_major_breaks` A numeric vector or function that defines the exact major tic locations along the y axis.

`y_minor_breaks` A numeric vector or function that defines the exact minor tic locations along the y axis.

`y_labels` A character vector with the same length as 'y_major_breaks', that labels the major tics.

`axis_text_size` A numeric that sets the font size along the axis'. Default is 11.

`line_color` A string that sets the color attribute of the lines.

`line_width` A numeric value that sets the width of lines.

`line_type` A string that sets the linetype. The default is "solid".

`line_alpha` A numeric value that sets the degree of color alpha attribute for the lines.

`show_major_grids` A logical that controls the appearance of major grids.

`show_minor_grids` A logical that controls the appearance of minor grids.

`panel_color` A string in hexidecimal or color name that sets the plot panel's color. The default is "white".

`panel_border_color` A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

`show_legend` A logical that controls the appearance of the legend.

`legend_pos` A string that sets the legend position. Acceptable values are "top", "bottom", "left", "right".

`legend_key_width` A numeric that sets the legend width in cm.

`legend_key_height` A numeric that sets the legend height in cm.

`legend_key_backgrd` A string that sets the legend's background color.

`bold_y` A numeric that plots a bold horizontal line at this y value.

`bold_y_color` A string that sets the bold horizontal line color. Default is "black".

`silent_NA_warning` A logical that controls the appearance of a console warning when NA's are removed.

`png_file_path` A character string with the directory and file name to produce a png image of the plot.

`png_width_height` A numeric vector that sets the width and height of the png image in pixels. The default is c(480,480). There are 37.8 pixels in a centimeter.

Value

A ggplot class object.

Examples

```
library(ggplot2)
library(RplotterPkg)

RplotterPkg::create_stick_plot(
  df = RplotterPkg::air_passengers,
  aes_x = "time",
  aes_y = "value",
  title = "Monthly Totals of International Airline Passengers",
  subtitle = "1949 - 1960 (classic Box & Jenkins)",
  x_title = "Time",
  y_title = "Totals",
  x_major_date_breaks = "1 year",
  x_date_labels = "%Y",
  rot_y_tic_label = TRUE,
  show_minor_grids = FALSE,
  bold_y = 0.0
)
```

`create_table` *create_table*

Description

Function wraps the gt package for creating tables from a data.frame object. Function incorporates the gt package to create tables primarily for R Markdown/Quarto documents. The function offers a quick alternative if just limited styling and formatting are required.

Usage

```
create_table(
  df = NULL,
  title = NULL,
  subtitle = NULL,
  container_width_px = NULL,
  container_height_px = NULL,
  rowname_col = NULL,
  col_label_lst = NULL,
  col_width_lst = NULL,
  header_line = FALSE,
  caption = NULL,
  source_note = NULL,
  hor_scroll_bar = FALSE,
  ver_scroll_bar = FALSE,
  decimals_lst = NULL,
  footnote_title = NULL,
  footnote_col_head_lst = NULL,
```

```
  footnote_body_lst = NULL
)
```

Arguments

df	A data.frame from which to display a table.
title	A string that sets the table's title. The string can contain R markdown/html style syntax.
subtitle	A string that sets the table's subtitle. The string can contain R markdown/html style syntax.
container_width_px	A numeric that sets the overall container width in pixels.
container_height_px	A numeric that sets the overall container height in pixels.
rowname_col	The column name from 'df' to use as row captions to be placed in the display table stub.
col_label_lst	A list that resets the table's column labels. The list's key is the old label and the value is the new label. Initially the old labels will be 'df's column names.
col_width_lst	A list that sets the table's column widths. The list's key is a column's index number and value is the column width in pixels. Both the key and value are strings.
header_line	A logical which if TRUE will place a bold, thick line between the header row and table body.
caption	A string that sets the table caption which appears above the title to use as cross-referencing in R Markdown/Quarto document. The string can contain R markdown/html style syntax.
source_note	A string located at the bottom of the table that notes the table's source. The string can contain R markdown/html style syntax.
hor_scroll_bar	A logical that controls the appearance of the horizontal scroll bar.
ver_scroll_bar	A logical that controls the appearance of the vertical scroll bar.
decimals_lst	A list that describes the decimals for one or more table columns. The list should have the following key-value pairs: 1. cols = (an integer vector of column indices from 'df') 2. decimal = (an integer that defines the number of decimals for the column(s))
footnote_title	A string at the bottom of the table that footnotes the table's title.
footnote_col_head_lst	A list that describes the text/location of a footnote for column headings. The list should have the following key-value pairs: 1. footnote = (the footnote text to be displayed at the bottom of the table) 2. col = (an integer vector setting the footnote's column heading index location's)
footnote_body_lst	A list that describes the text/location of a footnote within the body of the table. The list should have the following key-value pairs:

