Package ‘heatmaply’

May 26, 2016

Type Package

Title Interactive Heat Maps Using 'plotly'

Version 0.3.2

Date 2016-05-25

Description Create interactive heatmaps that are usable from the R console, in the 'RStudio' viewer pane, in 'R Markdown' documents, and in 'Shiny' apps. Hover the mouse pointer over a cell to show details or drag a rectangle to zoom. A heatmap is a popular graphical method for visualizing high-dimensional data, in which a table of numbers are encoded as a grid of colored cells. The rows and columns of the matrix are ordered to highlight patterns and are often accompanied by dendrograms. Heatmaps are used in many fields for visualizing observations, correlations, missing values patterns, and more. Interactive heatmaps allow the inspection of specific value by hovering the mouse over a cell, as well as zooming into a region of the heatmap by dragging a rectangle around the relevant area. This work is based on the 'ggplot2' and 'plotly.js' engine. It produces similar heatmaps as 'd3heatmap', with the advantage of speed ('plotly.js' is able to handle larger size matrix), and the ability to zoom from the dendrogram panes.

Depends R (>= 3.0.0), plotly (>= 3.6.0), viridis

Imports ggplot2, dendextend, magrittr (>= 1.0.1), reshape2, scales, utils, stats

Suggests knitr, rmarkdown, gplots

VignetteBuilder knitr

License GPL-2 | GPL-3

URL https://cran.r-project.org/package=heatmaply, https://github.com/talgalili/heatmaply/, http://www.r-statistics.com/tag/heatmaply/

BugReports https://github.com/talgalili/heatmaply/issues

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

Author Tal Galili [aut, cre, cph] (http://www.r-statistics.com)

Maintainer Tal Galili <`tal.galili@gmail.com`>

Repository CRAN

Date/Publication 2016-05-26 17:50:25

R topics documented:

`heatmaply` *Creates a plotly heatmap*

Description

An object of class heatmapr includes all the needed information for producing a heatmap. The goal is to seperate the pre-processing of the heatmap elements from the graphical rendaring of the object, which could be done

Usage

```
heatmaply(x, colors = viridis(n = 256, alpha = 1, begin = 0, end = 1, option =
  "viridis"), limits = NULL, na.value = "grey50", row_text_angle = 0,
  column_text_angle = 45, margin = 0, row_dend_left = FALSE, ...,
  scale_fill_gradient_fun = scale_fill_gradientn(colors = colors, na.value =
  na.value, limits = limits), srtRow, srtCol)
```

Arguments

`x` can either be a heatmapr object, or a numeric matrix Defaults to `TRUE` unless `x` contains any `NA`s.

`colors` a vector of colors to use for heatmap color. The default uses `viridis(n=256, alpha = 1, begin = 0, e`
It is passed to scale_fill_gradientn.

`limits` a two dimensional numeric vector specifying the data range for the scale.

`na.value` color to use for missing values (default is "grey50").

`row_text_angle` numeric (Default is 0), the angle of the text of the rows. (this is called srtRow in heatmap.2)

`column_text_angle`
numeric (Default is 45), the angle of the text of the columns. (this is called srtCol in heatmap.2)

`margin` passed to subplot. Default is 0. Either a single value or four values (all between 0 and 1). If four values are provided, the first is used as the left margin, the second is used as the right margin, the third is used as the top margin, and the fourth is used as the bottom margin. If a single value is provided, it will be used as all four margins.

`row_dend_left` logical (default is FALSE). Should the row dendrogram be plotted on the left side of the heatmap. If false then it will be plotted on the right side.

`...` other parameters passed to heatmapr (currently, various parameters may be ignored.

`scale_fill_gradient_fun`

A function that creates a smooth gradient for the heatmap. The default uses scale_fill_gradientn with the values of colors, limits, and na.value that are supplied by the user. The user can input a customized function, such as scale_colour_gradient() in order to get other results (although the virids default is quite recommended)

`srtRow` if supplied, this overrides row_text_angle (this is to stay compatible with heatmap.2)

`srtCol` if supplied, this overrides column_text_angle (this is to stay compatible with heatmap.2)

Please submit an issue on github if you have a feature that you wish to have added)

Examples

```
## Not run:

# mtcars
# x <- heatmapr(mtcars)
library(heatmaply)
heatmaply(iris[,-5], k_row = 3, k_col = 2)
heatmaply(cor(iris[,-5]))
heatmaply(cor(iris[,-5]), limits = c(-1,1))
heatmaply(mtcars, k_row = 3, k_col = 2)

# make sure there is enough room for the labels:
heatmaply(mtcars) %>% layout(margin = list(l = 130, b = 40))

# control text angle
heatmaply(mtcars, column_text_angle = 90) %>% layout(margin = list(l = 130, b = 40))
# the same as using srtCol:
# heatmaply(mtcars, srtCol = 90) %>% layout(margin = list(l = 130, b = 40))



x <- mtcars
# different colors
heatmaply(x, colors = heat.colors(200))
# using special scale_fill_gradient_fun colors
heatmaply(x, scale_fill_gradient_fun = scale_colour_gradient())


## End(Not run)
```

heatmapr *Creates a heatmapr object*

Description

An object of class heatmapr includes all the needed information for producing a heatmap. The goal is to seperate the pre-processing of the heatmap elements from the graphical rendaring of the object, which could be done using plotly (but potentially also with other graphical devices).

