Package: utilitybeltgg (via r-universe)

August 31, 2024

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

Imports assertthat, forcats, ggplot2, ggpubr, ggthemes, magrittr, rlang, scales

Suggests lifecycle, testthat (>= 3.0.0)

Config/testthat/edition 3

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

RemoteUrl https://github.com/selkamand/utilitybeltgg

RemoteRef HEAD

RemoteSha 94bdae63af4ff3b6162e001f609d0c5c2ad78ca1

Contents

`geom_barplot_counts` *geom barplot counts*

Description

Add text labels indicating counts above barplot columns. Works on plots that have a geom_bar() layer. Works even if you flip axis with coord_flip().

Usage

```
geom_barplot_counts(
  distance_from_bar = 1.5,
  orientation = "h",
  size = 4,
  fontface = "bold",
  alpha = 0.8,
  color = "black",
  family = "Helvetica"
)
```

Arguments

`distance_from_bar`	distance between text and cbar
`orientation`	orientation of barplot ("h" / "horizontal" / "v" / "vertical")
`size`	size of text (number)
`fontface`	Font face ("plain", "italic", "bold", "bold.italic") (string)
`alpha`	transparancy (number)
`color`	colour (string)
`family`	font family (string)

Value

ggplot geom

Examples

```
mtcars %>%
ggplot2::ggplot(ggplot2::aes(x=as.character(cyl))) +
ggplot2::geom_bar() +
ggplot2::xlab("cylinders") +
geom_barplot_counts()
```

`geom_crossbar_predefined` *Add crossbar to ggplot*

Description

Adds a crossbar to a ggplot. Best used when comparing one categorical and one numeric variable using geom_point / geom_jitter(height=0).

Usage

```
geom_crossbar_predefined(
  summaryfunction = stats::median,
  width = 0.4,
  size = 0.3,
  colour
)
```

Arguments

`summaryfunction`	a function run on the y aesthetic to determine where line is drawn. Options include median, mean, max, min, or any other function that summarises a numeric vector into a single number (function)
`width`	width of crossbar
`size`, `colour`	ggplot aesthetics

Value

ggplot geom

Examples

```
mtcars %>%
  ggplot2::ggplot(ggplot2::aes(cyl>6, mpg)) +
  ggplot2::geom_point() +
  geom_crossbar_predefined()
```

ggbarplot *Plot*

Description

Plot

Usage

```
ggbarplot(
  data,
  col,
  orientation = "h",
  fill = NULL,
  title = NULL,
  position = "stack",
  ...
)
```

Arguments

data	a dataset to plot (dataframe)
col	column of dataframe to plot (don't quote)
orientation	orientation of barplot ("h" / "horizontal" / "v" / "vertical")
fill	column of dataframe to color based on (don't quote)
title	title of graph (string)
position	see ?ggplot2::geom_bar for details (Usually one of: "stack", "dodge", "fill")
...	set any other barplot property. See ?ggplot2::geom_bar for details. (e.g. alpha = 0.4)

Value

ggplot

Examples

```
ggbarplot(iris, Species, fill = Species, orientation = "horizontal", title = "My Freq Graph")
```

ggdensity *Density Plot*

Description

Density Plot

Usage

```
ggdensity(
  data,
  col,
  fill = "steelblue",
  alpha = 0.5,
  summary_stats = FALSE,
  text_xpos = NULL,
  text_ypos = 0.04,
  text_sigfigs = Inf,
  ...
)
```

Arguments

data	datafrmae
col	column (don't quote)
fill	geom_density fill color
alpha	geom_density alpha color
summary_stats	add text with summary stats to plot (boolean)
text_xpos	x position of summary stat text (numeric)
text_ypos	y position of summary stat text (numeric)
text_sigfigs	number of significant figures to write summary statistic to (numeric)
...	other geom_density aesthetics (e.g. linetype = "dashed")

Value

gglot

Examples

```
ggdensity(mtcars, mpg, summary = TRUE)
```

`scale_show_infinites_x` *Include Infinite Values in Plot*

Description

Will replace infinite values by the nearest limit.

