Package 'ddplot'

June 29, 2025

Type Package

Title Create D3 Based SVG Graphics

Version 0.0.2

Maintainer Mohamed El Fodil Ihaddaden <ihaddaden.fodeil@gmail.com>

Description Create 'D3' based 'SVG' ('Scalable Vector Graphics') graphics using a simple 'R' API.
The package aims to simplify
the creation of many 'SVG' plot types using a straightforward 'R' API.
The package relies on the 'r2d3' 'R' package and the 'D3' 'JavaScript' library.
See <https://rstudio.github.io/r2d3/> and <https://d3js.org/> respectively.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/feddelegrand7/ddplot

BugReports https://github.com/feddelegrand7/ddplot/issues

RoxygenNote 7.3.1

Imports r2d3

Suggests knitr, rmarkdown, ggplot2, dplyr, tidyr, zoo, gapminder

VignetteBuilder knitr

NeedsCompilation no

Author Mohamed El Fodil Ihaddaden [aut, cre],
June Choe [ctb, cph],
Mike Bostock [ctb, cph] (D3.js developer),
RStudio [ctb, cph] (developers of the r2d3 package)

Repository CRAN

Date/Publication 2025-06-29 11:00:03 UTC

Contents

`animatedHistogram` *Create an animated histogram.*

Description

Create an animated histogram.

Usage

```
animatedHistogram(
  x,
  bins = 30,
  duration = 2000,
  delay = 100,
  fill = "crimson",
  xFontSize = 10,
  yFontSize = 10,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  stroke = "crimson",
  strokeWidth = NULL,
  font = "Verdana, Geneva, Tahoma, sans-serif",
```

```
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

x	A vector of data.
bins	The number of bins to consider. Defaults to 30.
duration	The duration of the bars' transition in milliseconds. Defaults to 2000.
delay	The amount of time (in milliseconds) that precedes before triggering the appearance of each bar. Defaults to 100.
fill	The color of the bars. Defaults to 'crimson'.
xFontSize	the font size of the x-axis labels. Defaults to 10.
yFontSize	the font size of the y-axis labels. Defaults to 10.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
stroke	The stroke color of the bars. Defaults to 'crimson'.
strokeWidth	Optional. the stroke width of the bars.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the bars (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An animated SVG histogram.

Examples

```
animatedHistogram(
 x = mtcars$mpg,
 duration = 2000,
 delay = 100
)
```

`animLineChart` *Create an animated line chart*

Description

Create an animated line chart

Usage

```
animLineChart(
  data,
  x,
  y,
  curve = "curveLinear",
  duration = 5000,
  stroke = "crimson",
  strokeWidth = 1.5,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

`data`	The data frame containing the variables to consider.
`x`	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
`y`	The y-variable to consider.
`curve`	Optional. The line's curve type to render. A complete list can be found here <https://github.com/d3/d3-shape#curves>. Defaults to 'curveLinear'.

duration	The duration in Milliseconds of the animation. Defaults to 5000.
stroke	The color of the line. Defaults to 'crimson'.
strokeWidth	The width of the line. Defaults to 1.5.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the bars (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An animated SVG line chart.

Examples

```
airpassengers <- data.frame(
 passengers = as.matrix(AirPassengers),
 date= zoo::as.Date(time(AirPassengers))
)
animLineChart(
 data = airpassengers,
 x = "date",
 y = "passengers",
 duration = 10000 # in milliseconds (10 seconds)
)
```

areaBand *Create a band chart*

Description

Create a band chart

Usage

```
areaBand(
  data,
  x,
  yLower,
  yUpper,
  fill = "crimson",
  stroke = NULL,
  strokeWidth = NULL,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
yLower	The y-lower band variable to consider.
yUpper	The y-upper band variable to consider.
fill	The color of the band. Defaults to 'crimson'.
stroke	Optional. The color of the stroke of the band.
strokeWidth	Optional. The width of the band stroke.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.

xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the area chart (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG band chart

Examples

