Package ‘d3po’

September 23, 2021

Type Package

Title Fast and Beautiful Interactive Visualization for 'Markdown' and 'Shiny'

Version 0.3.2

Description Apache licensed alternative to 'Highcharter' which provides functions for both fast and beautiful interactive visualization for 'Markdown' and 'Shiny'.

Depends htmlwidgets, magrittr, R (>= 2.10)

License Apache License (>= 2.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

NeedsCompilation no

Imports assertthat, dplyr, purrr, rlang

Suggests knitr, igraph, rmarkdown

VignetteBuilder knitr

Author Mauricio Vargas [aut, cre, cph],
    John Coene [aut],
    Riva Quiroga [ctb],
    Ariel Alvarado [ctb],
    Sylvain Lesage [ctb],
    Curran Kelleher [ctb],
    Fernando Becerra [ctb],
    R Consortium [fnd]

Maintainer Mauricio Vargas <mavargas11@uc.cl>

Repository CRAN

Date/Publication 2021-09-23 08:40:05 UTC
R topics documented:

d3po ................................................................. 2
d3po-shiny ......................................................... 3
d3po_graph .......................................................... 4
daes ................................................................. 5
pokemon ............................................................... 5
po_area ............................................................... 6
po_bar ................................................................. 7
po_box ............................................................... 8
po_font .............................................................. 9
po_labels ............................................................ 9
po_legend .......................................................... 10
po_line .............................................................. 10
po_pie ............................................................... 11
po_scatter .......................................................... 12
po_title ............................................................. 13
po_treemap ......................................................... 13

Index 15

---

R

d3po

An htmlwidget interface to the d3po javascript chart library

Description

This function provides 'd3po' methods from R console

Usage

d3po(data = NULL, ..., width = NULL, height = NULL, elementId = NULL)

Arguments

data d3po need explicit specified data objects formatted as JSON, and this parameter passed it from R.

... Aesthetics to pass, see daes()

width Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.

height Same as width parameter.

elementId Dummy string parameter. Useful when you have two or more charts on the same page.

Value

Creates a basic 'htmlwidget' object for simple visualization
**d3po-shiny**

**Author(s)**

Mauricio Vargas

---

**d3po-shiny  Shiny bindings for 'd3po'**

**Description**

Output and render functions for using d3po within Shiny applications and interactive Rmd documents.

**Usage**

```r

d3po_output(output_id, width = "100\%", height = "400px")
render_d3po(expr, env = parent.frame(), quoted = FALSE)
d3po_proxy(id, session = shiny::getDefaultReactiveDomain())
```

**Arguments**

- `output_id`  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 d3po object
- `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.
- `id`  Id of plot to create a proxy of.
- `session`  A valid shiny session.

**Value**

Creates a basic `htmlwidget` object for 'Shiny' and interactive documents
d3po_graph

Graph

Description

Customise edges and nodes.

Usage

```r
po_nodes(d3po, ..., data = NULL, inherit_daes = TRUE)
po_edges(d3po, ..., data = NULL, inherit_daes = TRUE)
po_layout(d3po, method = igraph::layout_nicely)
```

Arguments

d3po Either the output of `d3po()` or `d3po_proxy()`.
...
Aesthetics, see `daes()`.
data Any dataset to use for plot, overrides data passed to `d3po()`.
inherit_daes Whether to inherit aesthetics previous specified.
method The igraph function to compute node positions.

Value

Appends nodes arguments to a network-specific `htmlwidgets` object

Examples

```r
tr <- igraph::make_tree(40, children = 3, mode = "undirected")
d3po(tr) %>%
po_layout()
edges <- igraph::as_data_frame(tr, "edges")
d3po(daes(group = "name")) %>%
po_edges(data = edges)
```
Description

Aesthetics of the chart.

Usage

daes(x, y, ...)

Arguments

x, y, ... List of name value pairs giving aesthetics to map to variables. The names for x and y aesthetics are typically omitted because they are so common; all other aspects must be named.

Value

Aesthetics for the plots such as axis (x,y), group, color and/or size

Aesthetics

Valid aesthetics (depending on the geom)

- x, y: cartesian coordinates.
- group: grouping data.
- color: color of geom.
- size: size of geom.

pokemon

Description

Statistical information about 151 Pokemon from Nintendo RPG series.

