

Package: ddplot (via r-universe)

October 26, 2026

Type Package

Title Create D3 Based SVG Graphics

Version 0.1.0

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

LazyData true

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

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

RoxygenNote 7.3.2

Imports r2d3

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

VignetteBuilder knitr

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://feddelegrand7.r-universe.dev>

Date/Publication 2026-08-25 14:04:43 UTC

RemoteUrl <https://github.com/feddelegrand7/ddplot>

RemoteRef HEAD

RemoteSha 38af222d193d19c8a349eeec04aeec40b1c6389c

Contents

anim_line_chart	2
animated_histogram	4
area_band	6
area_chart	7
bar_chart	9
bar_chart_race	11
beeswarm_plot	13
bullet_chart	16
dumbbell_chart	18
flame	21
flower	22
gauge_chart	23
glass_fill	25
heart_fill	26
histogram	28
horz_bar_chart	29
horz_lollipop	31
line_chart	33
liquid_chart	35
lollipop_chart	36
parliament_chart	38
pie_chart	40
plant_growth	41
pulse_grid	42
scatter_plot	45
stacked_area_chart	46
waterfall_chart	48
Index	52

anim_line_chart	Create an animated line chart
-----------------	-------------------------------

Description

Create an animated line chart

Usage

```
anim_line_chart(  
  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))  
)  
anim_line_chart(  
  data = airpassengers,  
  x = "date",  
  y = "passengers",  
  duration = 10000 # in milliseconds (10 seconds)  
)
```

animated_histogram	Create an animated histogram.
--------------------	-------------------------------

Description

Create an animated histogram.

Usage

```
animated_histogram(  
  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

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

Value

An animated SVG histogram.

Examples

```

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

```

area_band

*Create a band chart***Description**

Create a band chart

Usage

```

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

area_chart	Create an area chart
------------	----------------------

Description

Create an area chart

Usage

```

area_chart(
  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

<code>data</code>	The data frame containing the variables to consider.
<code>x</code>	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
<code>y</code>	The y-variable to consider.
<code>fill</code>	The color of the area chart. Defaults to 'crimson'.
<code>stroke</code>	Optional. The color of the stroke of the area.
<code>strokeWidth</code>	Optional. The width of the area stroke.
<code>xticks</code>	Optional. the number of x-axis ticks to consider.
<code>yticks</code>	Optional. The number of y-axis ticks to consider.
<code>xtitle</code>	Optional. The title of the x-axis.
<code>xtitleFontSize</code>	The font size of the x-axis title. Defaults to 16.
<code>ytitle</code>	Optional. The title of the y-axis.
<code>ytitleFontSize</code>	The font size of the y-axis title. Defaults to 16.
<code>title</code>	Optional. The title of the plot.
<code>titleFontSize</code>	The font size of the plot title. Defaults to 22.
<code>font</code>	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
<code>bgcol</code>	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
area_chart(
  data = airpassengers,
  x = "date",
  y = "passengers",
  fill = "purple",
  bgcol = "white"
)
```

bar_chart

Create a bar chart.

Description

Create a bar chart.

Usage

```
bar_chart(
  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)) %>%
  bar_chart(
    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
  )
```

bar_chart_race	Create a bar chart race.
----------------	--------------------------

Description

Create a bar chart race.

Usage

```
bar_chart_race(
  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")
  )

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

Description

A beeswarm plot displays individual data points along a numeric axis, spreading them out so they do not overlap (like a swarm of bees). It is a transparent alternative to box plots or violin plots because every observation is visible. When a grouping variable is supplied, one swarm is drawn per group along the opposite axis.

The layout is computed with a D3 force simulation that pulls each point toward its true value on the x-axis while resolving collisions so no two circles overlap.

The plot is interactive: hovering a point highlights it while dimming the rest of the swarm, and (when tooltip columns are supplied) shows a tooltip with those columns' values. When a group is supplied, hovering a group label focuses that swarm and clicking a label toggles it on or off. An optional mean/median reference line can be drawn per swarm. The interactive behaviors are controlled by tooltip, hover, animate, and stat.

Usage