1. footnote = (the footnote text to be displayed at the bottom of the table)
2. col = (an integer vector setting the footnote's column index location's)
3. row = (an integer vector setting the footnote's row index location's)

Value

A gt object of class "gt_tbl" which can be further modified.

Examples

```
library(gt)
library(glue)
library(purrr)
library(RplotterPkg)

RplotterPkg::create_table(
  df = RplotterPkg::people,
  container_width_px = 500,
  col_label_lst = list(
    col_0 = "ID",
    col_1 = "First Name",
    col_2 = "Last Name",
    col_3 = "Age",
    col_4 = "Address",
    col_5 = "Spouse"
  ),
  decimals_lst = list(
    cols = 4,
    decimal = 0
  ),
  rowname_col = "ID",
  source_note = "Source: *Chardon City Directory*",
  footnote_title = "Employees are current",
  footnote_col_head_lst = list(
    footnote = "As reported on their job application",
    col = c(2,3)
  ),
  footnote_body_lst = list(
    footnote = "Age is approximated",
    col = 4,
    row = c(1,2)
  )
)
```

---

create_table_graphic *create_table_graphic*

Description

Function creates a ggplot2 graphic table from a data frame. Function creates a simple, non-scrollable static table to be laid out with other ggplot2 graphics. The function is based on `grid` along with the `gtable` package.

For estimating the overall height of the table consider that each row of the table is 0.2 inches high, .54 inches for heading and if a title is defined then add an additional 0.5 inches.

Usage

```
create_table_graphic(
  df = NULL,
  table_width = 8,
  head_color = "black",
  head_fill = "white",
  head_font_sz = 14,
  cell_color = "black",
  cell_fill = "white",
  cell_font_sz = 11,
  cell_just = "center",
  cell_hor_pos = 0.5,
  title = NULL,
  title_font_sz = 16,
  show_row_names = TRUE,
  row_names_width = 0.5,
  return_gtable = FALSE
)
```

Arguments

`df`	The target data frame from which to graph.
`table_width`	The overall table width in inches.
`head_color`	Color of the headings foreground.
`head_fill`	Color of the headings background.
`head_font_sz`	Font size of the headings.
`cell_color`	Color of the cells foreground.
`cell_fill`	Color of the cells background.
`cell_font_sz`	Font size of the cells.
`cell_just`	An string that sets justification of the cell content. Acceptable values are "left" and "right" and "center".
`cell_hor_pos`	A numeric that sets the starting horizontal location of text in a cell. Acceptable range from 0.0 to 1.0.
`title`	A string that sets the table title.
`title_font_sz`	Font size of the title.
`show_row_names`	A logical that controls the appearance of row names in the data frame.

`row_names_width`
: The width of the column for row names in inches.

`return_gtable`
: A logical which if `TRUE` returns a gtable/grob of the table. The default is to return a ggplot class object.

Value

A gtable or ggplot class object.

