Package 'dygraphs'

May 8, 2026

Type Package

Title Interface to 'Dygraphs' Interactive Time Series Charting Library

Version 1.1.1.6

URL https://github.com/rstudio/dygraphs

BugReports https://github.com/rstudio/dygraphs/issues

Description An R interface to the 'dygraphs' JavaScript charting library (a copy of which is included in the package). Provides rich facilities for charting time-series data in R, including highly configurable series- and axis-display and interactive features like zoom/pan and series/point highlighting.

License MIT + file LICENSE

Depends R (>= 3.0)

Imports magrittr, htmlwidgets (>= 0.6), htmltools (>= 0.3.5), zoo (>= 1.7-10), xts (>= 0.9-7)

Suggests testthat

Enhances rmarkdown (>= 0.3.3), shiny (>= 0.10.2.1)

RoxygenNote 6.0.1

NeedsCompilation no

Author Dan Vanderkam [aut, cph] (dygraphs library in htmlwidgets/lib, http://dygraphs.com/),
Petr Shevtsov [cre, cph],
JJ Allaire [aut],
RStudio [cph],
Jonathan Owen [aut, cph],
Daniel Gromer [aut, cph],
Benoit Thieurmel [aut, cph],
Kent Laukhuf [ctb],
jQuery Foundation [cph] (jQuery library),
jQuery contributors [ctb, cph] (jQuery library; authors listed in inst/htmlwidgets/lib/jquery/AUTHORS.txt)

Maintainer Petr Shevtsov <petr.shevtsov@gmail.com>

Repository CRAN
Date/Publication 2018-07-11 22:00:03 UTC

Contents

`dyAnnotation` *Annotation for dygraph chart*

Description

Define a text annotation for a data-point on a dygraph chart.

Usage

```
dyAnnotation(dygraph, x, text, tooltip = NULL, width = NULL,
  height = NULL, cssClass = NULL, tickHeight = NULL,
  attachAtBottom = FALSE, clickHandler = NULL, mouseOverHandler = NULL,
  mouseOutHandler = NULL, dblClickHandler = NULL, series = NULL)
```

Arguments

dygraph	Dygraph to add an annotation to
x	Either numeric or date value indicating where to place the annotation. For date value, this should be of class POSIXct or convertible to POSIXct.
text	Text to overlay on the chart at the location of x
tooltip	Additional tooltip text to display on mouse hover
width	Width (in pixels) of the annotation flag.
height	Height (in pixels) of the annotation flag.
cssClass	CSS class to use for styling the annotation.
tickHeight	Height of the tick mark (in pixels) connecting the point to its flag or icon.
attachAtBottom	If true, attach annotations to the x-axis, rather than to actual points.
clickHandler	JavaScript function to call when an annotation is clicked.
mouseOverHandler	JavaScript function to call when the mouse hovers over an annotation.
mouseOutHandler	JavaScript function to call when the mouse exits an annotation.
dblClickHandler	JavaScript function to call when an annotation is double clicked.
series	Series to attach the annotation to. By default, the last series defined using dySeries.

Value

Dygraph with specified annotation

Note

Annotations are bound to specific series in the input data. If you have only one series or if you define annotations immediately after a call to dySeries then you need not specify the series explicitly. Otherwise, you should use the series parameter to indicate which series the annotation should be bound to.

Annotation event handlers can also specified globally (see dyCallbacks).

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(presidents, main = "Presidential Approval") %>%
  dyAxis("y", valueRange = c(0, 100)) %>%
  dyAnnotation("1950-7-1", text = "A", tooltip = "Korea") %>%
  dyAnnotation("1965-1-1", text = "B", tooltip = "Vietnam")
```

dyAxis *dygraph axis*

Description

Define options for an axis on a dygraph plot. Note that options will use the default global setting (as determined by dyOptions) when not specified explicitly.

Usage