```
beeswarm_plot(  
  data,  
  x,  
  group = NULL,  
  tooltip = NULL,  
  hover = TRUE,  
  animate = TRUE,  
  stat = c("none", "mean", "median"),  
  statColor = "#d62728",  
  col = "steelblue",  
  colorPalette = "Tableau10",  
  radius = 4,  
  opacity = 0.75,  
  stroke = "white",  
  strokeWidth = 0.5,  
  xtitle = NULL,  
  xtitleFontSize = 13,  
  title = NULL,  
  titleFontSize = 16,  
  xFontSize = 11,  
  yFontSize = 11,  
  font = "Verdana, Geneva, Tahoma, sans-serif",  
  bgcol = "white",  
  axisCol = "black",  
  width = NULL,  
  height = NULL  
)
```

Arguments

data	The data frame containing the variables to consider.
x	The name of the numeric column whose distribution is displayed.

group	Optional. The name of a categorical column used to split the data into separate swarms. Defaults to NULL (single swarm).
tooltip	Optional. A column name, or a vector of column names, whose values are shown in a tooltip when a point is hovered. Each column appears on its own line as "column: value". When NULL (default) no tooltip is shown.
hover	Logical. When TRUE (default), hovering a point enlarges it and dims the rest of the swarm to make it stand out.
animate	Logical. When TRUE (default), points grow into place with a short staggered entry animation on first render.
stat	Character. Draws a reference line per swarm at the "mean" or "median" of the values. Defaults to "none" (no line).
statColor	Color of the mean/median reference line and its label. Defaults to "#d62728".
col	Fill color of the points when no group is supplied. Defaults to "steelblue".
colorPalette	D3 categorical color scheme used when group is supplied (e.g. "Tableau10", "Category10", "Set2"). Defaults to "Tableau10".
radius	Radius of each point in pixels. Defaults to 4.
opacity	Fill opacity of the points (0 to 1). Defaults to 0.75.
stroke	Stroke color of the points. Defaults to "white".
strokeWidth	Stroke width in pixels. Defaults to 0.5.
xtitle	Optional. Title of the x-axis.
xtitleFontSize	Font size of the x-axis title. Defaults to 13.
title	Optional. Title of the chart.
titleFontSize	Font size of the chart title. Defaults to 16.
xFontSize	Font size of the x-axis tick labels. Defaults to 11.
yFontSize	Font size of the y-axis group labels. Defaults to 11.
font	The font family used for all text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	Background color of the SVG. Defaults to "white".
axisCol	Color of axis lines, ticks, and labels. Defaults to "black".
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An SVG beeswarm plot.

Examples

```
library(ggplot2) # for the mpg dataset

# Single swarm - distribution of highway fuel economy
beeswarm_plot(
  data = mpg,
```

```

    x      = "hwy",
    col    = "steelblue",
    xtitle = "Highway miles per gallon",
    title  = "Distribution of hwy"
  )

  # Grouped swarm - hwy by vehicle class
  beeswarm_plot(
    data = mpg,
    x     = "hwy",
    group = "class",
    xtitle = "Highway miles per gallon",
    title  = "Highway fuel economy by vehicle class"
  )

  # Grouped swarm on the iris dataset
  beeswarm_plot(
    data      = iris,
    x         = "Sepal.Length",
    group     = "Species",
    colorPalette = "Set2",
    radius    = 5,
    xtitle    = "Sepal length (cm)",
    title     = "Sepal length by species"
  )

  # Interactive extras: median line per swarm and a tooltip label column
  beeswarm_plot(
    data = mpg,
    x     = "hwy",
    group = "class",
    tooltip = "model",
    stat    = "median",
    xtitle  = "Highway miles per gallon",
    title   = "Highway fuel economy by vehicle class"
  )

```

bullet_chart

Create a bullet chart.

Description

A bullet chart displays a single quantitative measure (the actual value) against qualitative range bands (e.g. poor / satisfactory / good) and an optional target marker. It was designed by Stephen Few as a compact, information-dense alternative to gauge charts.

Usage

