Package 'rPackedBar'

October 14, 2022

Type Package

Title Packed Bar Charts with 'plotly'

Version 0.2.2

Author Adam Spannbauer [aut, cre]

Maintainer Adam Spannbauer <spannbaueradam@gmail.com>

Description Packed bar charts are a variation of treemaps for visualizing skewed data. The concept was introduced by Xan Gregg at 'JMP'.

License MIT + file LICENSE

URL https://github.com/AdamSpannbauer/rPackedBar

BugReports https://github.com/AdamSpannbauer/rPackedBar/issues

Imports plotly, data.table, scales, shiny

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

RoxygenNote 6.1.1

Suggests covr, testthat, R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-17 09:00:03 UTC

R topics documented:

---

packed_bar_shiny   *Plot rPackedBar in 'shiny'*

Description

Output and render functions for using 'rPackedBar' within 'shiny' applications and interactive Rmd documents.

Usage

```
packedBarOutput(outputId, width = "100%", height = "400px",
  inline = FALSE, clickedBarInputId = paste0(outputId, "_clicked"))

renderPackedBar(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.
inline	use an inline (span()) or block container (div()) for the output
clickedBarInputId	
	The input id to assign the label of the clicked bar. Defaults to outputId_clicked and can be referenced in the server as input$outputId_clicked.
expr	An expression that generates an rPackedBar.
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.

---

plotly_packed_bar   *Create a 'plotly' packed bar chart*

Description

Packed bar charts are a variation of treemaps for visualizing skewed data. The concept was introduced by XanGregg at JMP (https://community.jmp.com/t5/JMP-Blog/Introducing-packed-bars-a-new-chart-fo
ba-p/39972).

Usage

```
plotly_packed_bar(input_data, label_column, value_column,
  number_rows = "guess", plot_title = "", xaxis_label = "",
  hover_label = "", min_label_width = 0.03,
  color_bar_color = "steelblue", label_color = "black")
```

Arguments

`input_data`	data.frame with data to plot, should have a column of labels for bars and column of numbers relating to bar length
`label_column`	either the column number or quoted name in `input_data` to be used as labels (will be used by `[[` to subset)
`value_column`	either the column number or quoted name in `input_data` to be used as numbers for bar lengths (will be used by `[[` to subset). See details.
`number_rows`	The number of rows to occur in barchart (i.e. the number of colored bars). The default of 'guess' will attempt to find the 'elbow' in the numeric data with a min value of 3 and a max of 25.
`plot_title`	main title for plot
`xaxis_label`	label to put on xaxis
`hover_label`	text to appear by number in hover information (typically same as xaxis label)
`min_label_width`	
	min relative length of bar to receive a static label (too small and text will overflow sides of bar)
`color_bar_color`	
	color of main bars in chart (can be name or hex) remaining bars will be variations of gray
`label_color`	color of text labels that appear over colored bars

Details

The packed barchart currently only works for uniformly positive or uniformly negative data; negative values will be removed before plotting when mixed data is provided.

Value

'plotly' object of the packed bar chart

Examples

```
## Not run:
data(GNI2014, package = 'treemap')

data.table::setDT(GNI2014)
my_input_data = GNI2014[,sum(population), by=country]

plotly_packed_bar(my_input_data,
                  label_column    = "country",
                  value_column    = "V1",
                  plot_title      = "Population 2014",
                  xaxis_label     = "Population",
                  hover_label     = "Population",
                  min_label_width = .025,
                  color_bar_color = "orange")

## End(Not run)
```

Index