```
dyAxis(dygraph, name, label = NULL, valueRange = NULL, logscale = NULL,
  ticker = NULL, rangePad = NULL, labelWidth = NULL, labelHeight = NULL,
  axisHeight = NULL, axisLineColor = NULL, axisLineWidth = NULL,
  pixelsPerLabel = NULL, axisLabelColor = NULL, axisLabelFontSize = NULL,
  axisLabelWidth = NULL, axisLabelFormatter = NULL, valueFormatter = NULL,
  drawGrid = NULL, gridLineColor = NULL, gridLineWidth = NULL,
  independentTicks = NULL)
```

Arguments

dygraph	Dygraph to add an axis definition to
name	Axis name ('x', 'y', or 'y2')
label	Label to display for axis (defaults to none).
valueRange	Explicitly set the vertical range of the graph to c(low, high). This may be set on a per-axis basis to define each y-axis separately. If either limit is unspecified, it will be calculated automatically (e.g. c(NULL, 30) to automatically calculate just the lower bound).
logscale	When set for the y-axis or x-axis, the graph shows that axis in log scale. Any values less than or equal to zero are not displayed. Showing log scale with ranges that go below zero will result in an unviewable graph. Not compatible with showZero. connectSeparatedPoints is ignored. This is ignored for date-based x-axes.
ticker	This lets you specify an arbitrary JavaScript function to generate tick marks on an axis. The tick marks are an array of (value, label) pairs. The built-in functions go to great lengths to choose good tick marks so, if you set this option, you'll most likely want to call one of them and modify the result. See dygraph-tickers.js and the dygraphs documentation for additional details).
rangePad	Add the specified amount of extra space (in pixels) around the value range to ensure points at the edges remain visible.
labelWidth	Width of the div which contains the y-axis label. Since the y-axis label appears rotated 90 degrees, this actually affects the height of its div.
labelHeight	Height of the x-axis label, in pixels. This also controls the default font size of the x-axis label. If you style the label on your own, this controls how much space is set aside below the chart for the x-axis label's div.

`axisHeight` Height, in pixels, of the x-axis. If not set explicitly, this is computed based on `axisLabelFontSize` and `axisTickSize`.

`axisLineColor` Color of the x- and y-axis lines. Accepts any value which the HTML canvas strokeStyle attribute understands, e.g. 'black' or 'rgb(0, 100, 255)'.

`axisLineWidth` Thickness (in pixels) of the x- and y-axis lines.

`pixelsPerLabel` Number of pixels to require between each x- and y-label. Larger values will yield a sparser axis with fewer ticks. Defaults to 50 (x-axis) or 30 (y-axes).

`axisLabelColor` Color for x- and y-axis labels. This is a CSS color string. This may also be set globally using `dyOptions`.

`axisLabelFontSize`
Size of the font (in pixels) to use in the axis labels, both x- and y-axis. This may also be set globally using `dyOptions`.

`axisLabelWidth` Width, in pixels, of the axis labels

`axisLabelFormatter`
JavaScript function to call to format the tick values that appear along an axis (see the dygraphs documentation for additional details).

`valueFormatter` JavaScript function to call to provide a custom display format for the values displayed on mouseover (see the dygraphs documentation for additional details).

`drawGrid` Whether to display grid lines in the chart.

`gridLineColor` The color of the grid lines.

`gridLineWidth` Thickness (in pixels) of the grid lines drawn under the chart.

`independentTicks`
Only valid for y and y2, has no effect on x: This option defines whether the y axes should align their ticks or if they should be independent. Possible combinations: 1.) y=true, y2=false (default): y is the primary axis and the y2 ticks are aligned to the the ones of y. (only 1 grid) 2.) y=false, y2=true: y2 is the primary axis and the y ticks are aligned to the the ones of y2. (only 1 grid) 3.) y=true, y2=true: Both axis are independent and have their own ticks. (2 grids) 4.) y=false, y2=false: Invalid configuration causes an error.

Value

Axis options

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyAxis("y", label = "Temp (F)", valueRange = c(40, 60)) %>%
  dyOptions(axisLineWidth = 1.5, fillGraph = TRUE, drawGrid = FALSE)
```

dyCallbacks *Callbacks for dygraph events*

Description

Set JavaScript callbacks for various dygraph events. See the dygraph options reference for additional details on the signature of each callback.