```
bullet_chart(
  value,
```

```

    ranges,
    target = NULL,
    rangeColors = NULL,
    title = "",
    subtitle = "",
    min = 0,
    valueColor = "#1a1a2e",
    targetColor = "#e63946",
    tickCount = 5,
    fontSize = 12,
    font = "Verdana, Geneva, Tahoma, sans-serif",
    bgcol = "white",
    width = NULL,
    height = NULL
  )

```

Arguments

value	Numeric. The actual value to display (the main bar).
ranges	Numeric vector of upper bounds for each qualitative band, in ascending order (e.g. <code>c(40, 70, 100)</code>). The lower bound of the first band is min.
target	Optional numeric. The target / goal value shown as a short vertical marker. Defaults to NULL (no marker drawn).
rangeColors	Character vector of colors for each band, in the same order as ranges. Defaults to a sequence of grays from darker (low range) to lighter (high range).
title	Character. Label displayed to the left of the chart. Defaults to "".
subtitle	Character. Smaller label displayed below title. Defaults to "".
min	Numeric. The minimum value of the scale. Defaults to 0.
valueColor	The color of the actual-value bar. Defaults to "#1a1a2e".
targetColor	The color of the target marker. Defaults to "#e63946".
tickCount	Number of axis ticks. Defaults to 5.
fontSize	Base font size in pixels. Defaults to 12.
font	The font family used for all text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "white".
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An SVG bullet chart.

Examples

```
# Sales revenue vs target
bullet_chart(
  value = 270,
  target = 300,
  ranges = c(150, 225, 350),
  title = "Revenue",
  subtitle = "USD thousands"
)

# Customer satisfaction score
bullet_chart(
  value = 7.4,
  target = 8.0,
  ranges = c(4, 7, 10),
  title = "Satisfaction",
  subtitle = "out of 10",
  valueColor = "steelblue"
)

# Server response time (lower is better – ranges go poor → good left to right)
bullet_chart(
  value = 320,
  target = 250,
  ranges = c(200, 500, 1000),
  rangeColors = c("#f5f5f5", "#e0e0e0", "#c0c0c0"),
  title = "Response",
  subtitle = "ms",
  min = 0
)
```

dumbbell_chart

Create a dumbbell chart.

Description

A dumbbell chart (also called a connected dot plot) displays two values per category as dots connected by a horizontal line. It is ideal for showing the change or gap between two groups, time points, or conditions across multiple categories — a clean alternative to grouped bar charts favoured by outlets such as the New York Times and The Economist.

Usage

```
dumbbell_chart(
  data,
  x1,
  x2,
  y,
  x1Label = NULL,
```

```

x2Label = NULL,
col1 = "steelblue",
col2 = "crimson",
lineCol = "#aaaaaa",
lineWidth = 2,
circleRadius = 6,
sort = "none",
xtitle = NULL,
xtitleFontSize = 13,
title = NULL,
titleFontSize = 16,
xFontSize = 11,
yFontSize = 11,
legend = TRUE,
legendFontSize = 11,
font = "Verdana, Geneva, Tahoma, sans-serif",
bgcol = "white",
axisCol = "black",
width = NULL,
height = NULL
)

```

Arguments

<code>data</code>	The data frame containing the variables to consider.
<code>x1</code>	The name of the column for the first value (left dot).
<code>x2</code>	The name of the column for the second value (right dot).
<code>y</code>	The name of the categorical column used as row labels.
<code>x1Label</code>	Legend label for the first value. Defaults to the value of <code>x1</code> .
<code>x2Label</code>	Legend label for the second value. Defaults to the value of <code>x2</code> .
<code>col1</code>	Color of the first dot. Defaults to "steelblue".
<code>col2</code>	Color of the second dot. Defaults to "crimson".
<code>lineCol</code>	Color of the line connecting the two dots. Defaults to "#aaaaaa".
<code>lineWidth</code>	Width of the connecting line in pixels. Defaults to 2.
<code>circleRadius</code>	Radius of the dots in pixels. Defaults to 6.
<code>sort</code>	Whether to sort the rows. One of "none" (default), "ascending", or "descending", ordered by the <code>x1</code> value.
<code>xtitle</code>	Optional. Title of the x-axis.
<code>xtitleFontSize</code>	Font size of the x-axis title. Defaults to 13.
<code>title</code>	Optional. Title of the chart.
<code>titleFontSize</code>	Font size of the chart title. Defaults to 16.
<code>xFontSize</code>	Font size of the x-axis tick labels. Defaults to 11.
<code>yFontSize</code>	Font size of the y-axis category labels. Defaults to 11.

<code>legend</code>	Whether to display a legend. Defaults to TRUE.
<code>legendFontSize</code>	Font size of the legend text. Defaults to 11.
<code>font</code>	The font family used for all text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
<code>bgcol</code>	Background color of the SVG. Defaults to "white".
<code>axisCol</code>	Color of axis lines, ticks, and labels. Defaults to "black".
<code>width</code>	Optional. The width of the SVG output.
<code>height</code>	Optional. The height of the SVG output.

Value

An SVG dumbbell chart.

Examples

