Package ‘ggupset’

March 6, 2019

Type Package
Title Combination Matrix Axis for 'ggplot2' to Create 'UpSet' Plots
Version 0.1.0
URL https://github.com/const-ae/ggupset
BugReports https://github.com/const-ae/ggupset/issues
Description Replace the standard x-axis in 'ggplots' with a combination matrix to visualize complex set overlaps. 'UpSet' has introduced a new way to visualize the overlap of sets as an alternative to Venn diagrams. This package provides a simple way to produce such plots using 'ggplot2'. In addition it can convert any categorical axis into a combination matrix axis.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Depends R (>= 2.10)
Suggests testthat
Imports ggplot2, gtable, grid, tibble, rlang, scales
NeedsCompilation no
Author Constantin Ahlmann-Eltze [aut, cre]
Maintainer Constantin Ahlmann-Eltze <artjom31415@googlemail.com>
Repository CRAN
Date/Publication 2019-03-06 13:30:03 UTC

R topics documented:

axis_combmatrix .......................................................... 2
df_complex_conditions .............................................. 3
scale_x_mergelist ...................................................... 3
axis_combmatrix

Convert delimited text labels into a combination matrix axis

Description
The function splits the text based on the sep argument and views each occurring element as potential set.

Usage
axis_combmatrix(sep = "[^[:alnum:]]+", levels = NULL, xlim = NULL, ylim = NULL, expand = TRUE, clip = "on", ytrans = "identity")

Arguments
sep The separator that is used to split the string labels. Can be a regex. Default: "[^[:alnum:]]+"
levels The selection of string elements that are displayed in the combination matrix axis. Default: NULL, which means simply all elements in the text labels are used
xlim, ylim The limits for the x and y axes
expand Boolean with the same effect as in ggplot2::coord_cartesian(). Default: TRUE
clip String with the same effect as in ggplot2::coord_cartesian(). Default: "on"
ytrans transformers for y axis. For more information see ggplot2::coord_trans(). Default: "identity"

Details
Technically the function appends a coord system to the ggplot object. To maintain compatibility additional arguments like ytrans, ylim, and clip are forwarded to coord_trans().

Examples
library(ggplot2)
mtcars$combined <- paste0("Cyl: ", mtcars$cyl, "Gears: ", mtcars$gear)
head(mtcars)
ggplot(mtcars, aes(x=combined)) +
  geom_bar() +
  axis_combmatrix(sep = ",")
df_complex_conditions  

A fictional biological dataset with a complex experimental design

Usage

df_complex_conditions

Format

a data frame with 360 rows and 4 variables

- KO. Boolean value if the sample had a knock out.
- DrugA. character vector with "Yes" and "No" elements indicating if the sample was treated with drug A.
- Timepoint. Numeric vector with elements 8, 24, and 48 indicating the time of measurement since the beginning of the experiment.
- response. Numeric vector with the response of the sample to the treatment conditions. Could for example be the concentration of a metabolite.

Examples

dim(df_complex_conditions)
head(df_complex_conditions)

scale_x_mergelist  

Merge list columns into character vectors

Description

The function handles list columns by collapsing them into delimited strings using the sep argument. This is useful to show sets and in combination with the axis_combmatrix() function.

Usage

scale_x_mergelist(sep = "-", ..., position = "bottom")

Arguments

sep  
String the is used to delimit the elements in each list entry. Default: 

...  
additional arguments that are passed on to ggplot2::scale_x_discrete

position  
either "top" or "bottom" to specify where the x axis drawn. Default: "bottom"
See Also
discrete_scale

Examples
library(ggplot2)
ggplot(tidy_movies[1:100, ], aes(x=Genres)) +
ggplot(tidy_movies[1:100, ], aes(x=Genres)) +

scale_x_upset(order_by = c("freq", "degree"),
n_sets = Inf, n_intersections = Inf,
sets = NULL, intersections = NULL,
reverse = FALSE, ytrans = "identity", ..., position = "bottom")

Description
This function takes a list column and turns it into a combination matrix axis. It internally wraps
the call to scale_x_mergelist() and axis_combmatrix() and makes sure that the elements are
sorted by size.

Usage
scale_x_upset(order_by = c("freq", "degree"), n_sets = Inf,
               n_intersections = Inf, sets = NULL, intersections = NULL,
               reverse = FALSE, ytrans = "identity", ..., position = "bottom")

Arguments
order_by either "freq" or "degree". Default: "freq"
n_sets maximum number of sets that are displayed. Default: Inf
n_intersections maximum number of intersections that are displayed. Default: Inf
sets character vector that specifies which sets are displayed
intersections a list of character vectors that specifies which intersections are displayed
reverse boolean if the order of the intersections is reversed. Default: FALSE
ytrans transformers for y axis. For more information see axis_combmatrix(). De-
fault: "identity"
... additional parameters for ggplot2::discrete_scale()
position either "top" or "bottom" to specify where the combination matrix is drawn. De-
fault: "bottom"
theme_combmatrix

Examples

library(ggplot2)

```r
ggplot(tidy_movies[1:100,], aes(x=Genres)) +
  geom_bar() +
  scale_x_upset(reverse = TRUE, sets=c("Drama", "Action"))

ggplot(tidy_movies[1:100,], aes(x=Genres)) +
  geom_bar() +
  scale_x_upset(n_intersections = 5, ytrans="sqrt")

ggplot(tidy_movies[1:100,], aes(x=Genres, y=year)) +
  geom_boxplot() +
  scale_x_upset(intersections = list(c("Drama", "Comedy"), c("Short"), c("Short", "Animation")),
                 sets = c("Drama", "Comedy", "Short", "Animation", "Horror"))
```

theme_combmatrix  Theme for the combination matrix

Description

This theme sets the default styling for the combination matrix axis by extending the default ggplot2 theme().