Examples

```
library(grid)
library(gtable)
library(ggplotify)
library(RplotterPkg)

RplotterPkg::create_table_graphic(
  df = RplotterPkg::car_stats,
  table_width = 7,
  show_row_names = TRUE,
  cell_just = "right",
  cell_hor_pos = 0.95,
  title = "Average Measures by Cylinders"
)
```

`create_waffle_chart` *create_waffle_chart*

Description

Function creates a "waffle" chart that displays percentages for multiple variables. The chart is similar to a heatmap with a 10 by 10 array of tiles (each representing one percent) that are colored to the extent of the variable's proportion value. For more information on "waffle" charts see What is a waffle chart.

Usage

```
create_waffle_chart(
  x = NULL,
  name_col = NULL,
  prop_col = NULL,
  title = NULL,
  select_fill = "brown",
  default_fill = "lightgray"
)
```

Arguments

x	Either a dataframe or a vector of named integer proportions.
name_col	If 'x' is a dataframe, a required string setting the column name having the variable names.
prop_col	If 'x' is a dataframe, a required string setting the column name having the variable proportions.
title	A string that sets the main title for the chart.
select_fill	A string that sets the fill color for tiles with proportion values.
default_fill	A string that sets the fill color for tiles without proportion values.

Value

A ggplot2 class object

Examples

```
library(grid)
library(gtable)
library(ggplotify)
library(ggplot2)
library(RplotterPkg)

proportions_v <- c(
var1=10,var2=40,var3=20, var4=50, var5=5,
var6=30, var7=10, var8=67, var9=42, var10=33,
var11=7, var12=35,var13=22, var14=90
)
create_waffle_chart(
  x = proportions_v,
  title = "Test For 14 Proportion Variables"
)
```

---

farms *farms data*

Description

Number of farms in the states of the U.S. The raw tab delimited data (`farms.txt`) is read by `data.table::fread()` to produce a data frame with columns for "state" and "count".

state	character as a state abbreviation
count	numeric of state's total farm count

Usage

```
farms
```

Format

An object of class `data.table` (inherits from `data.frame`) with 50 rows and 2 columns.

Source

LearnEDAfunctions

---

gapminder_data *gapminder_data*

Description

Based on `gapminder::gapminder` dataset, selects 10 year intervals and 4 continents. The "year" variable is defined as a factor and the "pop" variable is divided by 1e+6.

year	numeric factor as 1952, 1972, 1992, 2002
pop	population in units of 1e+6
continent	factor with 4 levels
lifeExp	life expectancy at birth, in years

Usage

```
gapminder_data
```

Format

An object of class `data.table` (inherits from `data.frame`) with 560 rows and 6 columns.

Source

`gapminder::gapminder`

---

get_grob_component *get_grob_component*

Description

Function retrieves a grob component from a ggplot2 plot object.

Usage

```
get_grob_component(a_plot = NULL, component_name = NULL)
```

Arguments

a_plot	A required ggplot2 plot object
component_name	A required string that sets the components name

Value

A list of grob objects

Examples

```
library(ggplot2)
library(ggplotify)
library(grid)
library(RplotterPkg)

mtcars_plot <- ggplot2::ggplot(
  data = datasets::mtcars,
) +
ggplot2::geom_point(aes(x = mpg, y = wt, color = cyl), size = 3)

legend_right <- RplotterPkg::get_grob_component(
  a_plot = mtcars_plot,
  component_name = "guide-box-right"
)
```

---

homeruns_2000 *homeruns_2000 data*

Description

Team home run numbers for different seasons. The raw tab delimited data (homeruns_2000.txt) is read by data.table::fread() to produce a data frame with numeric columns for "YEARS" and "HOMERUNS".

YEARS	numeric years for baseball teams from 1900 to 2000
HOMERUNS	numeric of total homeruns for a baseball team

Usage

```
homeruns_2000
```

Format

An object of class data.table (inherits from data.frame) with 210 rows and 2 columns.

Source

LearnEDAfunctions

kentucky_counties *kentucky_counties data*

Description

A simple feature object with the boundary geometries for nine southeast Kentucky counties. .