```
# Life expectancy change between 1952 and 2007 for selected countries
life_exp <- data.frame(
  country = c("Brazil", "China", "Egypt", "India",
              "Japan", "Mexico", "Nigeria", "Turkey"),
  year_1952 = c(50.9, 44.0, 41.9, 37.4, 63.0, 50.8, 36.3, 43.6),
  year_2007 = c(72.4, 72.9, 71.3, 64.7, 82.6, 76.2, 46.9, 71.8)
)

dumbbell_chart(
  data      = life_exp,
  x1        = "year_1952",
  x2        = "year_2007",
  y         = "country",
  x1Label   = "1952",
  x2Label   = "2007",
  title     = "Life expectancy: 1952 vs 2007",
  xtitle    = "Life expectancy (years)",
  sort      = "ascending"
)

# Comparing city vs highway fuel economy by car class
library(ggplot2)
library(dplyr)

mpg_summary <- summarise(group_by(mpg, class),
  city = mean(cty), highway = mean(hwy))

dumbbell_chart(
  data      = mpg_summary,
  x1        = "city",
  x2        = "highway",
  y         = "class",
  x1Label   = "City",
  x2Label   = "Highway",
  col1     = "steelblue",
```

```

col2    = "darkorange",
title   = "City vs highway fuel economy by class",
xtitle  = "Miles per gallon",
sort    = "ascending"
)

```

flame	<i>Display an Animated Flame Visualization</i>
-------	--

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
)
```

gauge_chart

Create a gauge (speedometer) chart.

Description

A gauge (speedometer) chart displays a single quantitative measure against colored zones (warning and danger) with a needle indicator. By default the needle animates on load, sweeping up from the minimum to the value and settling with a small wobble, like a car dashboard.

Usage

```
gauge_chart(
  value,
  min = 0,
  max = 100,
  title = "",
  warningZone = NULL,
  warningColor = "orange",
  dangerZone = NULL,
  dangerColor = "red",
  arcBgColor = "#ddd",
  needleColor = "crimson",
  pivotColor = "steelblue",
```

```

    titleFontSize = 16,
    valueFontSize = 14,
    font = "Verdana, Geneva, Tahoma, sans-serif",
    bgcol = "white",
    animate = TRUE,
    animationDuration = 1500,
    width = NULL,
    height = NULL
  )

```

Arguments

value	Numeric. The current value to display on the gauge.
min	Numeric. The minimum value of the gauge. Defaults to 0.
max	Numeric. The maximum value of the gauge. Defaults to 100.
title	Character. Label shown inside the gauge. Defaults to "".
warningZone	Numeric. The value at which the warning (first) arc begins. Defaults to 75% of max.
warningColor	The color of the warning zone arc. Defaults to "orange".
dangerZone	Numeric. The value at which the danger (second, critical) arc begins. Defaults to 90% of max.
dangerColor	The color of the danger zone arc. Defaults to "red".
arcBgColor	The background color of the gauge arc track. Defaults to "#ddd".
needleColor	The color of the needle. Defaults to "crimson".
pivotColor	The color of the center pivot circle. Defaults to "steelblue".
titleFontSize	The font size of the gauge title. Defaults to 16.
valueFontSize	The font size of the current value label. Defaults to 14.
font	The font family used for all text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	The background color of the SVG. Defaults to "white".
animate	Logical. When TRUE (default), the needle sweeps from the minimum up to the value on load and settles with a small wobble. Set to FALSE to draw the needle at the value immediately with no animation.
animationDuration	Numeric. Duration of the needle sweep in milliseconds. Defaults to 1500.
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An SVG gauge chart.

