

Package ‘sunburstR’

October 15, 2016

Type Package

Title 'Htmlwidget' for 'Kerry Rodden' 'd3.js' Sequence Sunburst

Version 0.6.3

Date 2016-10-14

Maintainer Kent Russell <kent.russell@timelyportfolio.com>

URL <https://github.com/timelyportfolio/sunburstR>

BugReports <https://github.com/timelyportfolio/sunburstR/issues>

Description Make interactive 'd3.js' sequence sunburst diagrams in R with the convenience and infrastructure of an 'htmlwidget'.

License MIT + file LICENSE

LazyData TRUE

Imports d3r, htmlwidgets, htmltools

Suggests jsonlite, knitr, markdown, pipeR

RoxygenNote 5.0.1

NeedsCompilation no

Author Mike Bostock [aut, cph] (d3.js library, <http://d3js.org>),
Kerry Rodden [aut, cph] (sequences library in htmlwidgets/lib,
<https://gist.github.com/kerryrodde/7090426>),
Kent Russell [aut, cre] (R interface),
Florian Breitwieser [ctb] (R interface)

Repository CRAN

Date/Publication 2016-10-15 18:52:15

R topics documented:

add_shiny	2
sunburst	3
sunburst-shiny	7

Index	9
--------------	----------

`add_shiny`*Add Shiny Events*

Description

Add Shiny Events

Usage

```
add_shiny(sunburst = NULL)
```

Arguments

`sunburst` `sunburst` htmlwidget to which you would like to add event handling

Value

`sunburst` htmlwidget

Examples

```
## Not run:

library(shiny)
library(sunburstR)

sequences <- read.csv(
  system.file("examples/visit-sequences.csv", package="sunburstR")
  ,header=F
  ,stringsAsFactors = FALSE
)

server <- function(input,output,session){

  output$sunburst <- renderSunburst({
    #invalidateLater(1000, session)

    sequences <- sequences[sample(nrow(sequences),1000),]

    add_shiny(sunburst(sequences))
  })

  selection <- reactive({
    input$sunburst_mouseover
  })

  output$selection <- renderText(selection())
}
```

```

ui<-fluidPage(
  sidebarLayout(
    sidebarPanel(
      ),
    # plot sunburst
    mainPanel(
      sunburstOutput("sunburst"),
      textOutput("selection")
    )
  )
)

shinyApp(ui = ui, server = server)

## End(Not run)

```

sunburst

'd3.js' Sequence Sunburst Diagrams

Description

Sequences sunburst diagrams provide an interactive method of exploring sequence data, such as website navigation paths.

Usage

```

sunburst(csvdata = NULL, jsondata = NULL, legendOrder = NULL,
  colors = NULL, percent = TRUE, count = FALSE, explanation = NULL,
  breadcrumb = list(), legend = list(), sortFunction = NULL,
  width = NULL, height = NULL, elementId = NULL, sizingPolicy = NULL)

```

Arguments

csvdata	data in csv source,target form
jsondata	data in nested d3 JSON hierarchy with 'name:..., children:[]';
legendOrder	string vector if you would like to manually order the legend. If legendOrder is not provided, then the legend will be in the descending order of the top level hierarchy.
colors	vector of strings representing colors as hexadecimal for manual colors. If you want precise control of colors, supply a list with range and/or domain.
percent	logical to include percentage of total in the explanation.
count	logical to include count and total in the explanation.

explanation	JavaScript function to define a custom explanation for the center of the sunburst. Note, this will override percent and count.
breadcrumb, legend	list to customize the breadcrumb trail or legend. This argument should be in the form <code>list(w =, h =, s =, t =)</code> where <code>w</code> is the width, <code>h</code> is the height, <code>s</code> is the spacing, and <code>t</code> is the tail all in px. <code>w</code> is <code>0</code> by default for breadcrumbs widths based on text length.
sortFunction	JS function to sort the slices. The default sort is by size.
height, width	height and width of sunburst htmlwidget containing div specified in any valid CSS size unit.
elementId	string id as a valid CSS element id.
sizingPolicy	see sizingPolicy .

