Package 'd3po'

June 12, 2020

Title D3 Prototype Options

Version 0.0.1

Description A collection of scripts to create common d3 diagrams using r2d3.

Maintainer Taylor McKenzie <tkmckenzie@gmail.com>

Depends R (>= 4.0.0)

Imports r2d3 (>= 0.2.3), webshot (>= 0.5.2)

Suggests ggplot2 (>= 3.1.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Taylor McKenzie [aut, cre]

R topics documented:

`chord` *Chord diagram*

Description

Creates chord diagram from edge data.frame.

Usage

```
chord(
  df,
  source.column = "source",
  target.column = "target",
  value.column = "value",
  edge.color = c("path", "input", "output", "none"),
  width = NULL,
  height = NULL,
  viewer = c("internal", "external", "browser")
)
```

Arguments

df	data.frame containing edge data
source.column	Name of column containing source nodes. Defaults to "source".
target.column	Name of column containing target nodes. Defaults to "target".
value.column	Name of column containing edge values. Defaults to "value".
edge.color	Method of coloring edges. The value "path" will create a gradient between two nodes. Defaults to "path".
width	Desired width for output widget.
height	Desired height for output widget.
viewer	"internal" to use the RStudio internal viewer pane for output; "external" to display in an external RStudio window; "browser" to display in an external browser.

Details

Utilizes a script similar to https://observablehq.com/@d3/chord-diagram adapted to work with r2d3.

Value

A d3 object as returned by r2d3::r2d3.

Examples

```
labels = c("spam", "eggs", "foo", "bar")

df = data.frame(source = rep(labels, each = 4),
                target = rep(labels, times = 4),
                value = c(11975, 5871, 8916, 2868,
                          1951, 10048, 2060, 6171,
                          8010, 16145, 8090, 8045,
                          1013, 990, 940, 6907))

chord(df)
```

cloud *Word cloud diagram*

Description

Creates word cloud diagram text and value data.frame.

Usage

```
cloud(
  df,
  text.column = "text",
  value.column = "value",
  group.column = "group",
  text.color = c("group", "word", "none"),
  width = NULL,
  height = NULL,
  viewer = c("internal", "external", "browser")
)
```

Arguments

df	data.frame containing text, value, and group data.
text.column	Name of column containing text. Defaults to "text".
value.column	Name of column containing edge values. Defaults to "value".
group.column	Name of column containing group data. Defaults to "group". If group.column is not found in df, a new column with a single group will be created.
text.color	How to color text; "group" (default) colors by group, "word" colors by word, and "none" colors all words black.
width	Desired width for output widget.
height	Desired height for output widget.
viewer	"internal" to use the RStudio internal viewer pane for output; "external" to display in an external RStudio window; "browser" to display in an external browser.

Details

Utilizes a script similar to https://observablehq.com/@d3/word-cloud adapted to work with r2d3.

Value

A d3 object as returned by r2d3::r2d3.

Examples

```
df = data.frame(text = c("foo", "bar", "spam", "eggs"),
                value = c(0.5, 10, 1, 10),
                group = c("Not Python", "Not Python", "Python", "Python"))

cloud(df)
```

`df.to.adjacency` *Conversion from data.frame to adjacency matrix*

Description

Creates an adjacency matrix from an edge list data.frame.

Usage

```
df.to.adjacency(
  df,
  source.column = "source",
  target.column = "target",
  value.column = "value"
)
```

Arguments

`df`	data.frame containing edge data
`source.column`	Name of column containing source nodes. Defaults to "source".
`target.column`	Name of column containing target nodes. Defaults to "target".
`value.column`	Name of column containing edge values. Defaults to "value".

Details

Utilizes a script similar to https://observablehq.com/@d3/chord-diagram adapted to work with r2d3.