```
airpassengers <- data.frame(
 passengers_lower = as.matrix(AirPassengers),
 passengers_upper = as.matrix(AirPassengers) + 40,
 date= zoo::as.Date(time(AirPassengers))
)

areaBand(
 data = airpassengers,
 x = "date",
 yLower = "passengers_lower",
 yUpper = "passengers_upper",
 fill = "yellow",
 stroke = "black"
)
```

---

areaChart *Create an area chart*

Description

Create an area chart

Usage

```
areaChart(
  data,
  x,
  y,
  fill = "crimson",
  stroke = NULL,
  strokeWidth = NULL,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
y	The y-variable to consider.
fill	The color of the area chart. Defaults to 'crimson'.
stroke	Optional. The color of the stroke of the area.
strokeWidth	Optional. The width of the area stroke.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.

opacity	The color opacity of the area chart (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

a SVG area chart

Examples

```
# 1. converting AirPassengers to a tidy data frame
airpassengers <- data.frame(
  passengers = as.matrix(AirPassengers),
  date= zoo::as.Date(time(AirPassengers))
)

# 2. plotting the area chart
areaChart(
  data = airpassengers,
  x = "date",
  y = "passengers",
  fill = "purple",
  bgcol = "white"
)
```

---

barChart *Create a bar chart.*

Description

Create a bar chart.

Usage

```
barChart(
  data,
  x,
  y,
  fill = "crimson",
  sort = "none",
  paddingWidth = 0.1,
  xticks = NULL,
  xFontSize = 10,
  yFontSize = 10,
  yticks = NULL,
  xtitle = NULL,
```

```
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  stroke = "crimson",
  strokeWidth = NULL,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The x-variable to consider.
y	The y-variable to consider.
fill	The color of the bars. Defaults to 'crimson'.
sort	Whether to sort or not the bars. Takes three values 'none' which is the default, 'ascending' or 'descending'.
paddingWidth	The distance between each bar. The value goes from 0 to 0.99 included. Defaults to 0.1.
xticks	Optional. the number of x-axis ticks to consider.
xFontSize	the font size of the x-axis labels. Defaults to 10.
yFontSize	the font size of the y-axis labels. Defaults to 10.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
stroke	The stroke color of the bars. Defaults to 'crimson'.
strokeWidth	Optional. the stroke width of the bars.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the bars (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG bar chart.

Examples

```
library(ggplot2) #needed for the mpg data frame
library(dplyr) #needed for data wrangling

mpg %>% group_by(manufacturer) %>%
 summarise(mean_cty = mean(cty)) %>%
 barChart(
   x = "manufacturer",
   y = "mean_cty",
   sort = "ascending",
   xFontSize = 10,
   yFontSize = 10,
   fill = "orange",
   strokeWidth = 1,
   ytitle = "average cty value",
   title = "Average City Miles per Gallon by manufacturer",
   titleFontSize = 16
 )
```

---

barChartRace *Create a bar chart race.*

Description

Create a bar chart race.

Usage

```
barChartRace(
  data,
  x,
  y,
  time,
  ease = "Linear",
  frameDur = 500,
  transitionDur = 500,
  colorCategory = "Accent",
  sort = "descending",
  paddingWidth = 0.1,
  xFontSize = 10,
  yFontSize = 10,
  xticks = 10,
  xtitle = NULL,
  xtitleFontSize = 16,
```

```
  ytitle = NULL,
  ytitleFontSize = 14,
  title = NULL,
  titleFontSize = 22,
  stroke = "black",
  strokeWidth = NULL,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  panelcol = "#EBEBEBFF",
  xgridlinecol = "white",
  opacity = 1,
  timeLabel = TRUE,
 timeLabelOpts = list(size = 32, prefix = "", suffix = "", xOffset = 0.5, yOffset = 1),
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The x-variable to consider.
y	The y-variable to consider.
time	The time variable to consider.
ease	The easing method, you can find more here <https://github.com/d3/d3-ease>. Defaults to 'Linear'.
frameDur	The time spent paused on each frame (time point) in milliseconds.
transitionDur	The time spent transitioning between frames in milliseconds.
colorCategory	A D3 categorical color scheme, you can find more here <https://github.com/d3/d3-scale-chromatic#categorical>. Defaults to 'Accent'.
sort	Whether to sort or not the bars. Takes three values 'none' which is the default, 'ascending' or 'descending'. Defaults to 'descending'.
paddingWidth	The distance between each bar. The value goes from 0 to 0.99 included. Defaults to 0.1.
xFontSize	the font size of the x-axis labels. Defaults to 10.
yFontSize	the font size of the y-axis labels. Defaults to 10.
xticks	the number of y-axis ticks to consider. Defaults to 10.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 14.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
stroke	The stroke color of the bars. Defaults to 'black'.