Examples

```
# Basic usage
gauge_chart(value = 8, title = "Memory")

# Custom warning and danger thresholds
gauge_chart(
  value = 72,
  min = 0,
  max = 100,
  title = "CPU Load",
  warningZone = 60,
  warningColor = "orange",
  dangerZone = 80,
  dangerColor = "red"
)

# Non-percentage range (response time in ms)
gauge_chart(
  value = 530,
  min = 0,
  max = 1000,
  title = "Response (ms)",
  warningZone = 400,
  dangerZone = 700
)
```

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"
)

```

horz_bar_chart

Create a horizontal bar chart

Description

Create a horizontal bar chart

Usage

```

horz_bar_chart(
  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

<code>data</code>	The data frame containing the variables to consider.
<code>label</code>	The categorical variable to consider. Will be plotted on the y-axis.
<code>value</code>	The numeric variable to consider. Will be plotted on the x-axis.
<code>fill</code>	The color of the bars. Defaults to 'crimson'.
<code>sort</code>	Optional. Takes the following arguments: 'none', 'ascending' or 'descending', default to 'none'
<code>paddingWidth</code>	The distance between each bar. The value goes from 0 to 0.99 included. Defaults to 0.1.
<code>stroke</code>	Optional. The color of the stroke of the bars.
<code>strokeWidth</code>	The width of the stroke of the bars. Defaults to 1 when the 'stroke' parameter is used.
<code>bgcol</code>	Optional. The color of the background, default to: '#CAD0D3'
<code>valueTicks</code>	Optional. the number of x-axis ticks to consider.
<code>valueFontSize</code>	The font size of the x-axis values. Defaults to 10.
<code>labelFontSize</code>	The font size of the y-axis labels. Defaults to 10.
<code>valueTitle</code>	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)) %>%
  horz_bar_chart(
    label = "manufacturer",
    value = "median_hwy",
    sort = "ascending"
  )
```

horz_lollipop

Create a horizontal lollipop chart

Description

Create a horizontal lollipop chart

Usage

```
horz_lollipop(
  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

<code>data</code>	The data frame containing the variables to consider.
<code>label</code>	The categorical variable to consider. Will be plotted on the x-axis.
<code>value</code>	The numeric variable to consider. Will be plotted on the y-axis.
<code>sort</code>	Whether to sort or not the vertical lines. Takes three values 'none' which is the default, 'ascending' or 'descending'.
<code>bgcol</code>	The background-color of the SVG output. Defaults to 'salmon'.
<code>valueTicks</code>	Optional. the number of x-axis ticks to consider.
<code>labelTicks</code>	Optional. The number of y-axis ticks to consider.
<code>valueFontSize</code>	the font size of the x-axis labels. Defaults to 10.
<code>labelFontSize</code>	the font size of the y-axis labels. Defaults to 10.
<code>font</code>	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
<code>valueTitle</code>	Optional. The title of the x-axis.
<code>valueTitleFontSize</code>	The font size of the x-axis title. Defaults to 14.
<code>labelTitle</code>	Optional. The title of the y-axis.
<code>labelTitleFontSize</code>	The font size of the y-axis title. Defaults to 14.
<code>title</code>	Optional. The title of the plot.
<code>titleFontSize</code>	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.

line_chart	Create a line chart
------------	---------------------

Description

Create a line chart

Usage

```
line_chart(  
  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

<code>data</code>	The data frame containing the variables to consider.
<code>x</code>	The x-variable to consider. Must be a date variable in 'yyyy-mm-dd' format.
<code>y</code>	The y-variable to consider.
<code>curve</code>	The line's curve type to render. A complete list can be found here < https://github.com/d3/d3-shape#curves >. Defaults to 'curveLinear'.
<code>stroke</code>	The color of the line. Defaults to 'crimson'.
<code>strokeWidth</code>	The width of the line. Defaults to 1.5.
<code>xticks</code>	Optional. the number of x-axis ticks to consider.
<code>yticks</code>	Optional. The number of y-axis ticks to consider.
<code>xtitle</code>	Optional. The title of the x-axis.
<code>xtitleFontSize</code>	The font size of the x-axis title. Defaults to 16.
<code>ytitle</code>	Optional. The title of the y-axis.
<code>ytitleFontSize</code>	The font size of the y-axis title. Defaults to 16.
<code>title</code>	Optional. The title of the plot.
<code>titleFontSize</code>	The font size of the plot title. Defaults to 22.
<code>font</code>	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
<code>bgcol</code>	The background color of the SVG. Defaults to "#CAD0D3" HEX color.
<code>opacity</code>	The color opacity of the bars (from 0 to 1). Defaults to 1.
<code>axisCol</code>	the color of the x and y axis. It includes the ticks, the labels and titles. Defaults to 'black'.
<code>width</code>	Optional. The width of the SVG output.
<code>height</code>	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
line_chart(
  data = airpassengers,
  x = "date",
  y = "passengers"
)
```

liquid_chart	Create a liquid fill gauge chart.
--------------	-----------------------------------

Description

A liquid fill gauge represents a single value as a circle filled with an animated liquid up to the corresponding level. The fill rises smoothly on load, two wave layers give the water a natural depth, and the label text changes color at the water surface — appearing in `textColor` above the water and in `waveTextColor` below it.

