Package: voronoiTreemap (via r-universe)

September 13, 2024

Type Package

Title Voronoi Treemaps with Added Interactivity by Shiny

Version 0.2.1

Description The d3.js framework with the plugins d3-voronoi-map, d3-voronoi-treemap and d3-weighted-voronoi are used to generate Voronoi treemaps in R and in a shiny application. The computation of the Voronoi treemaps are based on Nocaj and Brandes (2012) <doi:10.1111/j.1467-8659.2012.03078.x>.

URL https://github.com/uRosConf/voronoiTreemap

License GPL-3

Imports data.tree,rlang,htmlwidgets,shiny,shinyjs,DT

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests rmarkdown, scales, testthat

Repository https://urosconf.r-universe.dev

RemoteUrl https://github.com/urosconf/voronoitreemap

RemoteRef HEAD

RemoteSha 69708642a2d0848dc803e2cc0ec837ea6ff594db

Contents

---

canada *canada*

Description

An example data.frame using Canadian Consumer Price Index (CPI) to demonstrate the voronoiTree package

Usage

```
data(canada)
```

Format

A data frame with 247 rows and 5 variables:

h1 Name of first-level (region)

h2 Leaf names of second-level (elementary_aggregate)

h3 Leaf names of third-level values (intermediate_aggregate)

color colors in which the plot-regions will be filled

weight CPI in percent of the overall total

codes NAs

Examples

```
data("canada")
head(canada)
```

ExampleGDP *ExampleGDP*

Description

An example data.frame using GDP data to demonstrate the voronoiTree package

Usage

```
data(ExampleGDP)
```

Format

A data frame with 42 rows and 6 variables:

h1 Name of first-level (redundant)

h2 Leaf names of second-level (continents)

h3 Leaf names of third-level values (countries)

color colors in which the plot-regions will be filled

weight GDP values in percent of the overall total

codes short labels used for overlays in plotting

Examples

```
data("ExampleGDP")
head(ExampleGDP)
```

`vt_add_nodes` *vt_add_nodes*

Description

add (sub)nodes to a node generated by `vt_create_node` or returned from `vt_add_nodes`

Usage

```
vt_add_nodes(node, refnode, node_names, colors = NULL, weights = NULL,
  codes = NULL)
```

Arguments

`node`	a node object
`refnode`	name of the reference node
`node_names`	new node names
`colors`	optionally a vector of colors matching the length of `node_names`
`weights`	optionally a vector of weights matching the length of `node_names`
`codes`	optionally a vector of short labels matching the length of `node_names`

Value

a Node object

Examples

```
n <- vt_create_node("Total")
n <- vt_add_nodes(n, refnode="Total",node_names=c("Asia","Europe"), colors=c("red","blue"))
n <- vt_add_nodes(n, refnode="Asia",node_names=c("China","Thailand"),
  weights=c(0.5, 0.8), codes=c("CN","TH"))
n <- vt_add_nodes(n, refnode="Europe",node_names=c("Netherlands","Austria"),
  weights=c(0.9, 1.1), codes=c("NL","AT"))
print(n, "weight", "code", "color")
```

---

vt_app *vt_app*

Description

starts the graphical user interface developed with *shiny*.

Usage

```
vt_app(maxRequestSize = 50, ...)
```

Arguments

maxRequestSize	(numeric) number defining the maximum allowed filesize (in megabytes) for uploaded files, defaults to 50MB
...	arguments (e.g host) that are passed through vt_app when starting the shiny application

Value

starts the interactive graphical user interface which may be used to perform the anonymisation process.

Examples

```
## Not run:
vt_app()

## End(Not run)
```

vt_create_node *vt_create_node*

Description

vt_create_node

Usage

```
vt_create_node(total_lab = "Total")
```

Arguments

total_lab	name of the total level

Value

a Node

Examples

```
vt_create_node("Total")
```

vt_d3 *Voronoi Treemap in an htmlwidget*

Description

Function to generate an htmlwidget with a voronoi treemap

Usage

```
vt_d3(data, elementId = NULL, width = NULL, height = NULL,
  seed = NULL, title = NULL, legend = FALSE, legend_title = NULL,
  footer = NULL, label = TRUE, color_circle = "#aaaaaa",
  color_border = "#ffffff", color_label = "#000000",
  size_border = "1px", size_border_hover = "3px",
  size_circle = "2px")
```