strokeWidth	Optional. the stroke width of the bars.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
panelcol	The background color of the panel. Defaults to "#EBEBEBFF" HEX color.
xgridlinecol	The color of the x-axis grid lines. Defaults to 'white'.
opacity	The color opacity of the bars (from 0 to 1). Defaults to 1.
timeLabel	Whether to show a label for the value of the time variable. Defaults to TRUE.
timeLabelOpts	Options for labeling the value of the time variable. Takes a list specifying the 'size', 'prefix', 'suffix', 'xOffset', and 'yOffset'. Offsets are scaled relative to the dimensions of the label, from the bottom-right corner of the panel.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An animated SVG bar chart race, wrapped in a div.

Examples

```
library(gapminder)
library(dplyr)
# let's select a set of countries only
gapminder <- gapminder %>%
filter(
country %in% c("Algeria", "Mexico", "Iceland", "Greece", "Finland")
)

barChartRace(
data = gapminder,
x = "lifeExp",
y = "country",
time = "year",
ytitle = "Country",
xtitle = "Life expectancy",
title = "Bar chart race of countries life expectancy"
)
```

---

flame *Display an Animated Flame Visualization*

Description

Creates an animated flame SVG visualization whose size and color gradient can be customized. The flame grows or shrinks based on the intensity parameter, with smooth pulsing and wobbling animation.

Usage

```
flame(
  intensity = 50,
  flameGradientColors = c("white", "yellow", "darkred"),
  flameOutline = "darkred",
  bgcol = "white",
  width = NULL,
  height = NULL
)
```

Arguments

intensity	Numeric value controlling the size of the flame. Values greater than 100 cause the flame to grow beyond default scaling, while smaller values shrink it. Defaults to 50.
flameGradientColors	A length-3 character vector specifying the colors of the flame gradient, from the center outward. Defaults to `c("white", "yellow", "darkred")`.
flameOutline	Color string for the flame's outline stroke. Defaults to "darkred".
bgcol	Background color of the SVG canvas. Defaults to "white".
width	Optional width of the SVG output.
height	Optional height of the SVG output.

Value

An `r2d3` object displaying the animated flame visualization.

Examples

```
flame(intensity = 5)
flame(
  intensity = 50,
  flameGradientColors = c("lightblue", "blue", "darkblue")
)
```

---

flower *Display a Rotating Flower Visualization*

Description

Display a Rotating Flower Visualization

Usage

```
flower(
  petalCount = 6,
  petalLength = 100,
  petalWidth = 60,
  petalColor = "lightpink",
  petalStroke = "deeppink",
  centerRadius = 20,
  centerColor = "gold",
  centerStroke = "darkorange",
  centerText = NULL,
  centerTextSize = 16,
  centerTextColor = "black",
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "white",
  rotationSpeed = 2,
  width = NULL,
  height = NULL
)
```

Arguments

petalCount	The number of petals. Defaults to 6.
petalLength	The length of each petal. Defaults to 100.
petalWidth	The width of each petal. Defaults to 60.
petalColor	The fill color of the petals. Defaults to "lightpink".
petalStroke	The stroke color of the petals. Defaults to "deeppink".
centerRadius	The radius of the flower's center circle. Defaults to 20.
centerColor	The fill color of the center. Defaults to "gold".
centerStroke	The stroke color of the center. Defaults to "darkorange".
centerText	Optional. Text to display inside the center (e.g., a number or emoji).
centerTextSize	The size of the center text. Defaults to 16.
centerTextColor	The color of the center text. Defaults to "black".
font	The font family for the center text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the visualization. Defaults to "white".
rotationSpeed	The speed of rotation (degrees per animation frame). Defaults to 2.
width	The width of the SVG output. Optional.
height	The height of the SVG output. Optional.