ID	character giving the name of the county
geom	MULTIPOLYGON of the Kentucky county boundary

Usage

```
kentucky_counties
```

Format

An object of class `sf` (inherits from `data.table`, `data.frame`) with 9 rows and 2 columns.

Source

`maps::map()`

kentucky_elevation *kentucky_elevation data*

Description

Tiled raster elevation data(in meters) converted to a data frame, covering the southeast region of Kentucky.

x	numeric longitude of the elevation
y	numeric latitude of the elevation
elevation	the elevation in meters

Usage

```
kentucky_elevation
```

Format

An object of class `data.frame` with 609 rows and 3 columns.

Source

`elevatr::get_elev_raster()` and the Amazon Web Services Terrain Tiles service.

midwest *midwest data*

Description

From the `ggplot2::midwest` dataset, selects for midwest counties percent high school diploma, percent college educated, percentage professional degree, percent white, percent black, percent asian.

HS_Diploma	numeric percent high school diploma
College_Edu	numeric percent college educated
Prof_Deg	percentage professional degree
White	percent white
Black	percent black
Asian	percent asian

Usage

```
midwest
```

Format

An object of class `data.table` (inherits from `data.frame`) with 437 rows and 7 columns.

Source

`ggplot2::midwest`

multi_panel_grid *multi_panel_grid*

Description

Arranges a group of ggplot2 plots or grid::grob's into multiple panels.

Function creates a figure with title and arranges multiple plots/grobs across a given number of rows and columns. The function depends on the `grid`, `gtable`, and `ggplotify` packages.

Usage

```
multi_panel_grid(
  layout = NULL,
  title = NULL,
  title_fontsz = 20,
  plot_titles = NULL,
  y_tick_width = 0.5,
```

```
  cell_width = 8,
  cell_height = 8,
  do_grid = FALSE
)
```

Arguments

layout	A list containing a list for plot objects or grob "plots", row locations "rows" vector, column locations "cols" vector. This argument is required.
title	A string that sets the title of the figure.
title_fontsz	A numeric that sets the title's font size. The default is 20.
plot_titles	A character vector with the same length as the number of plot objects that defines each of their titles. The parameter is used only if 'do_grid' is FALSE.
y_tick_width	A numeric that sets the width of the vertical column containing the y axis tick labeling. The default is 0.5 cm and may be increased when a wider labeling is needed.
cell_width	A numeric that sets the cell widths in the gtable in cm
cell_height	A numeric that sets the cell widths in the gtable in cm
do_grid	A logical which if TRUE will arrange the plot objects according to 'layout' without considering their scale and label components. You define the axis scaling, titling, etc for each individual plot object.

Details

The function has two modes of layout: a grid and a non-grid. In the grid layout, each plot submitted will be laid out "as-is" with no manipulation of labels, legends, or axis'.

For the non-grid layout, the function goes beyond just placing a group of plots across a grid of rows and columns. It is somewhat opinionated in favor of common titling/labeling and scaling that make sense for all the plots and provide the ability to compare. The function has a 'title' parameter that is assumed appropriate for all the plots–any titles/subtitles among the plots themselves will be removed. The first left y-axis scaling the function comes across will be used for all the plots. Similarly the first bottom x-axis scaling the function comes across will be assumed appropriate for all the plots. The first legend that is found is assumed right for all the plots and will be placed at the top in a horizontal direction. Of course none of these sub-elements among the plots need to exist and you are back to simply a grid of plots.

As an example of defining the 'layout' list argument, 'layout' has three named elements: "plots", "rows", and "cols". "plots" is a list of ggplot2 plot objects; "rows" defines the row number for each plot and "cols" defines the column number for each plot. If we have two plots with both plots on a single row then "rows" = c(1,1) and "cols" = c(1,2).

