Package ‘vistime’

January 10, 2020

Title Pretty Timeline Creation
Version 0.9.0
Date 2020-01-07
Author Sandro Raabe [aut, cre]
Maintainer Sandro Raabe <sa.ra.online@posteo.de>
Description Create interactive timelines or Gantt charts that are usable in the 'RStudio' viewer pane, in 'R Markdown' documents and in 'Shiny' apps. Hover the mouse pointer over a point or task to show details or drag a rectangle to zoom in. Timelines and their components can afterwards be manipulated using plotly_build(), which transforms the plot into a mutable list.
License GPL-3 | file LICENSE
URL https://shosaco.github.io/vistime/
BugReports https://github.com/shosaco/vistime/issues
Depends R (>= 3.2.0)
Imports assertive (>= 0.1.4), plotly (>= 4.0.0), RColorBrewer (>= 0.2.2)
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Suggests ggplot2, knitr, rmarkdown, devtools, testthat, covr, purrr, dplyr
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2020-01-10 00:40:02 UTC

R topics documented:
vistime ................................................................. 2

Index 5
**vistime**

*Create a Timeline*

**Description**

Visualize interactive timelines offline.

Provide a data frame with event data to create a visual timeline plot. Simplest drawable dataframe can have columns `event` and `start`.

**Usage**

```r
vistime(data, events = "event", start = "start", end = "end",
groups = "group", colors = "color", fontcolors = "fontcolor",
tooltips = "tooltip", optimize_y = TRUE, linewidth = NULL,
title = NULL, showLabels = NULL, show_labels = TRUE,
lineInterval = NULL, background_lines = 10)
```

**Arguments**

- **data** (required) data.frame that contains the data to be visualised
- **events** (optional) the column name in data that contains event names. Default: `event`.
- **start** (optional, character) the column name in data that contains start dates. Default: `start`.
- **end** (optional, character) the column name in data that contains end dates. Default: `end`.
- **groups** (optional, character) the column name in data to be used for grouping. Default: `group`.
- **colors** (optional, character) the column name in data that contains colors for events. Default: `color`, if not present, colors are chosen via `RColorBrewer`.
- **fontcolors** (optional, character) the column name in data that contains the font color for event labels. Default: `fontcolor`, if not present, color will be black.
- **tooltips** (optional, character) the column name in data that contains the mouseover tooltips for the events. Default: `tooltip`, if not present, then tooltips are build from event name and date.
- **optimize_y** (optional, logical) distribute events on y-axis by smart heuristic (default), otherwise use order of input data.
- **linewidth** (optional, numeric) the linewidth (in pixel) for the events (typically used for large amount of parallel events). Default: heuristic value.
- **title** (optional, character) the title to be shown on top of the timeline. Default: `NULL`.
- **showLabels** deprecated and replaced by argument `show_labels`.
- **show_labels** (optional, boolean) choose whether or not event labels shall be visible. Default: `TRUE`.
- **lineInterval** deprecated, use argument `background_lines` instead.
background_lines
(optional, integer) the number of vertical lines to draw in the background to
demonstrate structure (default: 10). Less means more memory-efficient plot.

Value

vistime returns an object of class plotly and htmlwidget.

Author(s)

Sandro Raabe <shosaco_nospam@hotmail.com>

Examples

# presidents and vice presidents
pres <- data.frame(
  Position = rep(c("President", "Vice"), each = 3),
  Name = c("Washington", rep(c("Adams", "Jefferson"), 2), "Burr"),
  start = c("1789-03-29", "1797-02-03", "1801-02-03"),
  end = c("1797-02-03", "1801-02-03", "1809-02-03"),
  color = c("#cbb69d", "#603913", "#c69c6e"),
  fontcolor = c("black", "white", "black"))

vistime(pres, events = "Position", groups = "Name", title = "Presidents of the USA")