Usage

```
liquid_chart(
  value,
  label = NULL,
  fillColor = "steelblue",
  circleColor = NULL,
  circleThickness = 0.05,
  textColor = NULL,
  waveTextColor = "white",
  textSize = 0.38,
  waveAmplitude = 0.09,
  waveCount = 2,
  waveSpeed = 1,
  title = NULL,
  titleFontSize = 14,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "white",
  width = NULL,
  height = NULL
)
```

Arguments

<code>value</code>	Numeric. The fill level, between 0 and 1 (0 = empty, 1 = full).
<code>label</code>	Character. Text displayed in the center of the gauge. Defaults to value formatted as a percentage (e.g. "55%").
<code>fillColor</code>	The color of the liquid. Defaults to "steelblue".
<code>circleColor</code>	The color of the outer ring. Defaults to <code>fillColor</code> .
<code>circleThickness</code>	Thickness of the outer ring as a fraction of the circle radius. Defaults to 0.05.
<code>textColor</code>	Color of the label text above the water surface. Defaults to <code>fillColor</code> .
<code>waveTextColor</code>	Color of the label text below (inside) the water surface. Defaults to "white".
<code>textSize</code>	Font size of the label as a fraction of the circle radius. Defaults to 0.38.
<code>waveAmplitude</code>	Height of the wave as a fraction of the circle radius. Defaults to 0.09.

waveCount	Number of full wave cycles visible across the circle. Defaults to 2.
waveSpeed	Speed of the horizontal wave animation. Higher values are faster. Defaults to 1.
title	Optional. A label displayed below the circle.
titleFontSize	Font size of the title in pixels. Defaults to 14.
font	The font family used for all text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	Background color of the SVG. Defaults to "white".
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An animated SVG liquid fill gauge.

Examples

```
# Basic usage - 55% fill
liquid_chart(value = 0.55)

# Custom color and label
liquid_chart(
  value      = 0.72,
  label      = "72%",
  fillColor  = "#e67e22",
  title      = "Disk usage"
)

# Low fill with a raw value label
liquid_chart(
  value      = 0.28,
  label      = "28%",
  fillColor  = "tomato",
  circleColor = "tomato",
  textColor  = "tomato",
  waveTextColor = "white"
)
```

lollipop_chart

Create a lollipop chart

Description

Create a lollipop chart

Usage

```

lollipop_chart(
  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

<code>data</code>	The data frame containing the variables to consider.
<code>x</code>	The categorical variable to consider. Will be plotted on the x-axis.
<code>y</code>	The numeric variable to consider. Will be plotted on the y-axis.
<code>sort</code>	Whether to sort or not the vertical lines. Takes three values 'none' which is the default, 'ascending' or 'descending'.
<code>bgcol</code>	The background-color of the SVG output. Defaults to 'white'.
<code>xticks</code>	Optional. the number of x-axis ticks to consider.
<code>yticks</code>	Optional. The number of y-axis ticks to consider.
<code>xFontSize</code>	the font size of the x-axis labels. Defaults to 10.
<code>yFontSize</code>	the font size of the y-axis labels. Defaults to 10.
<code>font</code>	The font family to consider for the titles. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
<code>xtitle</code>	Optional. The title of the x-axis.
<code>xtitleFontSize</code>	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)) %>%
  lollipop_chart(
    x = "drv",
    y = "median_cty",
    sort = "ascending",
    xtitle = "drv variable",
    ytitle = "median cty",
    title = "Median cty per drv"
  )
```

parliament_chart	<i>Display a Parliament Chart</i>
------------------	-----------------------------------

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

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

```

pie_chart

Create a pie chart

Description

Create a pie chart

Usage

```

pie_chart(
  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 $\text{padRadius} * \text{padAngle}$.
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

pie_chart(
  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.

pulse_grid

Create a pulse grid.

Description

‘pulse_grid()’ creates a tiled grid where each cell color reflects a numeric value and the cells gently pulse according to their intensity.