Value

A ggplot2 class object

Examples

```
library(ggplot2)
library(ggplotify)
library(rlang)
library(grid)
library(gtable)
library(ggplotify)
library(purrr)
library(RplotterPkg)

years <- levels(RplotterPkg::gapminder_data$year)
build_plot <- function(id, dt, years){
  plot_dt <- dt[year == years[[id]], ]
  aplot <- RplotterPkg::create_scatter_plot(
    df = plot_dt,
    aes_x = "lifeExp",
    aes_y = "pop",
    aes_fill = "continent",
    x_limits = c(20, 80),
    x_major_breaks = seq(from = 20, to = 80, by = 10),
    y_limits = c(0, 400),
    y_major_breaks = seq(from = 0, to = 400, by = 50),
    subtitle = years[[id]],
    title = "Life Expectancy Across Continents",
    x_title = "Life Expectancy",
    y_title = "Population(1e+6)",
    pts_size = 5,
    legend_key_width = 0.8,
    legend_key_height = 0.6,
    show_minor_grids = FALSE,
    silent_NA_warning = TRUE
  )
  return(aplot)
}
plot_lst <- purrr::map(1:4,
  build_plot,
  dt = RplotterPkg::gapminder_data,
  years = years
)
layout <- list(
  plots = plot_lst,
  rows = c(1, 1, 1, 1),
  cols = c(1, 2, 3, 4)
)
RplotterPkg::multi_panel_grid(
  layout = layout,
  title = "Life expectancy vs population(millions) across continents",
  plot_titles = c("1952","1972","1992","2002"),
  cell_height = 12
)
```

organdata *organdata data*

Description

From the socviz::organdata dataset two variables are selected: Organ Donation rate per million population and country.

country	character country name
donors	Organ Donation rate per million population

Usage

```
organdata
```

Format

An object of class data.table (inherits from data.frame) with 204 rows and 2 columns.

Source

socviz::organdata

penguins_stats *penguins_stats data*

Description

The palmerpenguins:penguins dataset is grouped by the variable "species" and a data frame is produced with statistics for average, minimum and maximum body mass.

avg_body_mass	numeric species average body mass
min_body_mass	numeric species minimum body mass
max_body_mass	numeric species maximum body mass

Usage

```
penguins_stats
```

Format

An object of class data.table (inherits from data.frame) with 3 rows and 4 columns.

Source

palmerpenguins:penguins

people *people data*

Description

A made-up data frame with generic column names and mix of numeric and character data.

col_0	numeric random numbers for ids
col_1	character first names
col_2	character last names
col_3	numeric ages
col_4	character addresses
col_5	character spouse's first name

Usage

```
people
```

Format

An object of class `data.table` (inherits from `data.frame`) with 4 rows and 6 columns.

Source

R coding with `data.table::data.table()`

percentile_table *percentile_table*

Description

Function locates values corresponding to percentiles in a vector of numeric data. Function displays a table with lower/upper locations, values, average and spread between lower/upper from median to one hundred and twenty-eighth percentiles. The function returns a list with a viewable gt table along with the data.table containing the table's values.

The "label" column of the table correspond to the following percentiles:

1. "M" = Half (i.e. the median)
2. "F" = Fourth (i.e. quartiles 25/75)
3. "E" = Eighth
4. "D" = Sixteenth
5. "C" = Thirty-secondth
6. "B" = Sixty-fourth
7. "A" = One hundred and twenty-eighth

Usage

```
percentile_table(vals = NULL, na_rm = TRUE, title = NULL, table_width = NULL)
```

Arguments

vals	A required vector of numeric values
na_rm	A logical which if TRUE removes NA values before the calculations
title	A string that sets the table's title. The string can contain R markdown/html style syntax.
table_width	A numeric that sets the overall table width in pixels.

Value

A named list with a gt object and a data.table with percentile values contained in the display table.

Examples

```
library(gt)
library(data.table)
library(RplotterPkg)

set.seed(12345)
random_vals <- stats::rnorm(n = 1000, mean = 20, sd = 4)
percentile_random_lst <- RplotterPkg::percentile_table(
  vals = random_vals
)
percentile_random_lst$table_gt
```

---

religion *religion data*

Description

From the socviz::gss_sm dataset General Social Survey, 2016, a sub-data frame was created from the "happy" and "religion" variables.

happy	general happiness
religion	religion in six categories

Usage

```
religion
```

Format

An object of class data.table (inherits from data.frame) with 2844 rows and 2 columns.