Usage

pokemon

Format

A data frame with 151 observations and 15 variables.
Variables

- **id**: Pokedex number.
- **name**: Pokedex name.
- **height**: Height in meters.
- **weight**: Weight in kilograms.
- **base_experience**: How much the Pokemon has battled.
- **type_1**: Primary Pokemon type (i.e. Grass, Fire and Water)
- **type_2**: Secondary Pokemon type (i.e. Poison, Dragon and Ice)
- **attack**: How much damage a Pokemon deals when using a technique.
- **defense**: How much damage a Pokemon receives when it is hit by a technique.
- **hp**: How much damage a Pokemon can receive before fainting.
- **special_attack**: How much damage a Pokemon deals when using a special technique.
- **special_defense**: How much damage a Pokemon receives when it is hit by a special technique.
- **speed**: Determines the order of Pokemon that can act in battle, if the speed is tied then the 1st move is assigned at random.
- **color_1**: Hex color code for Type 1.
- **color_2**: Hex color code for Type 2.

Source

Adapted from highcharter package.

---

<table>
<thead>
<tr>
<th>po_area</th>
<th>Area</th>
</tr>
</thead>
</table>

**Description**

Plot an area chart.

**Usage**

```r
do_area(d3po, ..., data = NULL, inherit_daes = TRUE, stack = FALSE)
```

**Arguments**

- **d3po**: Either the output of `d3po()` or `d3po_proxy()`.
- **...**: Aesthetics, see `daes()`.
- **data**: Any dataset to use for plot, overrides data passed to `d3po()`.
- **inherit_daes**: Whether to inherit aesthetics previous specified.
- **stack**: Whether to stack the series.
Value
an 'htmlwidgets' object with the desired interactive plot

Examples

library(dplyr)
pokemon_density <- density(pokemon$weight, n = 30)
pokemon_density <- tibble(
    x = pokemon_density$x,
    y = pokemon_density$y,
    variable = "weight",
    color = "#5377e3"
)
d3po(pokemon_density) %>%
op_area(
    daes(x = x, y = y, group = variable, color = color)
) %>%
op_title("Approximated Density of Pokemon Weight")

---

Description
Draw a bar chart.

Usage

po_bar(d3po, ..., data = NULL, inherit_daes = TRUE)

Arguments

d3po Either the output of d3po() or d3po_proxy().
...
Aesthetics, see daes().
data Any dataset to use for plot, overrides data passed to d3po().
inherit_daes Whether to inherit aesthetics previous specified.

Value
an 'htmlwidgets' object with the desired interactive plot
Examples

```r
library(dplyr)

pokemon_count <- pokemon %>%
group_by(type_1, color_1) %>%
count()

d3po(pokemon_count) %>%
po_bar(
  daes(x = type_1, y = n, group = type_1, color = color_1)
) %>%
po_title("Count of Pokemon by Type 1")
```

---

**po_box**

*Boxplot*

**Description**

Draw a boxplot.

**Usage**

```r
po_box(d3po, ..., data = NULL, inherit_daes = TRUE)
```

**Arguments**

- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `...`: Aesthetics, see `daes()`.
- `data`: Any dataset to use for plot, overrides data passed to `d3po()`.
- `inherit_daes`: Whether to inherit aesthetics previous specified.

**Value**

an `htmlwidgets` object with the desired interactive plot

**Examples**

```r
d3po(pokemon) %>%
po_box(daes(x = type_1, y = speed, group = name, color = color_1)) %>%
po_title("Distribution of Pokemon Speed")
```
### po_font

**Font**

**Description**

Edit the font used in a chart.

**Usage**

```r
po_font(d3po, font)
```

**Arguments**

- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `font`: Font to use ("Roboto", "Merriweather", etc.).

**Value**

Appends custom font to an 'htmlwidgets' object.

### po_labels

**Labels**

**Description**

Edit labels positioning in a chart.

**Usage**

```r
po_labels(d3po, align, valign)
```

**Arguments**

- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `align`: Horizontal alignment (left, center, right, start, middle, end).
- `valign`: Vertical alignment (top, middle, bottom).

**Value**

Appends custom labels to an 'htmlwidgets' object.
**po_legend**  

**Legend**

**Description**

Add a legend to a chart.

**Usage**

```r
po_legend(d3po, legend)
```

**Arguments**

- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `legend`: Legend to add.

**Value**

Appends custom legend to an 'htmlwidgets' object

---

**po_line**  

**Line**

**Description**

Plot a line chart.