Usage

```
dyCallbacks(dygraph, clickCallback = NULL, drawCallback = NULL,
  highlightCallback = NULL, pointClickCallback = NULL,
  underlayCallback = NULL, unhighlightCallback = NULL,
  zoomCallback = NULL, drawHighlightPointCallback = NULL,
  drawPointCallback = NULL, annotationClickHandler = NULL,
  annotationMouseOverHandler = NULL, annotationMouseOutHandler = NULL,
  annotationDblClickHandler = NULL)
```

Arguments

dygraph	Dygraph to add callbacks to
clickCallback	A function to call when the canvas is clicked.
drawCallback	When set, this callback gets called every time the dygraph is drawn. This includes the initial draw, after zooming and repeatedly while panning.
highlightCallback	When set, this callback gets called every time a new point is highlighted.
pointClickCallback	A function to call when a data point is clicked. and the point that was clicked.
underlayCallback	When set, this callback gets called before the chart is drawn.
unhighlightCallback	When set, this callback gets called every time the user stops highlighting any point by mousing out of the graph.
zoomCallback	A function to call when the zoom window is changed (either by zooming in or out).
drawHighlightPointCallback	Draw a custom item when a point is highlighted. Default is a small dot matching the series color. This method should constrain drawing to within pointSize pixels from (cx, cy)
drawPointCallback	Draw a custom item when drawPoints is enabled. Default is a small dot matching the series color. This method should constrain drawing to within pointSize pixels from (cx, cy).

annotationClickHandler
: JavaScript function to call when an annotation is clicked. This can also be specified on a per-annotation basis.

annotationMouseOverHandler
: JavaScript function to call when the mouse hovers over an annotation. This can also be specified on a per-annotation basis.

annotationMouseOutHandler
: JavaScript function to call when the mouse exits an annotation. This can also be specified on a per-annotation basis.

annotationDblClickHandler
: JavaScript function to call when an annotation is double clicked. This can also be specified on a per-annotation basis.

Value

Dygraph with callbacks

`dyCandlestick` *Employ a dygraph plotter on a series, a group of series, or the whole dygraph*

Description

Plotters provide variuos ways to customize how your data appears on the dygraph. Series-based plotters allow users to mix-and-match different plotters on a per-series or (with dyGroup) a per-group basis. See `dyPlotter` for additional detail.

Usage

```
dyCandlestick(dygraph, compress = FALSE)

dyBarChart(dygraph)

dyStackedBarChart(dygraph)

dyMultiColumn(dygraph)

dyBarSeries(dygraph, name, ...)

dyStemSeries(dygraph, name, ...)

dyShadow(dygraph, name, ...)

dyFilledLine(dygraph, name, ...)

dyErrorFill(dygraph, name, ...)
```

```
dyMultiColumnGroup(dygraph, name, ...)

dyCandlestickGroup(dygraph, name, ...)

dyStackedBarGroup(dygraph, name, ...)

dyStackedLineGroup(dygraph, name, ...)

dyStackedRibbonGroup(dygraph, name, ...)
```

Arguments

dygraph	Dygraph to add plotter to
compress	(For dyCandlestick) If true, compress data yearly, quarterly, monthly, weekly or daily according to overall amount of bars and/or current zoom level.
name	name - or chrarcter vector of names - of (the) series within the data set
...	additional options to pass to dySeries

Value

A dygraph with the specified plotter(s) employed.