Source

socviz::gss_sm

---

sleep *sleep data*

Description

A dataframe with 20 observations that shows effect of two soporific drugs (increase in hours of sleep compared to control) on 10 patients.

Hours	extra numeric increase in hours of sleep
Drug	factor with 2 levels of drug groups (1,2)
ID	factor with 10 levels of patient ID (1 to 10)
Observation	numeric observation id (1 to 20)

Usage

sleep

Format

An object of class data.table (inherits from data.frame) with 20 rows and 4 columns.

Source

datasets::sleep

---

spinrates *spinrates data*

Description

From the raw data file spinrates.csv the comma delimited data was converted to a data frame using data.table::fread(). The data are baseball percent "swing and miss" with their respective average ball velocity and levels of spinrate.

id	character giving the observation id
velocity	numeric average velocity of the ball
spinrate	character factor with 12 levels
swing_miss	numeric rate of swing and miss

Usage

spinrates

Format

An object of class `data.table` (inherits from `data.frame`) with 168 rows and 4 columns.

Source

Fangraphs

`spread_level_plot` *spread_level_plot*

Description

Function produces a scatter plot of median versus spread across a variable's factor levels.

The spread is defined as the difference between the 75th and 25th quartiles. Function returns a named list with a data.table of each level's median, quartile values, and a ggplot2 scatter plot with medians along the x axis and spreads along the y axis.

Usage

```
spread_level_plot(
  df,
  meas_var,
  factor_var,
  plot_line_fit = TRUE,
  plot_log10 = TRUE,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = "Median",
  y_title = "Spread",
  x_limits = NULL,
  x_major_breaks = waiver(),
  y_limits = NULL,
  y_major_breaks = waiver(),
  label_pts = TRUE
)
```

Arguments

`df`	The required target data frame with a measure variable and a factor variable with multiple levels.
`meas_var`	A required string that names the measure variable from 'df'.
`factor_var`	A required string that names the factor variable from 'df'.
`plot_line_fit`	A logical which if `TRUE` plots a line fit between median and spread values.

plot_log10	A logical which if TRUE will plot log10 values of median versus spread instead of raw values.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL then the y axis title does not appear.
x_limits	A numeric 2 element vector that sets the minimum and maximum for the x axis.
x_major_breaks	A numeric vector or function that defines the exact major tic locations along the x axis.
y_limits	A numeric 2 element vector that sets the minimum and maximum for the y axis.
y_major_breaks	A numeric vector or function that defines the exact major tic locations along the y axis.
label_pts	A logical which if TRUE will label the plot points.

Value

A list object with a data.table and ggplot2 scatter plot.

Examples

```
library(here)
library(data.table)
library(ggplot2)
library(RplotterPkg)

spread_level_lst <- RplotterPkg::spread_level_plot(
  df = RplotterPkg::homeruns_2000,
  meas_var = "HOMERUNS",
  factor_var = "YEARS",
  x_title = "Log Median",
  y_title = "Log Spread"
)
spread_level_lst$scatter_plot
```

---

states *states data*

Description

Consolidation of data into one dataframe for the "South" region.

State	character state name
Income	numeric state income
Murder	numeric state murder rate

Usage

```
states
```

Format

An object of class `data.table` (inherits from `data.frame`) with 16 rows and 3 columns.

Source

`datasets::state_name, state_region, state_x77`

---

stem_leaf_display *stem_leaf_display*

Description

Function is a wrapper around `aplpack::stem.leaf` that provides one or more stem and leaf display(s).

Function accepts a named list of numeric vectors from which stem and leaf displays are provided.

