Package ‘ggupset’

February 11, 2025

Type Package

Title Combination Matrix Axis for 'ggplot2' to Create 'UpSet' Plots

Version 0.4.1

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 7.3.1

Depends R (>= 2.10)

Suggests testthat

Imports ggplot2 (>= 3.3.0), gtable, grid, tibble, rlang, scales

NeedsCompilation no

Author Constantin Ahlmann-Eltze [aut, cre]
(<`https://orcid.org/0000-0002-3762-068X`>)

Maintainer Constantin Ahlmann-Eltze <`artjom31415@googlemail.com`>

Repository CRAN

Date/Publication 2025-02-11 11:10:02 UTC

Contents

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,
  override_plotting_function = 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

override_plotting_function
to achieve maximum flexibility, you can provide a custom plotting function. For more information, see details. Default: NULL

xlim, ylim The limits fort 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()`.

Note: make sure that the argument to the 'x' aesthetic is character vector that contains the `sep` sequence. The only exception is if `axis_combmatrix()` is combined with a `scale_x_mergelist()`. This pattern works because in the first step `scale_x_mergelist()` turns a list argument to 'x' into a character vector that `axis_combmatrix()` can work with.

For maximum flexibility, you can use the 'override_plotting_function' parameter which returns a ggplot and is called with a `tibble` with one entry per point of the combination matrix. Specifically, it contains

labels the collapsed label string

single_label an ordered factor with the labels on the left of the plot

id consecutive numbering of the points

labels_split a list column that contains the splitted labels

at the x-position of the point

observed boolean to indicate if this element is active in the intersection

index the row of the point

See the examples how the `override_plotting_function` looks that recreates the default combination matrix

Examples

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

# Example of 'override_plotting_function'

ggplot(mtcars, aes(x=combined)) +
  geom_bar() +
    axis_combmatrix(sep = "_", override_plotting_function = function(df){
      ggplot(df, aes(x= at, y= single_label)) +
        geom_rect(aes(fill= index %% 2 == 0), ymin=df$index-0.5,
                  ymax=df$index+0.5, xmin=0, xmax=1) +
        geom_point(aes(color= observed), size = 3) +
        geom_line(data= function(dat) dat[dat$observed, ,drop=FALSE],
                  aes(group = labels), size= 1.2) +
        ylab("") + xlab("") +
        scale_x_continuous(limits = c(0, 1), expand = c(0, 0)) +
        scale_fill_manual(values= c(`TRUE` = "white", `FALSE` = "#F7F7F7")) +
        scale_color_manual(values= c(`TRUE` = "black", `FALSE` = "#E0E0E0")) +
        guides(color="none", fill="none") +
        theme(
          panel.background = element_blank(),
          axis.text.x = element_blank(),
          axis.ticks.y = element_blank(),
          axis.ticks.length = unit(0, "pt"),
          axis.title.y = element_blank(),
          axis.title.x = element_blank(),
          axis.line = element_blank(),
          panel.border = element_blank()
        )
    })
```

df_complex_conditions *A fictional biological dataset with a complex experimental design*

Description

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)
```

gene_pathway_membership *A fictional dataset describing which genes belong to certain pathways*

Description

A fictional dataset describing which genes belong to certain pathways

Usage

```
gene_pathway_membership
```

Format

a matrix with 6 rows and 37 columns. Each row is one pathway, with its name given as ‘rownames‘ and each column is a gene. The values in the matrix are Boolean indicators if the gene is a member of the pathway.

Examples

```
dim(gene_pathway_membership)
gene_pathway_membership[, 1:15]
```

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)) +
  geom_bar() +
  scale_x_mergelist() +
  theme(axis.text.x = element_text(angle = 90, hjust=1, vjust = 0.5))

ggplot(tidy_movies[1:100, ], aes(x=Genres)) +
  geom_bar() +
  scale_x_mergelist(sep = " & ", name = "Merged Movie Genres", position = "top") +
  theme(axis.text.x = element_text(angle = 90, hjust=0, vjust = 0.5))
```

scale_x_upset *Scale to make UpSet plots*

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()`. Default: "identity"

`...` additional parameters for `ggplot2::discrete_scale()`

`position` either "top" or "bottom" to specify where the combination matrix is drawn. Default: "bottom"

Examples

```
library(ggplot2)
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

```
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.striped_background = TRUE,
  combmatrix.panel.striped_background.color.one = "white",
  combmatrix.panel.striped_background.color.two = "#F7F7F7",
  combmatrix.panel.point.size = 3,
  combmatrix.panel.line.size = 1.2,
  combmatrix.panel.line.color = "black",
  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

`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.line.color`
Color of the lines connecting the points. Default: "black"

`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

```
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` *Tidy version of the movies dataset from the ggplot2 package*

Description

The original `ggplot2movies::movies` dataset has 7 columns that contain indicators if a movies 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 on 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

```
dim(tidy_movies)
head(tidy_movies)
```

Index