Available plotters

Currently the dygraphs package provides the following plotters:

dyBarChart() Draws a bar plot rather than a line plot. If the provided dygraph features more than one series, dyBarChart will call dyMultiColumn instead.

dyStackedBarChart() Draws a bar chart stacking all the underlying series.

dyMultiColumn() Draws multiple column bar chart.

dyBarSeries() Draws a single set of bars for just the provided series.

dyStemSeries() Draws a single set of stems for just the provided series.

dyShadow() An extraction of the _fillplotter from dygraph-combined-dev.js, drawing the filled area without the line.

dyFilledLIne() An extraction of the _fillplotter and _lineplotter combo from dygraph-combined-dev.js. dyFilledLine allows users to fill only a single series.

dyMultiColumnGroup() The multicolumn plotter, but on a subset of the series, leaving the others for other plotters.

dyCandlestick() Draw a candlestick chart.

dyCandleStickGroup() Employed on the provided series, but still plotting the others.

dyStackerBarGroup() Return the data group as stacked bars

dyStackerRibbonGroup() Return the data group as stacked ribbons

Examples

```
## The following two examples will results in the same dygraph:

dygraph(mdeaths) %>%
  dyBarChart()

lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths) %>%
  dyMultiColumn()


## Per-series plotters:

lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths) %>%
  dyBarSeries('fdeaths')

lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths) %>%
  dyStemSeries('fdeaths')

lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths) %>%
  dyShadow('fdeaths')

lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths) %>%
  dyFilledLine('fdeaths')

## A bunch of different plotters together:

lungDeaths <- cbind(fdeaths, mdeaths, ldeaths, foo = fdeaths/2, bar = fdeaths/3)
dygraph(lungDeaths) %>%
  dyRangeSelector() %>%
  dyBarSeries('bar') %>%
  dyStemSeries('mdeaths') %>%
  dyShadow('foo') %>%
  dyFilledLine('fdeaths')

## Group-based plotters:


## Candlestick plotters:

library(xts)
data(sample_matrix)
library(dygraphs)
dygraph(sample_matrix) %>%
  dyCandlestick()

sample<-data.frame(sample_matrix)
sample_2<-sample*2
```

```
names(sample_2)<-c('O', 'H', 'L', 'C')
sample<-cbind(sample, sample_2)
dygraph(sample) %>%
  dyOptions(stackedGraph = TRUE) %>%
  dyCandlestickGroup(c('Open', 'High', 'Low', 'Close')) %>%
  dyCandlestickGroup(c('O', 'H', 'L', 'C'))

## Stacked Bar and Ribbon Graphs:

dygraph(lungDeaths) %>%
  dySeries('mdeaths', axis = 'y2') %>%
  dyAxis('y', valueRange = c(-100, 1000)) %>%
  dyStackedBarGroup(c('ldeaths', 'fdeaths'))

lungDeaths <- cbind(ldeaths, fdeaths, mdeaths,
                    additive = rep.int(200, length(ldeaths)),
                    line = rep.int(3000, length(ldeaths)))
dygraph(lungDeaths) %>%
  dySeries('line', strokePattern = 'dashed') %>%
  dySeries('ldeaths', stepPlot = TRUE) %>%
  dyStackedBarGroup(c('additive', 'mdeaths')) %>%
  dyStackedRibbonGroup(c('fdeaths', 'ldeaths'))
```

`dyCrosshair` *The dyCrosshair plugin draws a crosshair line over the point closest to the mouse when the user hovers over the graph. It has a "direction" option which is provided in the R wrapper function and then forwarded to the plugin using the "options" argument to dyPlugin.*

Description

The dyCrosshair plugin draws a crosshair line over the point closest to the mouse when the user hovers over the graph. It has a "direction" option which is provided in the R wrapper function and then forwarded to the plugin using the "options" argument to dyPlugin.

Usage

```
dyCrosshair(dygraph, direction = c("both", "horizontal", "vertical"))
```

Arguments

`dygraph`	Dygraph to add plugin to
`direction`	Crosshair direction. Valid options are: "both", "horizontal", "vertical"

Value

Dygraph with Crosshair plugin enabled

Examples

```
library(dygraphs)
dygraph(ldeaths) %>%
  dyRangeSelector() %>%
  dyCrosshair(direction = "vertical")
```

---

dyCSS *CSS for dygraph labels and legend*

Description

Apply custom CSS to the text drawn within a dygraph. See the CSS documentation on the dygraphs website for additional details on which styles are available.