Value

An animated rotating flower SVG.

Examples

```
flower(
  petalCount = 5,
  petalColor = "plum",
  rotationSpeed = 1.5
)
```

`glass_fill` *Visualize a Glass Filling with Water using D3*

Description

This function generates an SVG visualization of a glass filled with water to a specified level, rendered via D3 using the r2d3 package. The fill level, appearance of the glass, and label settings can be customized.

Usage

```
glass_fill(
  fill_level = 0.65,
  glassWidth = 80,
  glassHeight = 200,
  strokeColor = "#555",
  strokeWidth = 3,
  fillColor = "skyblue",
  renderFillLabel = TRUE,
  labelFontSize = "16px",
  titleText = "Fill level",
  labelColor = "#333",
  titleColor = "#333",
  titleFontSize = "14px",
  font = "Verdana, Geneva, Tahoma, sans-serif"
)
```

Arguments

`fill_level`	Numeric value between 0 and 1 indicating how full the glass should appear.
`glassWidth`	Width of the glass in pixels.
`glassHeight`	Height of the glass in pixels.
`strokeColor`	Color of the glass outline (stroke).
`strokeWidth`	Width of the glass outline stroke.
`fillColor`	Color used to fill the water in the glass.
`renderFillLabel`	Logical indicating whether to display a percentage label above the glass.
`labelFontSize`	Font size of the label, defaults to "16px"

titleText	Text to display as the title beneath the glass.
labelColor	Color of the label.
titleColor	Color of the title text displayed below the glass.
titleFontSize	Font size of the title text, defaults to "14px"
font	The font name that will be used for the plot text. Defaults to "Verdana, Geneva, Tahoma, sans-serif"

Value

An interactive D3 visualization rendered in the RStudio Viewer or web browser.

Examples

```
glass_fill(fill_level = 0.75)
glass_fill(fill_level = 0.3, fillColor = "lightblue", titleText = "Water Intake")
```

---

heart_fill *Visualize a Heart Filling with Color using D3*

Description

This function renders a heart-shaped SVG graphic that fills from the bottom up based on the provided level. The appearance of the heart and the optional label can be fully customized. It uses the ‘r2d3‘ package to render the visualization with D3.js.

Usage

```
heart_fill(
  fill_level = 0.65,
  heartSize = 150,
  strokeColor = "#C00",
  strokeWidth = 4,
  fillColor = "red",
  renderFillLabel = TRUE,
  labelColor = "#333",
  labelFontSize = "16px",
  titleText = "Fill level",
  titleColor = "#333",
  titleFontSize = "14px",
  font = "Verdana, Geneva, Tahoma, sans-serif"
)
```

Arguments

fill_level	Value between 0 and 1 indicating how full the heart should appear (e.g., 0.65).
heartSize	Width/height scale of the heart in pixels.
strokeColor	Color of the heart outline.
strokeWidth	Width of the heart outline stroke.
fillColor	Color used to fill the heart based on the fill level.
renderFillLabel	Whether to display a percentage label above the heart.
labelColor	Color of the percentage label text.
labelFontSize	Font size of the percentage label text (e.g., "16px").
titleText	Optional title displayed below the heart.
titleColor	Color of the title text.
titleFontSize	Font size of the title text (e.g., "14px").
font	Font family used for text labels and title.

Value

An interactive D3 heart fill visualization rendered in the RStudio Viewer or web browser.

Examples

```
heart_fill(fill_level = 0.9)
heart_fill(
  fill_level = 0.4,
  fillColor = "pink",
  labelColor = "#C00",
  strokeColor = "#900"
)
```

---

histogram *Create a histogram.*

Description

Create a histogram.