Usage

```
stem_leaf_display(
  x = NULL,
  title = NULL,
  unit = 1,
  m = 1,
  min_val = NULL,
  max_val = NULL,
  outliers = TRUE,
  depths = FALSE,
  col_width = 4,
  row_height = 0.5,
  font_sz = 11,
  heading_color = "black"
)
```

Arguments

x	The named list of numeric vectors from which stem and leaf displays are provided.
title	A string that sets the title.
unit	Leaf unit, as a power of 10 (e.g. 100, 0.01). The default is 1.
m	Number of parts (1, 2, 5) into which each stem will be separated. The default is 1.
min_val	Optional numeric that sets the smallest non-outlying value.
max_val	Optional numeric that sets the largest non-outlying value.
outliers	A logical which if TRUE (the default), outliers are placed on LO and HI stems
depths	A logical which if TRUE (the default), print a column of "depths" to the left of the stems
col_width	A numeric that sets the display column widths in cm. The default is 4, which works when 'depths' is FALSE. You may need to increase this value to avoid cutting off wide leaf values.
row_height	A numeric that sets the display row height in cm. The default is 0.5. You may need to decrease this value for smaller font sizes and longer stem values.
font_sz	A numeric that sets the display's font size. The default is 11.
heading_color	A string that sets the heading's color in name or hex. The default is "black".

Value

A ggplot2 class object

Examples

```
library(grid)
library(gtable)
library(ggplotify)
library(ggplot2)
library(aplpack)
library(RplotterPkg)

# stem and leaf for marathon times of women across ages
marathon_times_lst <- list(
  "age_20" = RplotterPkg::boston_marathon[age == 20,]$time,
  "age_30" = RplotterPkg::boston_marathon[age == 30,]$time,
  "age_40" = RplotterPkg::boston_marathon[age == 40,]$time,
  "age_50" = RplotterPkg::boston_marathon[age == 50,]$time,
  "age_60" = RplotterPkg::boston_marathon[age == 60,]$time
)
# display stem and leaf of women times across ages
RplotterPkg::stem_leaf_display(
  x = marathon_times_lst,
  title = "Women times(min) in Boston marathon",
  heading_color = "#FF5500"
)
```

symmetry_plot *symmetry_plot*

Description

Function creates a ggplot2 based symmetry plot from a vector of numeric values.

Function orders the numeric vector from low to high. It then divides the data into two groups: values above the median(u group) and values below the median(v group). The distances from the median for values in each group are computed. A scatter plot is created where distances in the v group are plotted along the x axis and distances from the v group are plotted along the y axis. If the upper and lower distance data forms a straight line then the original data is perfectly symmetrical.

A reference symmetry line is also drawn. If points fall close to the line then the data is nearly symmetric. If the data is left-skewed, then the points fall below the line. If the data is right skewed, the points fall above the line.

Function returns a ggplot2 plot object of x/y scatter point distances from the median. Options are provided for axis scaling and point labeling.

Usage

```
symmetry_plot(
  df,
  var_name,
  position = position_jitter(width = 0, height = 0),
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  center_titles = FALSE,
  x_title = "V",
  y_title = "U",
  hide_x_tics = FALSE,
  hide_y_tics = FALSE,
  rot_x_tic_angle = 0,
  rot_y_tic_label = FALSE,
  x_limits = NULL,
  x_major_breaks = waiver(),
  x_minor_breaks = waiver(),
  x_labels = waiver(),
  y_limits = NULL,
  y_major_breaks = waiver(),
  y_minor_breaks = waiver(),
  y_labels = waiver(),
  x_y_decimals = NULL,
  x_y_scientific = NULL,
  axis_text_size = 11,
  pts_fill = "black",
  pts_shape = 21,
```

```
  pts_stroke = 1,
  pts_color = "black",
  pts_size = 1,
  pts_alpha = 1,
  line_width = 0.6,
  line_color = "blue",
  line_alpha = 0.7,
  line_linetype = "solid",
  show_major_grids = TRUE,
  show_minor_grids = TRUE,
  panel_color = "white",
  panel_border_color = "black"
)
```

