Package: network.r2d3 (via r-universe)

September 8, 2024

Title Makes interactive network graphs using r2d3

Version 0.0.0.9000

Description Makes interactive network graphs using r2d3.

URL https://cjyetman.github.io/network.r2d3

BugReports https://github.com/cjyetman/network.r2d3/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.0

Imports jsonlite (>= 0.9.6), r2d3

Suggests ape, chromote, codemetar, covr, cffr, data.tree, devtools, diffviewer, igraph, pak, pkgdown, preferably, shiny, testthat (>= 3.0.0), tibble, tidygraph

Config/testthat/edition 3

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

RemoteUrl https://github.com/cjyetman/network.r2d3

RemoteRef HEAD

RemoteSha 2fdcc8f58c023917fe39bc68649524abe3433a07

Contents

`as_force_data` *Convert one of numerous data types to force_network's 'native' data format*

Description

The `force_network` function uses a 'native' data format that consists of a...

Usage

```
as_force_data(.data, ...)

## S3 method for class 'character'
as_force_data(.data, ...)

## S3 method for class 'data.frame'
as_force_data(.data, ...)

## S3 method for class 'igraph'
as_force_data(.data, ...)

## S3 method for class 'hclust'
as_force_data(.data, ...)

## S3 method for class 'dendrogram'
as_force_data(.data, ...)

## S3 method for class 'list'
as_force_data(.data, ...)
```

Arguments

`.data`	a force network description in one of numerous forms (see details).
`...`	other arguments that will be passed on to as_force_data

Methods (by class)

- `as_force_data(character)`: Convert data found at a URL to an appropriate network data list
- `as_force_data(data.frame)`: Convert a data frame containing links data to an appropriate network data list

- `as_force_data(igraph)`: Convert an igraph object to an appropriate network data list
- `as_force_data(hclust)`: Convert a hclust object to an appropriate network data list
- `as_force_data(dendrogram)`: Convert a dendrogram object to an appropriate network data list
- `as_force_data(list)`: Convert a list object containg a links and a nodes data frame to an appropriate network data list

---

`as_sankey_data` *Convert one of numerous data types to sankey_network's 'native' data format*

Description

The `sankey_network` function uses a 'native' data format that consists of a...

Usage

```
as_sankey_data(.data, ...)
```

Arguments

`.data`	a sankey network description in one of numerous forms (see details).
`...`	other arguments that will be passed on to as_sankey_data

`as_tree_data` *Convert one of numerous data types to tree_network's 'native' treenetdf form*

Description

The `tree_network` function uses a 'native' data format that consists of a data frame with minimally 2 vectors/columns, one named `'nodeId'` and one named `'parentId'`. Other columns in the data frame are also passed on to the JavaScript code and attached to the elements in the D3 visualization so that they can potentially be accessed by other JavaScript functions. This is an advantageous format because:

- it's an easy to use and understand R-like format
- a hierarchical network can be succinctly defined by a list of each unique node and its parent node
- since each row defines a unique node, additional columns can be added to add node-specific properties
- in a hierarchical network, every link/edge can be uniquely identified by the node which it leads to, therefore each link/edge can also be specifically addressed by adding columns for formatting of the incoming link

as_tree_data can convert from any of the following data types:

- leafpathdf (table)–parent|parent|node–data.frame
- hierarchical nested list (JSON)
- hclust
- data.tree Node
- igraph
- ape phylo

Usage

```
as_tree_data(data, ...)

## S3 method for class 'character'
as_tree_data(data, ...)

## S3 method for class 'hclust'
as_tree_data(data, ...)

## S3 method for class 'list'
as_tree_data(data, children_name = "children", node_name = "name", ...)

## S3 method for class 'Node'
as_tree_data(data, ...)

## S3 method for class 'phylo'
as_tree_data(data, ...)

## S3 method for class 'tbl_graph'
as_tree_data(data, ...)

## S3 method for class 'igraph'
as_tree_data(data, root = "root", ...)

## S3 method for class 'data.frame'
as_tree_data(
  data,
  cols = NULL,
  df_type = "treenetdf",
  subset = names(data),
  root,
  ...
)
```

Arguments

data	a tree network description in one of numerous forms (see details).
...	other arguments that will be passed on to as_tree_data

children_name	character specifying the name used for the list element that contains the childeren elements.
node_name	character specifying the name used for the list element that contains the name of the node
root	root name.
cols	named character vector specifying the names of columns to be converted to the standard treenetdf names.
df_type	character specifying which type of data frame to convert. Can be treenetdf or leafpathdf.
subset	character vector specifying the names of the columns (in order) that should be used to define the hierarchy.

