Package: plotjs (via r-universe)

May 19, 2026

Type Package

Title An R 'graphics'-like Interface to 'Chart.js'

Version 0.3.0

Description A continuation of the author's earlier 'c3plot' project, create interactive plots with the 'Chart.js' <https://www.chartjs.org/> charting library using an interface modeled after that of base graphics. The main plotjs() function is an S3 generic with methods for various R objects that work similarly to their equivalent plot() methods. These plots can be embedded into 'Shiny' web applications or 'R Markdown' documents or displayed within the 'RStudio' 'Viewer' pane.

License GPL (>= 3)

URL https://arkraieski.github.io/plotjs/, https://github.com/arkraieski/plotjs

BugReports https://github.com/arkraieski/plotjs/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports htmlwidgets, grDevices

Suggests testthat, knitr, rmarkdown, gapminder, plotly, ggplot2, microbenchmark, dplyr

VignetteBuilder knitr

Config/pak/sysreqs cmake make libuv1-dev

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

Date/Publication 2026-04-19 01:43:10 UTC

RemoteUrl https://github.com/arkraieski/plotjs

RemoteRef HEAD

RemoteSha 1f29a1c6a4d9fe3c641a7b8bc587d8146fcbe240

Contents

`jsbarplot` *plotjs Bar Plots*

Description

Creates a bar plot using 'Chart.js'.

Usage

```
jsbarplot(
  heights,
  names.arg = NULL,
  col = NULL,
  main = NULL,
  ylab = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL,
  aria.label = NULL,
  ...
)
```

Arguments

`heights`	a vector of values describing the bars that make up the plot.
`names.arg`	a vector of names to be plotted below each bar.
`col`	a character string with the color for bars. This can be either a hex value or the name of an R built-in color.
`main`	a main title for the plot.
`ylab`	a label for the y axis.

width	width of the widget to create for the plot. The default is NULL, which results in automatic resizing based on the plot's container.
height	height of the widget to create for the plot. The default is NULL, which results in automatic resizing based on the plot's container.
elementId	Use an explicit element ID for the widget, rather than an automatically generated one.
aria.label	a character string set as the aria-label attribute on the chart's canvas element for accessibility.
...	arguments passed from methods.

`jsbarplot-shiny` *Shiny bindings for jsbarplot*

Description

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

Usage

```
jsbarplotOutput(outputId, width = "100%", height = "400px")

renderJsbarplot(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a jsbarplot
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

jspie *plotjs Pie Charts*

Description

Draw an interactive pie chart using 'Chart.js'

Usage

```
jspie(
  x,
  labels = names(x),
  col = NULL,
  slice.text = "pct",
  donut = FALSE,
  legend.position = "right",
  main = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL,
  aria.label = NULL
)
```

Arguments

x	a vector of non-negative numerical quantities. The values in x are displayed as the areas of pie slices.
labels	character vector giving names for the slices.
col	character vector of colors to be used in filling the slices. Can be a hex value or an R built-in color name.
slice.text	"pct" to display percentage-formatted values inside pie slices or "id" to display the slice's name from labels.
donut	logical; should the chart be rendered as a donut chart instead of a full pie?
legend.position	Position of the legend. Possible values are "right", "bottom", "inset", and "hide".
main	a main title for the plot.
width	width of the widget to create for the plot. The default is NULL, which results in automatic resizing based on the plot's container.
height	height of the widget to create for the plot. The default is NULL, which results in automatic resizing based on the plot's container.
elementId	Use an explicit element ID for the widget, rather than an automatically generated one.
aria.label	a character string set as the aria-label attribute on the chart's canvas element for accessibility.

Examples

```
pie.sales <- c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
names(pie.sales) <- c("Blueberry", "Cherry",
                      "Apple", "Boston Cream", "Other", "Vanilla Cream")
jspie(pie.sales)
jspie(pie.sales, col = c("purple", "violetred1", "green3",
                         "cornsilk", "cyan", "white"))
jspie(pie.sales, donut = TRUE)
```

`jspie-shiny` *Shiny bindings for jspie*

Description

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

Usage

```
jspieOutput(outputId, width = "100%", height = "400px")

renderJspie(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	output variable to read from
`width, height`	Must be a valid CSS unit (like `'100%'`, `'400px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`expr`	An expression that generates a jspie
`env`	The environment in which to evaluate `expr`.
`quoted`	Is `expr` a quoted expression (with `quote()`)? This is useful if you want to save an expression in a variable.

`plotjs` *Generic Plotting with 'Chart.js'*

Description

This is a generic function for plotting 'R' objects using the Chart.js charting library. The syntax is similar to that of the `plot()` generic function.

Usage

```
plotjs(x, ...)
```

Arguments

x	x coordinates for points or any 'R' object with a plotjs method.
...	arguments passed to methods.

Details

For simple scatter and line plots, plotjs.default will be used. However, there are plotjs methods for various 'R' objects. Use methods(plotjs) and the documentation for these.