Usage

```
histogram(
  x,
  bins = 30,
  fill = "crimson",
  xFontSize = 10,
```

```
  yFontSize = 10,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  stroke = "crimson",
  strokeWidth = NULL,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

x	A vector of data.
bins	The number of bins to consider. Defaults to 30.
fill	The color of the bars. Defaults to 'crimson'.
xFontSize	the font size of the x-axis labels. Defaults to 10.
yFontSize	the font size of the y-axis labels. Defaults to 10.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
stroke	The stroke color of the bars. Defaults to 'crimson'.
strokeWidth	Optional. the stroke width of the bars.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the bars (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG histogram.

Examples

```
histogram(
  x = mtcars$mpg,
  bins = 20,
  fill = "crimson",
  stroke = "white",
  strokeWidth = 1,
  title = "Distribution of the hwy variable",
  width = "20",
  height = "10"
)
```

---

horzBarChart *Create a horizontal bar chart*

Description

Create a horizontal bar chart

Usage

```
horzBarChart(
  data,
  label,
  value,
  fill = "crimson",
  sort = "none",
  paddingWidth = 0.1,
  stroke = NULL,
  strokeWidth = 1,
  bgcol = "#CAD0D3",
  valueTicks = NULL,
  valueFontSize = 10,
  labelFontSize = 10,
  valueTitle = NULL,
  valueTitleFontSize = 14,
  labelTitle = NULL,
  labelTitleFontSize = 14,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  title = NULL,
  titleFontSize = 20,
  opacity = 1,
  axisCol = "black",
  width = NULL,
```

```
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
label	The categorical variable to consider. Will be plotted on the y-axis.
value	The numeric variable to consider. Will be plotted on the x-axis.
fill	The color of the bars. Defaults to 'crimson'.
sort	Optional. Takes the following arguments: 'none', 'ascending' or 'descending', default to 'none'
paddingWidth	The distance between each bar. The value goes from 0 to 0.99 included. Defaults to 0.1.
stroke	Optional. The color of the stroke of the bars.
strokeWidth	The width of the stroke of the bars. Defaults to 1 when the 'stroke' parameter is used.
bgcol	Optional. The color of the background, default to: '#CAD0D3'
valueTicks	Optional. the number of x-axis ticks to consider.
valueFontSize	The font size of the x-axis values. Defaults to 10.
labelFontSize	The font size of the y-axis labels. Defaults to 10.
valueTitle	Optional. The title of the x-axis.
valueTitleFontSize	The font size of the x-axis title if specified. Defaults to 14.
labelTitle	Optional. The title of the y-axis.
labelTitleFontSize	The font size of the y-axis title. Defaults to 14.
font	The font family of the text. Defaults to "Verdana, Geneva, Tahoma, sans-serif"
title	Optional. The title of the overall graphic.
titleFontSize	The font size of the overall graphic's title when specified.
opacity	The color opacity of the bars. Goes from 0 to 1. Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG horizontal bar chart.

Examples

```
library(ggplot2) # needed for the mpg data frame
library(dplyr) # needed for the data wrangling process

mpg %>% group_by(manufacturer) %>%
 summarise(median_hwy = median(hwy)) %>%
 horzBarChart(
   label = "manufacturer",
   value = "median_hwy",
   sort = "ascending"
 )
```