Usage

```
heatmapr(x, Rowv = TRUE, Colv = if (symm) ”Rowv” else TRUE,
  distfun = dist, hclustfun = hclust, dendrogram = c(”both”, ”row”,
  ”column”, ”none”), reorderfun = function(d, w) reorder(d, w), k_row, k_col,
  symm = FALSE, revC, scale = c(”none”, ”row”, ”column”), na.rm = TRUE,
  labRow = rownames(x), labCol = colnames(x), cexRow, cexCol, digits = 3L,
  cellnote, cellnote_scale = TRUE, theme = NULL, colors = ”RdYlBu”,
  width = NULL, height = NULL, xaxis_height = 80, yaxis_width = 120,
  xaxis_font_size = NULL, yaxis_font_size = NULL, brush_color = ”#0000FF”,
  show_grid = TRUE, anim_duration = 500, ...)
```

Arguments

`x` A numeric matrix Defaults to `TRUE` unless `x` contains any `NA`s.

`Rowv` determines if and how the row dendrogram should be reordered. By default, it is TRUE, which implies dendrogram is computed and reordered based on row means. If NULL or FALSE, then no dendrogram is computed and no reordering is done. If a dendrogram, then it is used "as-is", ie without any reordering. If a vector of integers, then dendrogram is computed and reordered based on the order of the vector.

`Colv` determines if and how the column dendrogram should be reordered. Has the options as the Rowv argument above and additionally when x is a square matrix, Colv = "Rowv" means that columns should be treated identically to the rows.

`distfun` function used to compute the distance (dissimilarity) between both rows and columns. Defaults to dist.

`hclustfun` function used to compute the hierarchical clustering when Rowv or Colv are not dendrograms. Defaults to hclust.

`dendrogram` character string indicating whether to draw ’none’, ’row’, ’column’ or ’both’ dendrograms. Defaults to ’both’. However, if Rowv (or Colv) is FALSE or NULL and dendrogram is ’both’, then a warning is issued and Rowv (or Colv) arguments are honoured.

`reorderfun` function(d, w) of dendrogram and weights for reordering the row and column dendrograms. The default uses statsreorder.dendrogram

`k_row` an integer scalar with the desired number of groups by which to color the dendrogram’s branches in the rows (uses color_branches) k_col an integer scalar with the desired number of groups by which to color the dendrogram's branches in the columns (uses color_branches)

symm logical indicating if x should be treated symmetrically; can only be true when x is a square matrix.

revC logical indicating if the column order should be reversed for plotting. Default (when missing) - is FALSE, unless symm is TRUE. This is useful for cor matrix.

scale character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is "none".

na.rm logical indicating whether NA's should be removed.

labRow character vectors with row labels to use (from top to bottom); default to rownames(x).

labCol character vectors with column labels to use (from left to right); default to colnames(x).

cexRow positive numbers. If not missing, it will override xaxis_font_size and will give it a value cexRow*14

cexCol positive numbers. If not missing, it will override yaxis_font_size and will give it a value cexCol*14

digits integer indicating the number of decimal places to be used by round for 'label'.

cellnote (optional) matrix of the same dimensions as x that has the human-readable version of each value, for displaying to the user on hover. If NULL, then x will be coerced using as.character. If missing, it will use x, after rounding it based on the digits parameter.

cellnote_scale logical (default is TRUE). IF cellnote is missing and x is used, should cellnote be scaled if x is also scaled?

theme A custom CSS theme to use. Currently the only valid values are "" and "dark". "dark" is primarily intended for standalone visualizations, not R Markdown or Shiny.

colors Either a colorbrewer2.org palette name (e.g. "YlOrRd" or "Blues"), or a vector of colors to interpolate in hexadecimal "#RRGGBB" format, or a color interpolation function like colorRamp.

width Width in pixels (optional, defaults to automatic sizing).

height Height in pixels (optional, defaults to automatic sizing).

xaxis_height Size of axes, in pixels.

yaxis_width Size of axes, in pixels.

xaxis_font_size Font size of axis labels, as a CSS size (e.g. "14px" or "12pt").

yaxis_font_size Font size of axis labels, as a CSS size (e.g. "14px" or "12pt").

brush_color The base color to be used for the brush. The brush will be filled with a low-opacity version of this color. "#RRGGBB" format expected.

show_grid TRUE to show gridlines, FALSE to hide them, or a numeric value to specify the gridline thickness in pixels (can be a non-integer).

anim_duration Number of milliseconds to animate zooming in and out. For large x it may help performance to set this value to 0.

... currently ignored

Source

The interface was designed based on heatmap, heatmap.2, and d3heatmap.

See Also

heatmap, heatmap.2, d3heatmap

Examples

```
library(heatmaply)
hm <- heatmapr(mtcars, scale = "column", colors = "Blues")
heatmaply(hm)
```

Index