Usage

```
scale_show_infinites_x(
  trans = "identity",
  limits = NULL,
  position = "bottom",
  breaks = ggplot2::waiver(),
  ...
)
```

Arguments

`trans`	For continuous scales, the name of a transformation object or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "date", "exp", "hms", "identity", "log", "log10", "log1p", "log2", "logit", "modulus", "probability", "probit", "pseudo_log", "reciprocal", "reverse", "sqrt" and "time". A transformation object bundles together a transform, its inverse, and methods for generating breaks and labels. Transformation objects are defined in the scales package, and are called `<name>_trans` (e.g., `scales::boxcox_trans()`). You can create your own transformation with `scales::trans_new()`.
`limits`	One of: • `NULL` to use the default scale range • A numeric vector of length two providing limits of the scale. Use `NA` to refer to the existing minimum or maximum • A function that accepts the existing (automatic) limits and returns new limits. Also accepts rlang lambda function notation. Note that setting limits on positional scales will **remove** data outside of the limits. If the purpose is to zoom, use the limit argument in the coordinate system (see `coord_cartesian()`).
`position`	For position scales, The position of the axis. `left` or `right` for y axes, `top` or `bottom` for x axes.
`breaks`	One of: • `NULL` for no breaks • `waiver()` for the default breaks computed by the transformation object • A numeric vector of positions

- A function that takes the limits as input and returns breaks as output (e.g., a function returned by `scales::extended_breaks()`). Also accepts rlang lambda function notation.

`...` Other arguments to ggplot2::scale_x_continuous or ggplot2::scale_y_continuous

Details

This function also allows you to pass arguments along to `ggplot2::scale_x_continuous()` since once you add this call you won't be able to edit the continuous scale elsewhere. Oftentimes you'll want to use the `trans` argument to, for example, visualize your data on a log scale.

Value

ScaleContinuousPosition object that can be added to a ggplot object using '+'

See Also

`scale_show_infinites_y()`

`ggplot2::scale_x_continuous()`

Examples

```
## Not run:
data = mtcars
    dplyr::mutate(qsec2 = ifelse(qsec > 19, Inf, qsec))

# Plot with infinites set to nearest limit
data %>%
    ggplot2::ggplot(ggplot2::aes(x=qsec2, y=carb)) +
    ggplot2::geom_point() +
    scale_show_infinites_x()

# Plot on a log10 scale with infinites set to nearest limit
data %>%
    ggplot2::ggplot(ggplot2::aes(x=qsec2, y=carb)) +
    ggplot2::geom_point() +
    scale_show_infinites_x(trans="log10")

## End(Not run)
```

---

scale_show_infinites_y

Include Infinite Values in Plot

Description

Will replace infinite values by the nearest limit.

Usage

```
scale_show_infinites_y(
  trans = "identity",
  limits = NULL,
  position = "bottom",
  breaks = ggplot2::waiver(),
  ...
)
```

Arguments

trans	For continuous scales, the name of a transformation object or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "date", "exp", "hms", "identity", "log", "log10", "log1p", "log2", "logit", "modulus", "probability", "probit", "pseudo_log", "reciprocal", "reverse", "sqrt" and "time". A transformation object bundles together a transform, its inverse, and methods for generating breaks and labels. Transformation objects are defined in the scales package, and are called <name>_trans (e.g., scales::boxcox_trans()). You can create your own transformation with scales::trans_new().
limits	One of: • NULL to use the default scale range • A numeric vector of length two providing limits of the scale. Use NA to refer to the existing minimum or maximum • A function that accepts the existing (automatic) limits and returns new limits. Also accepts rlang lambda function notation. Note that setting limits on positional scales will **remove** data outside of the limits. If the purpose is to zoom, use the limit argument in the coordinate system (see coord_cartesian()).
position	For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.
breaks	One of: • NULL for no breaks • waiver() for the default breaks computed by the transformation object • A numeric vector of positions • A function that takes the limits as input and returns breaks as output (e.g., a function returned by scales::extended_breaks()). Also accepts rlang lambda function notation.
...	Arguments passed on to scale_show_infinites_x

Value

ScaleContinuousPosition object that can be added to a ggplot object using '+'

See Also

scale_show_infinites_x()