Plots created with this are interactive htmlwidgets that can be used in the RStudio Viewer or embedded into 'R Markdown' documents, 'Shiny' web applications, etc.

For the default method, the argument y with the y coordinates of points is required. For some methods, such as plotjs.density and plotjs.function, y is not required because those methods can compute or extract y coordinates from the x object.

Examples

```
data(mtcars)
plotjs(mtcars$disp, mtcars$hp, main = "Displacement vs. HP in mtcars")
plotjs(qnorm)
```

---

plotjs-shiny *Shiny bindings for plotjs*

Description

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

Usage

```
plotjsOutput(outputId, width = "100%", height = "400px")

renderPlotjs(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a plotjsScatter
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

plotjs.default *The Default plotjs Scatterplot Function*

Description

Draw an interactive scatterplot or line plot using 'Chart.js'.

Usage

```
## Default S3 method:
plotjs(
  x,
  y,
  type = "p",
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  zoom = TRUE,
  col.group = NULL,
  col = NULL,
  legend.title = NULL,
  legend.position = "right",
  sci.x = FALSE,
  sci.y = FALSE,
  ...
)
```

Arguments

x	the x coordinates for the plot.
y	the y coordinates for the plot.
type	1-character string giving the type of plot desired. The following values are possible: "p" for points, "l" for lines, and "b" for both points and lines.
xlim	the x limits (x1, x2) of the plot.
ylim	the y limits of the plot.
main	a main title for the plot.
xlab	a label for the x axis, defaults to a description of x.
ylab	a label for the y axis, defaults to a description of y.
zoom	logical; should the zooming feature be enabled for the plot?
col.group	optionally, a factor the same length as x by which to group and color points.
col	The colors for the lines and points. If col.group is specified, this can be a vector of colors to use for each group in the data. If NULL, Chart.js default colors are used.

legend.title	a title for the legend. Defaults to a description of col.group if not set. You can also use legend.title = FALSE to suppress the legend title.
legend.position	Position of the plot legend. Possible values are "right", "bottom", "inset", and "hide". This is ignored if all points are colored the same.
sci.x	logical indicating whether scientific notation should be used for the x-axis.
sci.y	logical indicating whether scientific notation should be used for the y-axis.
...	arguments passed to htmlwidgets::createWidget(): width, height, and elementId. These arguments default to NULL. You may also pass aria.label, a character string set as the aria-label attribute on the chart's canvas element for accessibility.

Examples

```
data(mtcars)
plotjs(mtcars$hp, mtcars$qsec)
plotjs(mtcars$disp, mtcars$hp, main = "Displacement vs. HP in mtcars")
```

---

plotjs.density *plotjs Method for Kernel Density Estimation*

Description

The plotjs method for density objects.

Usage

```
## S3 method for class 'density'
plotjs(x, main = NULL, xlab = NULL, ylab = "Density", type = "l", ...)
```

Arguments

x	a "density" object
main	a main title for the plot.
xlab	label for the x axis
ylab	a label for the y axis, defaults to a description of y.
type	1-character string giving the type of plot desired. The following values are possible: "p" for points, "l" for lines, and "b" for both points and lines.
...	arguments passed to other methods

plotjs.factor *Factor Variable Bar Plots*

Description

This function implements a bar plot method for `factor` arguments to the `plotjs` generic function. Bar heights will be counts for the factor levels.

Usage

```
## S3 method for class 'factor'
plotjs(x, ylab = "Count", ...)
```

Arguments

`x`	a factor.
`ylab`	a label for the y axis.
`...`	arguments passed to `jsbarplot`.

See Also

`jsbarplot()`

Examples

```
mtcars <- mtcars
mtcars$cyl <- as.factor(mtcars$cyl)
plotjs(mtcars$cyl)
```

plotjs.function *Draw Function Plots with 'Chart.js'*

Description

Draws a curve corresponding to a function over the interval `[from, to]` in an interactive plot using 'Chart.js'.

Usage

```
## S3 method for class '`function`'
plotjs(x, from = 0, to = 1, ylab = NULL, ...)
```

Arguments

x	The name of a function.
from	the lower limit of the range over which the function will be plotted.
to	the upper limit of the range over which the function will be plotted.
ylab	a label for the y axis, defaults to the name of the function specified in x.
...	arguments passed to other methods

Examples

```
plotjs(qnorm)
plotjs(sin, -pi, 2 * pi)
```

---

plotjs.lm *plotjs Diagnostics for an lm Object*

Description

A plot of residuals against fitted values.

Usage

```
## S3 method for class 'lm'
plotjs(x, which = 1, ...)
```

Arguments

x	an "lm" object
which	1 is the only supported value currently. This argument is included for consistency with the base plot.lm() method.
...	arguments passed to other methods.

Examples

```
lm.SR <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)
plotjs(lm.SR)
```

Index