Examples

```
# devtools::install_github("timelyportfolio/sunburstR")

library(sunburstR)

# read in sample visit-sequences.csv data provided in source
# https://gist.github.com/kerryrodden/7090426#file-visit-sequences-csv
sequences <- read.csv(
  system.file("examples/visit-sequences.csv", package="sunburstR")
  ,header = FALSE
  ,stringsAsFactors = FALSE
)

sunburst(sequences)

# explore some of the arguments
sunburst(
  sequences
  ,count = TRUE
)

## Not run:

sunburst(
  sequences
  # apply sort order to the legends
  ,legendOrder = unique(unlist(strsplit(sequences[,1], "-")))
  # just provide the name in the explanation in the center
  ,explanation = "function(d){return d.name}"
)

# try with json data
sequence_json <- jsonlite::fromJSON(
  system.file("examples/visit-sequences.json", package="sunburstR"),
  simplifyDataFrame = FALSE
)
```

```

)
sunburst(jsondata = sequence_json)

# try with csv data from this fork
# https://gist.github.com/mkajava/7515402
# great use for new breadbrumb wrapping
sunburst(
  csvdata = read.csv(
    file = paste0(
      "https://gist.githubusercontent.com/mkajava/",
      "7515402/raw/9f80d28094dc9dfed7090f8fb3376ef1539f4fd2/",
      "comment-sequences.csv"
    )
    ,header = FALSE
    ,stringsAsFactors = FALSE
  )
)

# try with csv data from this fork
# https://gist.github.com/rileycrane/92a2c36eb932b4f99e51/
sunburst( csvdata = read.csv(
  file = paste0(
    "https://gist.githubusercontent.com/rileycrane/",
    "92a2c36eb932b4f99e51/raw/",
    "a0212b4ca8043af47ec82369aa5f023530279aa3/visit-sequences.csv"
  )
  ,header=FALSE
  ,stringsAsFactors = FALSE
))

## End(Not run)
# use sunburst to analyze ngram data from Peter Norvig
# http://norvig.com/mayzner.html

library(sunburstR)
library(pipeR)

# read the csv data downloaded from the Google Fusion Table linked in the article
ngrams2 <- read.csv(
  system.file(
    "examples/ngrams2.csv"
    ,package="sunburstR"
  )
  , stringsAsFactors = FALSE
)

ngrams2 %>>%
  # let's look at ngrams at the start of a word, so columns 1 and 3
  (.[,c(1,3)]) %>>%
  # split the ngrams into a sequence by splitting each letter and adding -

```

```

(
  data.frame(
    sequence = strsplit(.,1,"") %>>%
      lapply( function(ng){ paste0(ng,collapse = "-") } ) %>>%
      unlist
    ,freq = .[,2]
    ,stringsAsFactors = FALSE
  )
) %>>%
sunburst

## Not run:

library(htmltools)

ngrams2 %>>%
(
  lapply(
    seq.int(3,ncol(.))
    ,function(letpos){
      (.,c(1,letpos)) %>>%
      # split the ngrams into a sequence by splitting each letter and adding -
      (
        data.frame(
          sequence = strsplit(.,1,"") %>>%
            lapply( function(ng){ paste0(ng,collapse = "-") } ) %>>%
            unlist
          ,freq = .[,2]
          ,stringsAsFactors = FALSE
        )
      ) %>>%
      ( tags$div(style="float:left;",sunburst( ., height = 300, width = 300 )) )
    }
  )
) %>>%
tagList %>>%
browsable

## End(Not run)
# calendar sunburst example

library(sunburstR)

df <- data.frame(
  date = seq.Date(
    as.Date('2014-01-01'),
    as.Date('2016-12-31'),
    by = "days"
  ),
  stringsAsFactors = FALSE
)

df$year = format(df$date, "%Y")

```

```
df$quarter = paste0("Q", ceiling(as.numeric(format(df$date,"%m"))/3))
df$month = format(df$date, "%b")
df$path = paste(df$year, df$quarter, df$month, sep="-")
df$count = rep(1, nrow(df))

sunburst(
  data.frame(xtabs(count~path,df)),
  # added a degree of difficulty by providing
  # not easily sortable names
  sortFunction = htmlwidgets::JS(
"
function(a,b){
  abb = {
    2014:-7,
    2015:-6,
    2016:-5,
    Q1:-4,
    Q2:-3,
    Q3:-2,
    Q4:-1,
    Jan:1,
    Feb:2,
    Mar:3,
    Apr:4,
    May:5,
    Jun:6,
    Jul:7,
    Aug:8,
    Sep:9,
    Oct:10,
    Nov:11,
    Dec:12
  }
  return abb[a.name] - abb[b.name];
}
"
)
)
```

sunburst-shiny

Shiny bindings for sunburst

Description

Output and render functions for using sunburst within Shiny applications and interactive Rmd documents.

Usage

```
sunburstOutput(outputId, width = "100%", height = "400px")
```

```
renderSunburst(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a sunburst
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.

Index

`add_shiny`, [2](#)

JS, [4](#)

`renderSunburst` (`sunburst-shiny`), [7](#)

`sizingPolicy`, [4](#)

`sunburst`, [3](#)

`sunburst-shiny`, [7](#)

`sunburstOutput` (`sunburst-shiny`), [7](#)