**Usage**

```r
po_line(d3po, ..., data = NULL, inherit_daes = TRUE)
```

**Arguments**

- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `...`: Aesthetics, see `daes()`.
- `data`: Any dataset to use for plot, overrides data passed to `d3po()`.
- `inherit_daes`: Whether to inherit aesthetics previously specified.

**Value**

an 'htmlwidgets' object with the desired interactive plot
Examples

```r
library(dplyr)

pokemon_decile <- pokemon %>%
  filter(type_1 %in% c("grass", "fire", "water")) %>%
  group_by(type_1, color_1) %>%
  summarise(
    decile = 0:10,
    weight = quantile(weight, probs = seq(0, 1, by = .1))
  )

d3po(pokemon_decile) %>%
  po_line(
    daes(x = decile, y = weight, group = type_1, color = color_1)
  ) %>%
  po_title("Decile of Pokemon Weight by Type 1")
```

Description

Plot a pie

Usage

`po_pie(d3po, ..., data = NULL, inherit_daes = TRUE)`

Arguments

- **d3po**: Either the output of `d3po()` or `d3po_proxy()`.
- **...**: Aesthetics, see `daes()`.
- **data**: Any dataset to use for plot, overrides data passed to `d3po()`.
- **inherit_daes**: Whether to inherit aesthetics previous specified.

Value

an 'htmlwidgets' object with the desired interactive plot

Examples

```r
library(dplyr)

pokemon_count <- pokemon %>%
  group_by(type_1, color_1) %>%
  count()
```
```r
d3po(pokemon_count) %>%
po_pie(
  daes(size = n, group = type_1, color = color_1)
) %>%
po_title("Share of Pokemon by Type 1")
```

---

**Description**
Plot an scatter chart.

**Usage**
```
po_scatter(d3po, ..., data = NULL, inherit_daes = TRUE)
```

**Arguments**
- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `...`: Aesthetics, see `daes()`.
- `data`: Any dataset to use for plot, overrides data passed to `d3po()`.
- `inherit_daes`: Whether to inherit aesthetics previous specified.

**Value**
an 'htmlwidgets' object with the desired interactive plot

**Examples**
```
library(dplyr)

pokemon_decile <- pokemon %>%
  filter(type_1 %in% c("grass", "fire", "water")) %>%
  group_by(type_1, color_1) %>%
  summarise(
    decile = 0:10,
    weight = quantile(weight, probs = seq(0, 1, by = .1))
  )

d3po(pokemon_decile) %>%
po_scatter(
  daes(x = decile, y = weight, group = type_1, color = color_1)
) %>%
po_title("Decile of Pokemon Weight by Type 1")
```
### po_title

**Title**

**Description**
Add a title to a chart.

**Usage**
```r
ggplot(data = iris)
```  

**Arguments**
- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `title`: Title to add.

**Value**
Appends a title to an `htmlwidgets` object.

---

### po_treemap

**Treemap**

**Description**
Plot a treemap.

**Usage**
```r
ggplot(data = iris)
```  

**Arguments**
- `d3po`: Either the output of `d3po()` or `d3po_proxy()`.
- `...`: Aesthetics, see `daes()`.
- `data`: Any dataset to use for plot, overrides data passed to `d3po()`.
- `inherit_daes`: Whether to inherit aesthetics previous specified.

**Value**
an `htmlwidgets` object with the desired interactive plot.
Examples

```r
library(dplyr)

pokemon_count <- pokemon %>%
  group_by(type_1, color_1) %>%
  count()

d3po(pokemon_count) %>%
  po_treemap(
    daes(size = n, group = type_1, color = color_1)
  ) %>%
  po_title("Share of Pokemon by Type 1")
```
Index

* datasets
  - pokemon, 5

  d3po, 2
  d3po(), 4, 6–13
  d3po-shiny, 3
  d3po_graph, 4
  d3po_output (d3po-shiny), 3
  d3po_proxy (d3po-shiny), 3
  d3po_proxy(), 4, 6–13
  daes, 5
  daes(), 2, 4, 6–8, 10–13

  po_area, 6
  po_bar, 7
  po_box, 8
  po_edges (d3po_graph), 4
  po_font, 9
  po_labels, 9
  po_layout (d3po_graph), 4
  po_legend, 10
  po_line, 10
  po_nodes (d3po_graph), 4
  po_pie, 11
  po_scatter, 12
  po_title, 13
  po_treemap, 13
  pokemon, 5

  render_d3po (d3po-shiny), 3