Usage

```

pulse_grid(
  data,
  x,
  y,
  value,
  low = "#E0FBFC",
  high = "#0B525B",
  stroke = "white",
  strokeWidth = 1,
  cellPadding = 0.12,
  cornerRadius = 10,
  pulse = TRUE,
  pulseStrength = 0.18,
  showValues = TRUE,
  digits = 1,
  valueFontSize = 12,
  xFontSize = 12,
  yFontSize = 12,
  title = NULL,
  titleFontSize = 22,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "#0F172A",
  labelColor = "#F8FAFC",
  titleColor = "#F8FAFC",
  width = NULL,
  height = NULL
)

```

Arguments

<code>data</code>	The data frame containing the grid coordinates and values.
<code>x</code>	The column name mapped to the grid columns.
<code>y</code>	The column name mapped to the grid rows.
<code>value</code>	The numeric column name mapped to tile intensity.
<code>low</code>	The low end color of the value scale. Defaults to <code>'#E0FBFC'</code> .
<code>high</code>	The high end color of the value scale. Defaults to <code>'#0B525B'</code> .
<code>stroke</code>	The border color of each tile. Defaults to <code>'white'</code> .
<code>strokeWidth</code>	Border width of each tile. Defaults to 1.
<code>cellPadding</code>	Space between tiles, from 0 to 0.9. Defaults to 0.12.
<code>cornerRadius</code>	Corner radius of each tile. Defaults to 10.
<code>pulse</code>	Logical. Whether to animate the tiles. Defaults to <code>'TRUE'</code> .
<code>pulseStrength</code>	Numeric multiplier controlling pulse amplitude. Defaults to 0.18.
<code>showValues</code>	Logical. Whether to print values inside tiles. Defaults to <code>'TRUE'</code> .

<code>digits</code>	Number of digits used for in-cell labels. Defaults to 1.
<code>valueFontSize</code>	Font size of in-cell labels. Defaults to 12.
<code>xFontSize</code>	Font size of the column labels. Defaults to 12.
<code>yFontSize</code>	Font size of the row labels. Defaults to 12.
<code>title</code>	Optional. The title of the plot.
<code>titleFontSize</code>	Font size of the plot title. Defaults to 22.
<code>font</code>	The font family used in labels and title. Defaults to 'Verdana, Geneva, Tahoma, sans-serif'.
<code>bgcol</code>	Background color of the SVG. Defaults to '#0F172A'.
<code>labelColor</code>	Color of the row and column labels. Defaults to '#F8FAFC'.
<code>titleColor</code>	Color of the title. Defaults to '#F8FAFC'.
<code>width</code>	Optional. The width of the SVG output.
<code>height</code>	Optional. The height of the SVG output.

Value

A SVG pulse grid.

Examples

```
pulse_data <- expand.grid(
  hour = c("09:00", "12:00", "15:00", "18:00"),
  team = c("North", "South", "East")
)
pulse_data$load <- c(0.35, 0.72, 0.56, 0.91,
                    0.41, 0.67, 0.83, 0.58,
                    0.28, 0.49, 0.77, 0.63)

pulse_grid(
  data = pulse_data,
  x = "hour",
  y = "team",
  value = "load",
  title = "Team Activity Pulse Grid",
  low = "#D9F0FF",
  high = "#EE6C4D",
  bgcol = "#0B132B"
)
```

scatter_plot	Create a scatter plot.
--------------	------------------------

Description

Create a scatter plot.

Usage

```
scatter_plot(  
  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

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

stacked_area_chart	Create a stacked area chart
--------------------	-----------------------------

Description

Create a stacked area chart

Usage

```
stacked_area_chart(  
  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
  )
)

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

waterfall_chart

Create an interactive waterfall chart.

Description

A waterfall chart shows how an initial value is built up or eroded by a series of positive and negative contributions to reach a final total. Each bar starts exactly where the previous one ended, making cumulative effects immediately visible. It is the standard chart for P&L decomposition, budget variance analysis, and any "what changed and why" narrative.

Hovering over a bar shows a tooltip with the label, the delta value, and the running total at that point.

