Package ‘ggcorset’

December 11, 2022

Type Package

Title The Corset Plot

Version 0.4.5

Author Kyla Belisario

Maintainer Kyla Belisario <kyla_belisario@protonmail.com>

Description Corset plots are a visualization technique used strictly to visualize repeat measures at 2 time points (such as pre- and post- data). The distribution of measurements are visualized at each time point, whilst the trajectories of individual change are visualized by connecting the pre- and post- values linearly. These lines can be coloured to represent the magnitude of change, or other user-defined value. This method of visualization is ideal for showing the heterogeneity of data, including differences by sub-groups. The package relies on 'ggplot2' allowing for easy integration so that users can customize their visualizations as required. Users can create corset plots using data in either wide or long format using the functions gg_corset() or gg_corset_elongated(), respectively.

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports ggplot2, dplyr, gghalves, ggstance

Encoding UTF-8

LazyData true

RoxygenNote 7.2.2

Suggests rmarkdown, knitr, viridis, MetBrewer

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-12-11 22:20:03 UTC

R topics documented:

`drinkdays` *DRINKDAYS*

Description

An example data set from simulated data.

Usage

```
data(drinkdays)
```

Format

An object of class `data.frame` with 300 rows and 3 columns.

Examples

```
## Not run:
data(drinkdays)

## End(Not run)
```

`gg_corset` *CORSET PLOT*

Description

This function visualizes a corset plot in wide format.

Arguments

`data` The name of the data frame.

`y_var1` The name of measured variable at time 1.

`y_var2` The name of measured variable at time 2.

`group` The name of units measured at each time point such as 'ID'. The trajectories of these units are visualized by the lines of the corset plot.

`c_var` The name of variable to visualize by line colour, such as percent change, magnitude of change, or direction of change.

`eyelets` Optional (default is FALSE). If set to true, this will visualize one of two mean types by c_var, as defined by the 'e_type' argument.

`e_type` Optional eyelet type if the eyelets parameter is set to TRUE. One of "SE" or "SD". The default is standard error ("SE") means. Alternatively, standard deviations ("SD") with means can be specified, which include horizontal lines to denote +1 and -1 standard deviation. Note that the visualization of standard deviations works best in tandem with the faceted option.

`faceted` Optional (default is FALSE). If set to true, the c_var will be faceted.

`facet_design` Optional facet type when the faceted parameter is set to TRUE. One of "original","group", or "line". The default is "original", which provides facets void of any special features. The "group" option includes the overall distribution of the entire sample in the background of each facet (which defaults to the 'vio_fill' colour), alongside each distribution for each c_var group. The "line" option includes all individual trajectories in the background of each facet using a soft grey (default) or custom colour as chosen by 'line_col' argument.

`vio_fill` Optional (defaults to a soft black). Use to change the fill colour of the half violins.

`line_size` Optional. Use to change the size (thickness) of the lines which visualize the c_var. Default is 0.25.

`line_col` Optional custom colour of the background individual lines when the facet_design is set to "line". Defaults to a soft grey.

Value

ggplot2 graphical object

Examples

```
wide.df <- data.frame(id = c(1,2,3,4,5),
             time1 = c(3,4,7,5,6),
             time2  = c(5,5,7,3,0),
             change = c(28.57,14.29,0,-28.57,-85.71),
             direction = c("increase","increase","no change","decrease","decrease"))

gg_corset(data = wide.df, y_var1 = "time1", y_var2 = "time2",
          group = "id", c_var = "change")

## Create corset plots with eyelets:

gg_corset(data = wide.df, y_var1 = "time1", y_var2 = "time2",
          group = "id", c_var = "direction", eyelets = TRUE)

## Create faceted corset plots based on direction of change:

gg_corset(data = wide.df, y_var1 = "time1", y_var2 = "time2",
          group = "id", c_var = "direction", faceted = TRUE)

## Create faceted corset plots with standard deviation eyelets:

gg_corset(data = wide.df, y_var1 = "time1", y_var2 = "time2", group = "id",
          c_var = "direction", e_type = "SD", faceted = TRUE)
```

gg_corset_elongated *CORSET PLOT ELONGATED*

Description

This function visualizes a corset plot in long format.

Arguments

`data` The name of the data frame.

`x_var` The name of the x_axis variable.

`x_vals` The values of the two time points.

`y_var` The repeated measure variable name.

`group` The name of units measured at each time point such as ’ID’. The trajectories of these units are visualized by the lines of the corset plot.

`c_var` The name of variable to visualize by line colour, such as percent change.

`eyelets` Optional (default is FALSE). If set to true, this will visualize one of two mean types by c_var, as defined by the ’e_type’ argument.

`e_type` Optional eyelet type if the eyelets parameter is set to TRUE. One of "SE" or "SD". The default is standard error ("SE") means. Alternatively, standard deviations ("SD") with means can be specified, which include horizontal lines to denote +1 and -1 standard deviation. Note that the visualization of standard deviations works best in tandem with the faceted option.

`faceted` Optional (default is FALSE). If set to true, the c_var will be faceted.

`facet_design` Optional facet type when the faceted parameter is set to TRUE. One of "original","group", or "line". The default is "original", which provides facets void of any special features. The "group" option includes the overall distribution of the entire sample in the background of each facet (which defaults to the ’vio_fill’ colour), alongside each distribution for each c_var group. The "line" option includes all individual trajectories in the background of each facet using a soft grey (default) or custom colour as chosen by ’line_col’ argument.

`vio_fill` Optional (defaults to a soft black). Use to change the fill colour of the half violins.

`line_size` Optional. Use to change the size (thickness) of the lines which visualize the c_var. Default is 0.25.

`line_col` Optional custom colour of the background individual lines when the facet_design is set to "line". Defaults to a soft grey.

Value

ggplot2 graphical object

Examples

```
long.df <- data.frame(id = c(1,1,2,2,3,3),
             time = c("pre","post","pre","post","pre","post"),
             days  = c(3,5,4,0,7,7),
             change = c(28.57,28.57,-57.14,-57.14,0,0),
         direction = c("increase","increase","decrease","decrease","no change","no change"))

gg_corset_elongated(data = long.df, x_var = "time",
                   x_vals = c("pre","post"), y_var = "days",
                   group = "id", c_var = "change")

## Create groupings based on direction of change to use for eyelets:

gg_corset_elongated(data = long.df, x_var = "time", x_vals = c("pre","post"),
                    y_var = "days", group = "id", c_var = "direction", eyelets = TRUE)

## Create faceted corset plots based on direction of change:

gg_corset_elongated(data = long.df, x_var = "time", x_vals = c("pre","post"),
                    y_var = "days", group = "id", c_var = "direction", faceted = TRUE)

## Create faceted corset plots with standard deviation eyelets:

gg_corset_elongated(data = long.df, x_var = "time", x_vals = c("pre","post"),
                    y_var = "days", group = "id", c_var = "direction",
                    e_type = "SD", faceted = TRUE)
```

theme_ggcorset *THEME_GGCORSET*

Description

This function offers a ggplot theme to make visualizations more polished.

Usage

```
theme_ggcorset()
```

Value

ggplot2 theme

Examples

```
wide.df <- data.frame(id = c(1,2,3,4,5),
             time1 = c(3,4,7,5,6),
             time2  = c(5,5,7,3,0),
             change = c(28.57,14.29,0,-28.57,-85.71))

plot1 <- gg_corset(data = wide.df, y_var1 = "time1", y_var2 = "time2",
          group = "id", c_var = "change")

plot1 + theme_ggcorset()
```

Index