## Not run:
# more complex and colorful example
data <- read.csv(text = "event,group,start,end,color
Phrase 1,Project,2018-12-22,2018-12-23,#c8e6c9
Phrase 2,Project,2018-12-23,2018-12-29,#a5d6a7
Phrase 3,Project,2018-12-29,2019-01-06,#fb8c00
Phrase 4,Project,2019-01-06,2019-02-02,#D4B93F
Room 334,Team 1,2018-12-22,2018-12-28,#DEEBF7
Room 335,Team 1,2018-12-28,2019-01-05,#C6DBEF
Room 335,Team 1,2019-01-05,2019-01-23,#9ECAE1
Group 1,Team 2,2018-12-22,2018-12-28,#E5F5E0
Group 2,Team 2,2018-12-28,2019-01-23,#C7E9C0
3-200,c category 1,2018-12-25,2018-12-25,#1565c0
3-330,c category 1,2018-12-25,2018-12-25,#1565c0
3-223,c category 1,2018-12-28,2018-12-28,#1565c0
3-225,c category 1,2018-12-28,2018-12-28,#1565c0
3-226,c category 1,2018-12-28,2018-12-28,#1565c0
3-226,category 1,2019-01-19,2019-01-19,#1565c0
3-330,category 1,2019-01-19,2019-01-19,#1565c0
1-217.0,category 2,2018-12-27,2018-12-27,#90ca9f
3-399.7,moon rising,2019-01-13,2019-01-13,#F4336
8-831.0,sundowner drink,2019-01-17,2019-01-17,#8d6e63
9-984.1,birthday party,2018-12-22,2018-12-22,#90a4ae
F01.9,Meetings,2018-12-26,2018-12-26,#E8A735
Z71,Meetings,2019-01-12,2019-01-12,#E8A735

vistime(pres, events = "Position", groups = "Name", title = "Presidents of the USA")

## Not run:
# more complex and colorful example
data <- read.csv(text = "event,group,start,end,color
Phrase 1,Project,2018-12-22,2018-12-23,#c8e6c9
Phrase 2,Project,2018-12-23,2018-12-29,#a5d6a7
Phrase 3,Project,2018-12-29,2019-01-06,#fb8c00
Phrase 4,Project,2019-01-06,2019-02-02,#D4B93F
Room 334,Team 1,2018-12-22,2018-12-28,#DEEBF7
Room 335,Team 1,2018-12-28,2019-01-05,#C6DBEF
Room 335,Team 1,2019-01-05,2019-01-23,#9ECAE1
Group 1,Team 2,2018-12-22,2018-12-28,#E5F5E0
Group 2,Team 2,2018-12-28,2019-01-23,#C7E9C0
3-200,c category 1,2018-12-25,2018-12-25,#1565c0
3-330,c category 1,2018-12-25,2018-12-25,#1565c0
3-223,c category 1,2018-12-28,2018-12-28,#1565c0
3-225,c category 1,2018-12-28,2018-12-28,#1565c0
3-226,c category 1,2018-12-28,2018-12-28,#1565c0
3-226,category 1,2019-01-19,2019-01-19,#1565c0
3-330,category 1,2019-01-19,2019-01-19,#1565c0
1-217.0,category 2,2018-12-27,2018-12-27,#90ca9f
3-399.7,moon rising,2019-01-13,2019-01-13,#F4336
8-831.0,sundowner drink,2019-01-17,2019-01-17,#8d6e63
9-984.1,birthday party,2018-12-22,2018-12-22,#90a4ae
F01.9,Meetings,2018-12-26,2018-12-26,#E8A735
Z71,Meetings,2019-01-12,2019-01-12,#E8A735

vistime(pres, events = "Position", groups = "Name", title = "Presidents of the USA")

## Not run:
vistime(data)

# ------ It is possible to change all attributes of the timeline using plotly_build(), # ------ which generates a list which can be inspected using str
p <- vistime(data.frame(event = 1:4, start = c("2019-01-01", "2019-01-10")))
pp <- plotly_build(p) # transform into a list

# Example 1: change x axis font size:
pp$x$layout$xaxis$tickfont <- list(size = 28)
pp

# Example 2: change y axis font size (several y-axes, therefore we need a loop):
for (i in grep("yaxis\*", names(pp$x$layout))) pp$x$layout[[i]]$tickfont <- list(size = 28)
pp

# Example 3: Changing events font size
for (i in 1:length(pp$x$data)) {
  if (pp$x$data[[i]]$mode == "text") pp$x$data[[i]]$textfont$size <- 28
}
pp

# or, using purrr:
text_idx <- which(purrr::map_chr(pp$x$data, "mode") == "text")
for(i in text_idx) pp$x$data[[i]]$textfont$size <- 28
pp

# Example 4: change marker size
# loop over pp$x$data, and change the marker size of all text elements to 50px
for (i in 1:length(pp$x$data)) {
  if (pp$x$data[[i]]$mode == "markers") pp$x$data[[i]]$marker$size <- 40
}
pp

# or, using purrr:
marker_idx <- which(purrr::map_chr(pp$x$data, "mode") == "markers")
for(i in marker_idx) pp$x$data[[i]]$marker$size <- 40
pp

## End(Not run)
Index

*Topic `gantt`
  vistime, 2
*Topic `plotly`
  vistime, 2
*Topic `timeline`
  vistime, 2
*Topic `vistime`
  vistime, 2

vistime, 2
vistime-package (vistime), 2