Methods (by class)

- as_tree_data(character): Convert JSON from URL to treenetdf
- as_tree_data(hclust): Convert hclust objects to treenetdf
- as_tree_data(list): Convert a nested list to treenetdf
- as_tree_data(Node): data.tree to treenetdf
- as_tree_data(phylo): Phylo tree to treenetdf
- as_tree_data(tbl_graph): tbl_graph_to_treenetdf
- as_tree_data(igraph): Convert igraph tree to treenetdf
- as_tree_data(data.frame): Convert a data.frame to a treenetdf

Examples

```
links <- read.csv(header = TRUE, stringsAsFactors = FALSE, text = '
                  source,target,name
                  1,,one
                  2,1,two
                  3,1,three
                  4,1,four
                  5,2,five
                  6,2,six
                  7,2,seven
                  8,6,eight')

 # Convert data
 as_tree_data(links, cols = c(nodeId = 'source', parentId = 'target'))
```

force_explorer *Interactive* force_network *options explorer*

Description

An interactive shiny widget to explore the force_network options.

Usage

```
force_explorer(data)
```

Arguments

data	a network description in one of numerous forms (see details)

force_network *Create an interactive force network plot in a htmlwidget*

Description

The force_network function creates an interactive force network plot in a htmlwidget

Usage

```
force_network(data, width = NULL, height = NULL, ..., viewer = "internal")
```

Arguments

data	a tree network description in one of numerous forms (see details)
width, height	width and height of exported htmlwidget in pixels (single integer value; default == NULL)
...	other options (see details)
viewer	whether to view the plot in the internal viewer or browser

sankey_explorer *Interactive* sankey_network *options explorer*

Description

An interactive shiny widget to explore the sankey_network options.

Usage

```
sankey_explorer(data)
```

Arguments

data	a network description in one of numerous forms (see details)

sankey_network *Create an interactive sankey network plot in a htmlwidget*

Description

The sankey_network function creates an interactive sankey network plot in a htmlwidget

Usage

```
sankey_network(data, width = NULL, height = NULL, ..., viewer = "internal")
```

Arguments

data	a network description in one of numerous forms (see details)
width, height	width and height of exported htmlwidget in pixels (single integer value; default == NULL)
...	other options (see details)
viewer	whether to view the plot in the internal viewer or browser

save_as_png *Save a PNG screenshot of a htmlwidget*

Description

The save_as_png function takes a screenshot of a htmlwidget as a PNG image.

Usage

```
save_as_png(widget, filepath, background = "white", delay = 0.5)
```

Arguments

widget	a htmlwidget
filepath	a filepath where to save the file
background	the background color underneath/behind the htmlwidget
delay	a delay (in seconds) to wait before taking the screenshot

save_as_svg *Save a SVG screenshot of a htmlwidget*

Description

The save_as_svg function takes a screenshot of a htmlwidget as a SVG image.

Usage

```
save_as_svg(widget, filepath, background = "white", delay = 0.5)
```

Arguments

widget	a htmlwidget
filepath	a filepath where to save the file
background	the background color underneath/behind the htmlwidget
delay	a delay (in seconds) to wait before taking the screenshot

tree_explorer *Interactive* tree_network *options explorer*

Description

An interactive shiny widget to explore the tree_network options.

Usage

```
tree_explorer(data)
```

Arguments

data	a tree network description in one of numerous forms (see details)

tree_network *Create an interactive tree network plot in an htmlwidget*

Description

The tree_network function creates an interactive tree network plot in an htmlwidget

Usage

```
tree_network(
  data,
  width = NULL,
  height = NULL,
  treeType = "tidy",
  direction = "right",
  linkType = "diagonal",
  ...,
  viewer = "internal"
)
```

Arguments

data	a tree network description in one of numerous forms (see details)
width, height	width and height of exported htmlwidget in pixels (single integer value; default == NULL)
treeType	type of tree; one of "tidy" or "cluster" (see details) (default == "tidy")
direction	direction toward which the tree grows; one of "right", "left", "down", or "up" (see details) (default == "right")
linkType	type on link shape; one of "diagonal" or "elbow" (see details) (default == "diagonal")
...	other options (see details)
viewer	whether to view the plot in the internal viewer or browser

Examples

```
treedf <- data.frame(nodeId = LETTERS[1:7],
                     parentId = c("", "A", "A", "B", "B", "C", "C"),
                     name = LETTERS[1:7],
                     stringsAsFactors = FALSE)
tree_network(treedf)
```

Index