Arguments

df	The target data frame from which the distance from the median are plotted.
var_name	A string that sets the variable name of interest from 'df'. This is a required parameter.
position	A string or function that does a slight adjustment to overlapping points. Typical values are "jitter" or `position_jitter(width = 0.1, height = 0.1)`.
title	A string that sets the plot title.
subtitle	A string that sets the plot subtitle.
caption	A string that sets the plot caption
center_titles	A logical which if TRUE centers both the 'title' and 'subtitle'.
x_title	A string that sets the x axis title. If NULL (the default) then the x axis title does not appear.
y_title	A string that sets the y axis title. If NULL (the default) then the y axis title does not appear.
hide_x_tics	A logical that controls the appearance of the x axis tics.
hide_y_tics	A logical that controls the appearance of the y axis tics.
rot_x_tic_angle	A numeric that sets the angle of rotation for the x tic labels. When x tic labels are long, a value of 40 for this argument usually works well.
rot_y_tic_label	A logical which if TRUE rotates the y tic labels 90 degrees for enhanced readability.
x_limits	Sets the minimum and maximum for the x axis.
x_major_breaks	A numeric vector or function that defines the exact major tic locations along the x axis.
x_minor_breaks	A numeric vector or function that defines the exact minor tic locations along the x axis.
x_labels	A character vector with the same length as 'x_major_breaks', that labels the major tics.

y_limits	A numeric 2 element vector that sets the minimum and maximum for the y axis.
y_major_breaks	A numeric vector or function that defines the exact major tic locations along the y axis.
y_minor_breaks	A numeric vector or function that defines the exact minor tic locations along the y axis.
y_labels	A character vector with the same length as 'y_major_breaks', that labels the major tics.
x_y_decimals	A two element numeric vector that set the number of decimals for the x and y tic labels.
x_y_scientific	A two element logical vector that if TRUE uses scientific notation for the x and y tic labels.
axis_text_size	A numeric that sets the font size along the axis'. Default is 11.
pts_fill	A string that sets the fill color attribute of the points.
pts_shape	A numeric integer that sets the shape attribute of the points. Typical values are 21 "circle", 22 "square", 23 "diamond", 24 "up triangle", 25 "down triangle".
pts_stroke	A numeric that sets the drawing stroke width attribute for a point shape.
pts_color	A string that sets the color attribute of the points.
pts_size	A numeric value that sets the size attribute of the points.
pts_alpha	A numeric value that sets the alpha level attribute of points.
line_width	A numeric value that sets the width of the symmetry line.
line_color	A string that sets the color of the symmetry line.
line_alpha	A numeric that sets the alpha of the symmetry line.
line_linetype	A string that sets the symmetry line type "twodash", "solid", "longdash", "dotted", "dotdash", "dashed", "blank".
show_major_grids	A logical that controls the appearance of major grids.
show_minor_grids	A logical that controls the appearance of minor grids.
panel_color	A string in hexidecimal or color name that sets the plot panel's color. The default is "white".
panel_border_color	A string in hexidecimal or color name that sets the plot panel's border color. The default is "black".

Value

A ggplot class plot object

Examples

```
library(ggplot2)
library(RplotterPkg)

RplotterPkg::symmetry_plot(
```

```
  df = RplotterPkg::farms,
  var_name = "count",
  title = "Symmetry in farm counts across US states",
  rot_y_tic_label = TRUE,
  line_linetype = "dotted"
)
```

`world_coffee` *world_coffee data*

Description

World coffee production by country (in thousands of 60 kg bags) is joined with a simple feature of country geometries via their common parameter 'name_long'.

name_long	character long name of country
coffee_production_2017	numeric of coffee production of 60 kg bags
geom	the sfc_MULTIPOLYGON country geometries

Usage

```
world_coffee
```

Format

An object of class `sf` (inherits from `data.table`, `data.frame`) with 34 rows and 3 columns.

Source

`spData::coffee_data` and `spData::world`

Index