Usage

```

waterfall_chart(
  data,
  x,
  y,
  measure = NULL,
  positiveColor = "#4CAF50",
  negativeColor = "#F44336",
  totalColor = "#5C85D6",
  connector = TRUE,
  connectorColor = "#aaaaaa",
  showLabels = TRUE,
  labelFontSize = 10,
  title = NULL,
  titleFontSize = 16,
  xtitle = NULL,
  xtitleFontSize = 13,
  ytitle = NULL,
  ytitleFontSize = 13,
  xFontSize = 11,
  yFontSize = 11,
  opacity = 1,
  font = "Verdana, Geneva, Tahoma, sans-serif",
  bgcol = "white",
  axisCol = "black",
  width = NULL,
  height = NULL
)

```

Arguments

<code>data</code>	The data frame containing the variables to consider.
<code>x</code>	The name of the column containing category labels.
<code>y</code>	The name of the column containing the delta values (positive for increases, negative for decreases).
<code>measure</code>	Optional. The name of a column that classifies each row as "relative" (default — a delta added to the running total) or "total" (a bar spanning from zero to the current running total, used for subtotals and the final total).
<code>positiveColor</code>	Fill color for bars representing an increase. Defaults to "#4CAF50".
<code>negativeColor</code>	Fill color for bars representing a decrease. Defaults to "#F44336".
<code>totalColor</code>	Fill color for "total" measure bars. Defaults to "#5C85D6".
<code>connector</code>	Logical. Whether to draw dashed connector lines between consecutive bars. Defaults to TRUE.
<code>connectorColor</code>	Color of the connector lines. Defaults to "#aaaaaa".
<code>showLabels</code>	Logical. Whether to display the delta value above or below each bar. Defaults to TRUE.

labelFontSize	Font size of the bar value labels. Defaults to 10.
title	Optional. Title of the chart.
titleFontSize	Font size of the chart title. Defaults to 16.
xtitle	Optional. Title of the x-axis.
xtitleFontSize	Font size of the x-axis title. Defaults to 13.
ytitle	Optional. Title of the y-axis.
ytitleFontSize	Font size of the y-axis title. Defaults to 13.
xFontSize	Font size of the x-axis tick labels. Defaults to 11.
yFontSize	Font size of the y-axis tick labels. Defaults to 11.
opacity	Opacity of the bars (0 to 1). Defaults to 1.
font	The font family used for all text. Defaults to "Verdana, Geneva, Tahoma, sans-serif".
bgcol	Background color of the SVG. Defaults to "white".
axisCol	Color of axis lines, ticks, and labels. Defaults to "black".
width	Optional. The width of the SVG output.
height	Optional. The height of the SVG output.

Value

An interactive SVG waterfall chart.

Examples

```
# P&L bridge: from revenue down to net income
pnl <- data.frame(
  label = c("Revenue", "COGS", "Gross Profit",
            "R&D", "S&M", "G&A", "Operating Income"),
  value = c(1200, -450, 750, -120, -90, -60, 480),
  measure = c("relative", "relative", "total",
              "relative", "relative", "relative", "total")
)

waterfall_chart(
  data = pnl,
  x = "label",
  y = "value",
  measure = "measure",
  title = "P&L bridge",
  ytitle = "USD thousands"
)

# Monthly cash flow
cashflow <- data.frame(
  month = c("Jan", "Feb", "Mar", "Apr", "May", "Jun"),
  delta = c(120, -30, 80, -60, 95, -20)
)
```

```
waterfall_chart(  
  data      = cashflow,  
  x         = "month",  
  y         = "delta",  
  title     = "Monthly cash flow",  
  ytitle    = "USD thousands",  
  positiveColor = "steelblue",  
  negativeColor = "tomato"  
)
```

Index

[anim_line_chart](#), 2
[animated_histogram](#), 4
[area_band](#), 6
[area_chart](#), 7

[bar_chart](#), 9
[bar_chart_race](#), 11
[beeswarm_plot](#), 13
[bullet_chart](#), 16

[dumbbell_chart](#), 18

[flame](#), 21
[flower](#), 22

[gauge_chart](#), 23
[glass_fill](#), 25

[heart_fill](#), 26
[histogram](#), 28
[horz_bar_chart](#), 29
[horz_lollipop](#), 31

[line_chart](#), 33
[liquid_chart](#), 35
[lollipop_chart](#), 36

[parliament_chart](#), 38
[pie_chart](#), 40
[plant_growth](#), 41
[pulse_grid](#), 42

[scatter_plot](#), 45
[stacked_area_chart](#), 46

[waterfall_chart](#), 48