Usage

```r
theme_combmatrix(combmatrix.label.make_space = TRUE,
                  combmatrix.label.width = NULL, combmatrix.label.height = NULL,
                  combmatrix.label.extra_spacing = 3,
                  combmatrix.label.total_extra_spacing = unit(10, "pt"),
                  combmatrix.label.text = NULL, combmatrix.panel.margin = unit(c(1.5, 1.5), "pt"),
                  combmatrix.panel.stripped_background = TRUE,
                  combmatrix.panel.stripped_background.color.one = "white",
                  combmatrix.panel.stripped_background.color.two = "#F7F7F7",
                  combmatrix.panel.point.size = 3, combmatrix.panel.line.size = 1.2,
                  combmatrix.panel.point.color.fill = "black",
                  combmatrix.panel.point.color.empty = "#E0E0E0", ...)
```

Arguments

- **combmatrix.label.make_space**
  
  Boolean indicator if the y-axis label is moved so far to the left to make enough space for the combination matrix labels. Default: TRUE

- **combmatrix.label.width**
  
  A unit that specifies how much space to make for the labels of the combination matrix. Default: NULL, which means the width of the label text is used
**theme_combmatrix**

- **combmatrix.label.height**
  A unit that specifies how high the combination matrix should be. Default: NULL, which means that the height of the label text + combmatrix.label.total_extra_spacing + #rows * combmatrix.label.extra_spacing is used. Default: 3

- **combmatrix.label.extra_spacing**
  A single number for the additional height per row. Default: unit(10, "pt")

- **combmatrix.label.total_extra_spacing**
  A unit that specifies the total offset for the height of the combination matrix

- **combmatrix.label.text**
  A `element_text()` to style the label text of the combination matrix. Default NULL, which means the style of `axis.text.y` is used.

- **combmatrix.panel.margin**
  A two element unit vector to specify top and bottom margin around the combination matrix. Default: unit(c(1.5, 1.5), "pt")

- **combmatrix.panel.striped_background**
  Boolean to indicate if the background of the plot is striped. Default: TRUE

- **combmatrix.panel.striped_background.color.one**
  Color of the first kind of stripes. Default: "white"

- **combmatrix.panel.striped_background.color.two**
  Color of the second kind of stripes. Default: "#F7F7F7"

- **combmatrix.panel.point.size**
  Number to specify the size of the points in the combination matrix. Default: 3

- **combmatrix.panel.line.size**
  Number to specify the size of the lines connecting the points. Default: 1.2

- **combmatrix.panel.point.color.fill**
  Color of the filled points. Default: "black"

- **combmatrix.panel.point.color.empty**
  Color of the empty points. Default: "#E0E0E0"

- **additional arguments** that are passed to `theme()`

**Examples**

```r
library(ggplot2)
# Ensure that the y-axis label is next to the axis by setting
# combmatrix.label.make_space to FALSE
ggplot(tidy_movies[1:100, ], aes(x=Genres)) +
  geom_bar() +
  scale_x_upset() +
  theme_combmatrix(combmatrix.label.text = element_text(color = "black", size=15),
                   combmatrix.label.make_space = FALSE,
                   plot.margin = unit(c(1.5, 1.5, 1.5, 65), "pt"))

# Change the color of the background stripes
ggplot(tidy_movies[1:100, ], aes(x=Genres)) +
  geom_bar() +
  scale_x_upset() +
  theme_combmatrix(combmatrix.panel.striped_background = TRUE,
                   combmatrix.panel.striped_background.color.one = "grey")
```
**tidy_movies**

*Description*

The original `ggplot2movies::movies` dataset has 7 columns that contain indicators if a movie belongs to a certain genre. In this version the 7 columns are collapsed to a single list column to create a tidy dataset. It also has information on only 5,000 movies to reduce the size of the dataset. Furthermore each star rating is in its own row.

**Usage**

`tidy_movies`

**Format**

a data frame with 50,000 rows and 10 columns

- **title.** The title of the movie.
- **year.** Year of release.
- **budget.** Total budget (if known) in US dollars.
- **length.** Length in minutes.
- **rating.** Average IMDB user rating.
- **votes.** Number of IMDB user who rated this movie.
- **mpaa.** MPAA rating.
- **Genres.** List column with all genres the movie belongs to.
- **stars, percent_rating.** The number of stars and the corresponding percentage of people rating the movie with this many stars.

**Examples**

```r
dim(tidy_movies)
head(tidy_movies)
```
Index

*Topic **datasets**
  - df_complex_conditions, 3
  - tidy_movies, 7
axis_combmatrix, 2

df_complex_conditions, 3
discrete_scale, 4

scale_x_mergelist, 3
scale_x_upset, 4

theme_combmatrix, 5
tidy_movies, 7