Arguments

data	a correct json data object
elementId	optional a custom elementId to be returned
width	width of the widget
height	height of the widget
seed	if defined, the plot is fixed
title	NULL or a string for the title
legend	TRUE/FALSE if a legend should be printed
legend_title	NULL or a string for the title of the legend
footer	NULL or a string for the footer text
label	TRUE/FALSE if the labels should be printed
color_circle	color for the outer circle
color_border	color for the inner lines
color_label	color for the label in the plot
size_border	thickness of the borders in css style, e.g. '1px'
size_border_hover	thickness of the borders when hovering in css style, e.g. '3px'
size_circle	thickness of the circle in css style, e.g. '2px'

Note

The JavaScript library d3-voronoi treemap can be found here https://github.com/Kcnarf/d3-voronoi-treemap and the example is based on the remake of HowMuch.net's article 'The Global Economy by GDP' by _Kcnarf https://bl.ocks.org/Kcnarf/fa95aa7b076f537c00aed614c29bb568.

References

Arlind Nocaj and Ulrik Brandes. (2012). Computing Voronoi Treempas: Faster, Simpler and Resolution-independent. Computer Graphics Forum. Vol.31. 855-864.

Examples

```
vt_d3(vt_export_json(vt_testdata()))
data(ExampleGDP)
gdp_json <- vt_export_json(vt_input_from_df(ExampleGDP, hierachyVar0 = "h1",
hierachyVar1 = "h2", hierachyVar2 = "h3", colorVar = "color", weightVar="weight",
labelVar = "codes"))
vt_d3(gdp_json)
data(canada)
canada$codes <- canada$h3
canada <- canada[canada$h1=="Canada",]
canadaH <- vt_input_from_df(canada,scaleToPerc = FALSE, hierachyVar0 = "h1",
  hierachyVar1 = "h2", hierachyVar2 = "h3", colorVar = "color", labelVar = "codes",
  weightVar = "weight")
vt_d3(vt_export_json(canadaH))
```

```
#without label
vt_d3(vt_export_json(canadaH), label=FALSE)
#Example with coloring from scales package
library(scales)
canada$color <- seq_gradient_pal()(exp(canada$weight)/500)
canadaH <- vt_input_from_df(canada,scaleToPerc = FALSE, hierachyVar0 = "h1",
hierachyVar1 = "h2", hierachyVar2 = "h3", colorVar = "color", labelVar = "codes",
  weightVar = "weight")
vt_d3(vt_export_json(canadaH))
```

`vt_d3-shiny` *Shiny bindings for d3vt*

Description

Output and render functions for using d3vt within Shiny applications and interactive Rmd documents.

Usage

```
vt_d3_output(outputId, width = "100%", height = "400px")

render_vt_d3(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 d3vt
`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.

`vt_export_json` *vt_export_json*

Description

exports a node to suitable json required by voronoi javascript function

Usage

```
vt_export_json(node, file = NULL)
```

Arguments

node	a Node object generated by vt_create_node or returned from vt_add_nodes
file	path where the json should be written to, if NULL, the json is returned as a character

Value

NULL or a character vector

See Also

vt_create_node vt_add_nodes

Examples

```
n <- vt_testdata()
vt_export_json(n)
vt_export_json(n, file=tempfile())
```

---

vt_input_from_df *vt_input_from_df*

Description

create a tree-structure from a data.frame

Usage

```
vt_input_from_df(inp, scaleToPerc = FALSE, hierachyVar0 = NULL,
  hierachyVar1, hierachyVar2, colorVar, weightVar, labelVar = NULL)
```

Arguments

inp	a data.frame with specific format
scaleToPerc	(logical) scale to percent
hierachyVar0	(character) variable name of toplevel hierachy (will be replaced with "Total" if empty)
hierachyVar1	(character) variable name of first level
hierachyVar2	(character) variable name of second level
colorVar	(character) variable name of the color information
weightVar	(character) variable name of the weight information
labelVar	(character) variable name of (short) labels

Value

a Node that can be written to json using vt_export_json

Examples

```
data(canada)
canada <- canada[canada$h1=="Canada",]
canada$codes <- canada$h3
v1 <- vt_input_from_df(canada, hierachyVar0 = "h1", hierachyVar1 = "h2",
  hierachyVar2 = "h3", colorVar = "color",weightVar = "weight",
  labelVar = "codes")
 colnames(canada) <- c("hier1","hier2","hier3","col","w","cod")
v2 <- vt_input_from_df(canada, hierachyVar0 = "hier1", hierachyVar1 = "hier2",
    hierachyVar2 = "hier3", colorVar = "col",weightVar = "w",
    labelVar = "cod")
```

`vt_testdata` *vt_testdata*

Description

vt_testdata

Usage

```
vt_testdata()
```

Value

returns a json-string as in the example from https://bl.ocks.org/Kcnarf/fa95aa7b076f537c00aed614c29bb568

Examples

```
vt_testdata()
```

Index