---

horzLollipop *Create a horizontal lollipop chart*

Description

Create a horizontal lollipop chart

Usage

```
horzLollipop(
  data,
  label,
  value,
  sort = "none",
  bgcol = "white",
  valueTicks = NULL,
  labelTicks = NULL,
  valueFontSize = 12,
  labelFontSize = 12,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  valueTitle = NULL,
  valueTitleFontSize = 14,
  labelTitle = NULL,
  labelTitleFontSize = 14,
  title = NULL,
  titleFontSize = 20,
  lineStroke = "maroon",
  lineStrokeWidth = 4,
  circleFill = "lime",
  circleStroke = "lime",
  circleStrokeWidth = 1,
  circleRadius = 5,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
label	The categorical variable to consider. Will be plotted on the x-axis.
value	The numeric variable to consider. Will be plotted on the y-axis.
sort	Whether to sort or not the vertical lines. Takes three values 'none' which is the default, 'ascending' or 'descending'.
bgcol	The background-color of the SVG output. Defaults to 'salmon'.
valueTicks	Optional. the number of x-axis ticks to consider.
labelTicks	Optional. The number of y-axis ticks to consider.
valueFontSize	the font size of the x-axis labels. Defaults to 10.
labelFontSize	the font size of the y-axis labels. Defaults to 10.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
valueTitle	Optional. The title of the x-axis.
valueTitleFontSize	The font size of the x-axis title. Defaults to 14.
labelTitle	Optional. The title of the y-axis.
labelTitleFontSize	The font size of the y-axis title. Defaults to 14.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
lineStroke	The stroke color of the vertical lines. Defaults to 'maroon'.
lineStrokeWidth	The vertical lines stroke's width. Defaults to 4.
circleFill	The color of the circles. Defaults to 'lime'.
circleStroke	The color of the stroke surrounding the circle. Defaults to 'lime'.
circleStrokeWidth	The width of the circles' stroke. Defaults to 1.
circleRadius	The radius of the circles. Defaults to 10.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG horizontal lollipop chart.

lineChart *Create a line chart*

Description

Create a line chart

Usage

```
lineChart(
  data,
  x,
  y,
  curve = "curveLinear",
  stroke = "crimson",
  strokeWidth = 1.5,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
y	The y-variable to consider.
curve	The line's curve type to render. A complete list can be found here <https://github.com/d3/d3-shape#curves>. Defaults to 'curveLinear'.
stroke	The color of the line. Defaults to 'crimson'.
strokeWidth	The width of the line. Defaults to 1.5.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.

xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the bars (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG line chart.

Examples

```
# 1. converting AirPassengers to a tidy data frame
airpassengers <- data.frame(
  passengers = as.matrix(AirPassengers),
  date= zoo::as.Date(time(AirPassengers))
)

# 2. plotting the line chart
lineChart(
  data = airpassengers,
  x = "date",
  y = "passengers"
)
```

---

lollipopChart *Create a lollipop chart*

Description

Create a lollipop chart

Usage

```
lollipopChart(
  data,
  x,
  y,
  sort = "none",
  bgcol = "white",
  xticks = NULL,
  yticks = NULL,
  xFontSize = 12,
  yFontSize = 12,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  xtitle = NULL,
  xtitleFontSize = 14,
  ytitle = NULL,
  ytitleFontSize = 14,
  title = NULL,
  titleFontSize = 20,
  lineStroke = "maroon",
  lineStrokeWidth = 4,
  circleFill = "lime",
  circleStroke = "lime",
  circleStrokeWidth = 1,
  circleRadius = 10,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The categorical variable to consider. Will be plotted on the x-axis.
y	The numeric variable to consider. Will be plotted on the y-axis.
sort	Whether to sort or not the vertical lines. Takes three values 'none' which is the default, 'ascending' or 'descending'.
bgcol	The background-color of the SVG output. Defaults to 'white'.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xFontSize	the font size of the x-axis labels. Defaults to 10.
yFontSize	the font size of the y-axis labels. Defaults to 10.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.

ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
lineStroke	The stroke color of the vertical lines. Defaults to 'maroon'.
lineStrokeWidth	The vertical lines stroke's width. Defaults to 4.
circleFill	The color of the circles. Defaults to 'lime'.
circleStroke	The color of the stroke surrounding the circle. Defaults to 'lime'.
circleStrokeWidth	The width of the circles' stroke. Defaults to 1.
circleRadius	The radius of the circles. Defaults to 10.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG lollipop chart.

Examples

```
library(ggplot2) # needed for the mpg data frame
library(dplyr) # needed for data wrangling

mpg %>% group_by(drv) %>%
  summarise(median_cty = median(cty)) %>%
  lollipopChart(
    x = "drv",
    y = "median_cty",
    sort = "ascending",
    xtitle = "drv variable",
    ytitle = "median cty",
    title = "Median cty per drv"
  )
```

---

parliament_chart *Display a Parliament Chart*

Description

Display a Parliament Chart

Usage