Examples

```
## Not run:
data = mtcars %>%
    dplyr::mutate(qsec2 = ifelse(qsec > 19, Inf, qsec))

# Plot with infinites set to nearest limit
data %>%
    ggplot2::ggplot(ggplot2::aes(x=carb, y=qsec2)) +
    ggplot2::geom_point() +
    scale_show_infinites_y()

# Plot on a log10 scale with infinites set to nearest limit
data %>%
    ggplot2::ggplot(ggplot2::aes(x=carb, y=qsec2)) +
    ggplot2::geom_point() +
    scale_show_infinites_y(trans="log10")

## End(Not run)
```

`theme_axis_titles_cleveland` *Custom Themes*

Description

Custom Themes

Usage

```
theme_axis_titles_cleveland()
```

Value

ggtheme

`theme_common_adjustments` *Custom Themes*

Description

ggplot theme that fixes a bunch of issues I tend to have with plots.

Usage

```
theme_common_adjustments(
  dist_from_plot_xlab = 10,
  dist_from_plot_ylab = 10,
  dist_from_plot_ggtitle = 10,
  no_background = FALSE,
  subtitle_face = "plain"
)
```

Arguments

dist_from_plot_xlab	distance between x axis title and plot (number)
dist_from_plot_ylab	distance between y axis title and plot (number)
dist_from_plot_ggtitle	distance between y axis title and plot (number)
no_background	set all backgrounds to clear? (flag)
subtitle_face	Font face ("plain", "italic", "bold", "bold.italic")

Details

By default, will bold center axis and plot titles and tweak distane of axis_titles to plot

Value

ggtheme

Examples

```
mtcars %>%
  ggplot2::ggplot(ggplot2::aes(cyl>6, mpg)) +
  ggplot2::geom_point() +
  theme_common_adjustments()
```

---

theme_fivethirtyeight_two

Custom Themes

Description

Custom Themes

Usage

```
theme_fivethirtyeight_two()
```

Value

ggtheme

`theme_legend_bottom` *Custom Themes*

Description

Custom Themes

Usage

```
theme_legend_bottom(direction = "horizontal")
```

Arguments

`direction`	"Vertical or Horizontal"

Value

ggtheme

`theme_legend_left` *Custom Themes*

Description

Custom Themes

Usage

```
theme_legend_left(direction = "vertical")
```

Arguments

`direction`	"Vertical or Horizontal"

Value

ggtheme

theme_legend_none *Custom Themes*

Description

Custom Themes

Usage

```
theme_legend_none()
```

Value

ggtheme

theme_legend_right *Custom Themes*

Description

Custom Themes

Usage

```
theme_legend_right(direction = "vertical")
```

Arguments

direction	"Vertical or Horizontal"

Value

ggtheme

theme_legend_title_none *Custom Themes*

Description

Custom Themes

Usage

```
theme_legend_title_none()
```

Value

ggtheme

theme_legend_top *Custom Themes*

Description

Custom Themes

Usage

```
theme_legend_top(direction = "horizontal")
```

Arguments

direction	"Vertical or Horizontal"

Value

ggtheme

theme_minimal_bordered *Miminal theme with border*

Description

A minimal theme with a border. Works well for faceted graphs

Usage

```
theme_minimal_bordered(border_color = "grey40", border_thickness = NULL, ...)
```

Arguments

border_color	Colour of border
border_thickness	border thickness in mm
...	Arguments passed on to ggplot2::theme_minimal base_size base font size, given in pts. base_family base font family base_line_size base size for line elements base_rect_size base size for rect elements

Examples

```
mtcars %>%
  ggplot2::ggplot(ggplot2::aes(x = mpg, y=disp)) +
  ggplot2::geom_point() +
  theme_minimal_bordered()
```

theme_no_legend *Remove theme legend*

Description

[Deprecated]

Usage

```
theme_no_legend(...)
```

Arguments

...	no arguments are used. Included only so code written for older versions of package doesn't break

Value

theme

`theme_no_legend_title` *Remove theme legend*

Description

[Deprecated]

Usage

```
theme_no_legend_title(...)
```

Arguments

...	no arguments are used. Included only so code written for older versions of package doesn't break

Value

theme

Index