Package ‘d3heatmap’

February 1, 2018

Type Package

Title Interactive Heat Maps Using 'htmlwidgets' and 'D3.js'

Version 0.6.1.2

Date 2016-02-23

Maintainer ORPHANED

Description Create interactive heat maps 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, drag a rectangle to zoom, and
click row/column labels to highlight.

License GPL-3 | file LICENSE

Imports scales (>= 0.2.5), htmlwidgets, png, base64enc, dendextend (>=
0.18.0), stats, grDevices

Suggests shiny, knitr, rmarkdown

VignetteBuilder knitr, rmarkdown

URL https://github.com/rstudio/d3heatmap

BugReports https://github.com/rstudio/d3heatmap/issues

Collate 'd3heatmap.R' 'dendrogram.R'

RoxygenNote 5.0.1

NeedsCompilation no

Author Joe Cheng [aut, cre],
Tal Galili [aut],
RStudio, Inc. [cph],
Michael Bostock [ctb, cph] (D3.js library),
Justin Palmer [ctb, cph] (d3.tip library)

Repository CRAN

Date/Publication 2018-02-01 15:19:34 UTC

X-CRAN-Original-Maintainer Joe Cheng <joe@rstudio.com>

X-CRAN-Comment Orphaned on 2018-02-01 as check errors were not
corrected despite reminders.
**R topics documented:**

- d3heatmap .................................................. 2
- d3heatmapOutput ......................................... 4

**Index**

<table>
<thead>
<tr>
<th>d3heatmap</th>
<th>D3 Heatmap widget</th>
</tr>
</thead>
</table>

**Description**

Creates a D3.js-based heatmap widget.

**Usage**

```r
d3heatmap(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 = TRUE, scale = c("none", "row", "column"), na.rm = TRUE,  
labRow = rownames(x), labCol = colnames(x), cexRow, cexCol, digits = 3L,  
cellnote, cellnote_scale = FALSE, 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 NAs.

- **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 FALSE). 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").
d3heatmapOutput

| 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 and heatmap.2

**See Also**

heatmap, heatmap.2

**Examples**

```r
library(d3heatmap)
d3heatmap(mtcars, scale = "column", colors = "Blues")
```

---

**Description**

Use d3heatmapOutput to create a UI element, and renderD3heatmap to render the heatmap.

**Usage**

```r
d3heatmapOutput(outputId, width = "100\%", height = "400px")
renderD3heatmap(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

- **outputId**: Output variable to read from
- **width**, **height**: The width and height of the map (see shinyWidgetOutput)
- **expr**: An expression that generates a d3heatmap object
- **env**: The environment in which to evaluate expr
- **quoted**: Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.
**Examples**

```r
library(d3heatmap)
library(shiny)

ui <- fluidPage(
  h1("A heatmap demo"),
  selectInput("palette", "Palette", c("YlOrRd", "RdYlBu", "Greens", "Blues")),
  checkboxInput("cluster", "Apply clustering"),
  d3heatmapOutput("heatmap")
)

server <- function(input, output, session) {
  output$heatmap <- renderD3heatmap({
    d3heatmap(
      scale(mtcars),
      colors = input$palette,
      dendrogram = if (input$cluster) "both" else "none"
    )
  })
}

shinyApp(ui, server)
```
Index

as.character, 3

color_branches, 3
colorRamp, 3
d3heatmap, 2, 4
d3heatmapOutput, 4

heatmap, 4
heatmap.2, 4

renderD3heatmap(d3heatmapOutput), 4
round, 3

shinyWidgetOutput, 4