```
parliament_chart(
  data,
  categorical_column,
  numerical_column,
  seatSize = 6,
  padding = 2,
  maxRows = NULL,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  width = NULL,
  height = NULL
)
```

Arguments

data	A data frame with a categorical column and a numerical column
categorical_column	The categorical column to consider
numerical_column	The numerical column to consider
seatSize	The size of each seat. Defaults to 6.
padding	The padding between seats. Defaults to 2.
maxRows	The maximum number of rows. Optional.
title	The title of the chart. Optional.
titleFontSize	Font size for the title. Defaults to 22.
font	Font family for text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	Background color of the chart. Defaults to "#CAD0D3".
width	Width of the SVG canvas. Optional.
height	Height of the SVG canvas. Optional.

Value

A D3-rendered Parliament chart

Examples

```
political_results_example <- data.frame(
    political_party = c("SDP", "CDU", "Linke"),
    number_of_seats = c(200, 40, 30)
)
parliament_chart(
  data = political_results_example,
  categorical_column = "political_party",
```

```
  numerical_column = "number_of_seats",
  title = "German Bundestag (results not real, just an example)",
  seatSize = 10,
  bgcol = "#fefefe"
)
```

`pieChart` *Create a pie chart*

Description

Create a pie chart

Usage

```
pieChart(
  data,
  value,
  label,
  colorCategory = "Paired",
  innerRadius = 0,
  outerRadius = "auto",
  padRadius = 0,
  padAngle = NULL,
  cornerRadius = 0,
  labelFont = "sans-serif",
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "white",
  opacity = 1,
  labelHeight = 18,
  width = NULL,
  height = NULL
)
```

Arguments

`data`	The data frame to consider.
`value`	The numeric variable to consider.
`label`	The labeling variable to consider.
`colorCategory`	A D3 categorical color scheme, you can find more here <https://github.com/d3/d3-scale-chromatic#categorical>. Defaults to 'Paired'
`innerRadius`	The size of the inner radius of the pie. Defaults to 0. Set the inner radius to a higher value to plot a donut chart.
`outerRadius`	The size of the outer radius of the pie.

padRadius	From the D3 official documentation, The pad radius compute the fixed linear distance separating adjacent arcs, defined as padRadius * padAngle.
padAngle	Optional. From the D3 official documentation, the padAngle is used to set the padding angle between consecutive arcs.
cornerRadius	From the D3 official documentation, the value of the corner radius for rounded corners. If the corner radius is greater than zero, the corners of the arc are rounded using circles of the given radius. Defaults to 0.
labelFont	The font family of the legend. Defaults to 'sans-serif'.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "white".
opacity	The color opacity of the pie (from 0 to 1). Defaults to 1.
labelHeight	The height of the legend. Defaults to 18.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

A SVG pie chart

Examples

```
library(dplyr) # needed for the starwars data frame

# starwars is part of the dplyr data frame
mini_starwars <- starwars %>% tidyr::drop_na(mass) %>%
  sample_n(size = 5) # getting 5 random values

pieChart(
  data = mini_starwars,
  value = "mass",
  label = "name"
)
```

---

plant_growth *Visualize Plant Growth Using D3*

Description

This function creates a plant growth meter visualization.

Usage

```
plant_growth(
  fill_level = 0.5,
  potWidth = 100,
  potHeight = 40,
  plantMaxHeight = 150,
  stemColor = "#228B22",
  potColor = "#8B4513",
  flowerColor = "#FF69B4",
  strokeColor = "#333",
  strokeWidth = 2,
  renderFillLabel = TRUE,
  titleText = "Plant Growth",
  titleColor = "#333",
  titleFontSize = "14px",
  font = "sans-serif"
)
```

Arguments

`fill_level`	Numeric between 0 and 1 indicating growth level.
`potWidth`	Width of the pot.
`potHeight`	Height of the pot.
`plantMaxHeight`	Max height of plant stem.
`stemColor`	Color of the plant stem and leaves.
`potColor`	Color of the pot.
`flowerColor`	Color of the flowers that bloom when growth is high.
`strokeColor`	Outline color for the pot.
`strokeWidth`	Outline width.
`renderFillLabel`	Whether to display a growth label.
`titleText`	Title shown below the pot.
`titleColor`	Title color.
`titleFontSize`	Font size of the title.
`font`	Font family.

`scatterPlot` *Create a scatter plot.*

Description

Create a scatter plot.

Usage