Value

A list with two components:

`matrix`	Adjacency matrix
`labels`	Names of nodes, in same order as rows/columns of adjacency matrix

Examples

```
labels = c("spam", "eggs", "foo", "bar")

df = data.frame(source = rep(labels, each = 4),
                target = rep(labels, times = 4),
                value = c(11975, 5871, 8916, 2868,
                          1951, 10048, 2060, 6171,
                          8010, 16145, 8090, 8045,
                          1013, 990, 940, 6907))

df.to.adjacency(df)
```

energy *Energy sources and sinks*

Description

Dataset describing energy generation and consumption as a directed network. Data come from the Department of Energy & Climate Change via Tom Counsell. See http://www.decc.gov.uk/en/content/cms/tackling/2050/calculator_on/calculator_on.aspx.

Usage

```
energy
```

Format

An object of class `data.frame` with 68 rows and 3 columns.

marimekko *Marimekko diagram*

Description

Creates Marimekko diagram from data.frame.

Usage

```
marimekko(
  df,
  x.column = "x",
  y.column = "y",
  value.column = "value",
  width = NULL,
  height = NULL,
  viewer = c("internal", "external", "browser")
)
```

Arguments

`df`	data.frame containing horizontal category, vertical category, and value data.
`x.column`	Name of column containing horizontal category data. Defaults to "x".
`y.column`	Name of column containing vertical category data. Defaults to "y".
`value.column`	Name of column containing value data. Defaults to "value".
`width`	Desired width for output widget.
`height`	Desired height for output widget.
`viewer`	"internal" to use the RStudio internal viewer pane for output; "external" to display in an external RStudio window; "browser" to display in an external browser.

Details

Utilizes a script similar to https://observablehq.com/@d3/marimekko-chart adapted to work with r2d3.

Value

A d3 object as returned by r2d3::r2d3.

Examples

```
data(sales)

marimekko(sales, x.column = "market", y.column = "segment")
```

---

sales *Synthetic sales data*

Description

Fictitious dataset describing sales of various products in various locales. Taken from https://observablehq.com/@d3/marimekko-chart.

Usage

```
sales
```

Format

An object of class `data.frame` with 16 rows and 3 columns.

---

sankey *Sankey diagram*

Description

Creates Sankey diagram from edge data.frame.

Usage

```
sankey(
  df,
  source.column = "source",
  target.column = "target",
  value.column = "value",
  align = c("justify", "left", "right", "center"),
  edge.color = c("path", "input", "output", "none"),
  width = NULL,
  height = NULL,
  viewer = c("internal", "external", "browser")
)
```

Arguments

df	data.frame containing edge data
source.column	Name of column containing source nodes. Defaults to "source".
target.column	Name of column containing target nodes. Defaults to "target".
value.column	Name of column containing edge values. Defaults to "value".
align	Alignment of node labels. Defaults to "justify".
edge.color	Method of coloring edges. The value "path" will create a gradient between two nodes. Defaults to "path".
width	Desired width for output widget.
height	Desired height for output widget.
viewer	"internal" to use the RStudio internal viewer pane for output; "external" to display in an external RStudio window; "browser" to display in an external browser.

Details

Utilizes a script similar to https://observablehq.com/@d3/sankey-diagram adapted to work with r2d3.

Value

A d3 object as returned by r2d3::r2d3.

Examples

```
data(energy)

sankey(energy)
```

---

save.d3 *Save d3 diagram as png*

Description

Saves d3 diagram as a png image using webshot.

Usage

```
save.d3(
  d3,
  file,
  width = 1000,
  height = 750,
  delay = 0.2,
  zoom = 1,
  background = "white",
  title = "D3 Visualization"
)
```

Arguments

d3	A d3 object.
file	Location to save image.
width	Width of image.
height	Height of image.
delay	Time to wait before taking screenshot, in seconds. Sometimes a longer delay is needed for all assets to display properly.
zoom	Zoom before screenshot.
background	Background color of diagram.
title	Title for HTML diagram.

Examples

```
data(energy)

d3 = sankey(energy)
f = tempfile()
save.d3(d3, f)
```

Index