```
scatterPlot(
  data,
  x,
  y,
  col = "crimson",
  size = 2,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  stroke = NULL,
  strokeWidth = NULL,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
  axisCol = "black",
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the quantitative variables.
x	The x-variable to consider.
y	The y-variable to consider.
col	The color of the dots. Defaults to 'crimson'.
size	The size of the dots. Defaults to 2.
xticks	Optional. The number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. the title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. the title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
stroke	Optional. the stroke color of the dots.
strokeWidth	Optional. the stroke width of the dots.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".

bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the dots (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
width	Optional. the width of the SVG output.
height	Optional. the height of the SVG output.

Value

A SVG scatter plot.

Examples

```
scatterPlot(
  data = mtcars,
  x = "mpg",
  y = "wt"
  )
```

---

stackedAreaChart *Create a stacked area chart*

Description

Create a stacked area chart

Usage

```
stackedAreaChart(
  data,
  x,
  colorCategory = "Category10",
  curve = "curveLinear",
  stroke = NULL,
  strokeWidth = 1.5,
  xticks = NULL,
  yticks = NULL,
  xtitle = NULL,
  xtitleFontSize = 16,
  ytitle = NULL,
  ytitleFontSize = 16,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#CAD0D3",
  opacity = 1,
```

```
  axisCol = "black",
  legendBoxSize = 18,
  legendTextSize = 18,
  width = NULL,
  height = NULL
)
```

Arguments

data	The data frame containing the variables to consider.
x	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
colorCategory	A D3 categorical color scheme, you can find more here <https://github.com/d3/d3-scale-chromatic#categorical>. Defaults to 'Category10'.
curve	The line's curve type to render. A complete list can be found here <https://github.com/d3/d3-shape#curves>. Defaults to 'curveLinear'.
stroke	Optional. The color of the stroke of the area.
strokeWidth	The width of the line. Defaults to 1.5.
xticks	Optional. the number of x-axis ticks to consider.
yticks	Optional. The number of y-axis ticks to consider.
xtitle	Optional. The title of the x-axis.
xtitleFontSize	The font size of the x-axis title. Defaults to 16.
ytitle	Optional. The title of the y-axis.
ytitleFontSize	The font size of the y-axis title. Defaults to 16.
title	Optional. The title of the plot.
titleFontSize	The font size of the plot title. Defaults to 22.
font	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
opacity	The color opacity of the area chart (from 0 to 1). Defaults to 1.
axisCol	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
legendBoxSize	The size of the legend rectangles. Defaults to 18.
legendTextSize	The font size of the legend text Defaults to 18.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

a SVG stacked area chart

Examples

```
data <- data.frame(
  date = c(
    "2000-01-01", "2000-02-01", "2000-03-01", "2000-04-01",
    "2000-05-01", "2000-06-01", "2000-07-01",
    "2000-08-01", "2000-09-01", "2000-10-01"
  ),
  Trade = c(
    2000,1023, 983, 2793, 1821, 1837, 1792, 1853, 791, 739
  ),
  Manufacturing = c(
    734, 694, 739, 736, 685, 621, 708, 685, 667, 693
  ),
  Leisure = c(
    1782, 1779, 1789, 658, 675, 833, 786, 675, 636, 691
  ),
  Agriculture = c(
    655, 587,623, 517, 561, 2545, 636, 584, 559, 2504
  )
)

stackedAreaChart(
  data = data,
  x = "date",
  legendTextSize = 14,
  curve = "curveCardinal",
  colorCategory = "Accent",
  bgcol = "white",
  stroke = "black",
  strokeWidth = 1
)
```

Index