Usage

```
dyCSS(dygraph, css)
```

Arguments

dygraph	Dygraph to add CSS styles to
css	Path to css file to be used for styling textual elements of the graph.

Value

dygraph with additional CSS styles

Note

See the CSS Styling article on the dygraphs for R website for additional details.

Note that CSS styles are global so will affect all dygraphs on a given web page. This also implies that for a page with multiple plots you only need to specify styles for the first one (alternatively you can just add them directly to the page by other means).

dyDataHandler *Include a dygraph data handler*

Description

Include a dygraph data handler

Usage

```
dyDataHandler(dygraph, name, path, version = "1.0")
```

Arguments

dygraph	Dygraph to add data handler to
name	Name of data handler
path	Path to data handler JavaScript file
version	Data handler version (e.g. version of package which provides the data handler)

Value

A dygraph with the specified data handler enabled.

dyDependency *Add external assets as a dygraph dependency*

Description

Add external assets as a dygraph dependency

Usage

```
dyDependency(dygraph, dependency)
```

Arguments

dygraph	Dygraph to add dependency to
dependency	An HTML dependency

Value

A dygraph with the specified dependency added.

dyEvent *dygraph event line*

Description

Add a vertical event line to a dygraph

Usage

```
dyEvent(dygraph, x, label = NULL, labelLoc = c("top", "bottom"),
  color = "black", strokePattern = "dashed", date)
```

Arguments

dygraph	Dygraph to add event line to
x	Either numeric or date/time for the event, depending on the format of the x-axis of the dygraph. (For date/time must be a POSIXct object or another object convertible to POSIXct via as.POSIXct)
label	Label for event. Defaults to blank.
labelLoc	Location for label (top or bottom)
color	Color of event line. This can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow". Defaults to black.
strokePattern	A predefined stroke pattern type ("dotted", "dashed", "dotdash", or "solid") or a custom pattern array where the even index is a draw and odd is a space in pixels. Defaults to dashed.
date	(deprecated) See argument x.

Value

A dygraph with the specified event line.

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(presidents, main = "Presidential Approval") %>%
  dyAxis("y", valueRange = c(0, 100)) %>%
  dyEvent("1950-6-30", "Korea", labelLoc = "bottom") %>%
  dyEvent("1965-2-09", "Vietnam", labelLoc = "bottom")
dygraph(presidents, main = "Presidential Approval") %>%
  dyAxis("y", valueRange = c(0, 100)) %>%
  dyEvent(c("1950-6-30", "1965-2-09"), c("Korea", "Vietnam"), labelLoc = "bottom")
```

dygraph *dygraph interactive plot for time series data*

Description

R interface to interactive time series plotting using the dygraphs JavaScript library.

Usage

```
dygraph(data, main = NULL, xlab = NULL, ylab = NULL, periodicity = NULL,
  group = NULL, elementId = NULL, width = NULL, height = NULL)
```

Arguments

data	Either time series data or numeric data. For time series, this must be an xts object or an object which is convertible to xts. For numeric data, this must be a named list or data frame, where the first element/column provides x-axis values and all subsequent elements/columns provide one or more series of y-values.
main	Main plot title (optional)
xlab	X axis label
ylab	Y axis label
periodicity	Periodicity of time series data (automatically detected via xts::periodicity if not specified).
group	Group to associate this plot with. The x-axis zoom level of plots within a group is automatically synchronized.
elementId	Use an explicit element ID for the widget (rather than an automatically generated one). Useful if you have other JavaScript that needs to explicitly discover and interact with a specific widget instance.
width	Width in pixels (optional, defaults to automatic sizing)
height	Height in pixels (optional, defaults to automatic sizing)

Value

Interactive dygraph plot

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)
lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths)

indoConc <- Indometh[Indometh$Subject == 1, c("time", "conc")]
dygraph(indoConc)
```

---

dygraph-shiny *Shiny bindings for dygraph*

Description

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

Usage

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

renderDygraph(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 dygraph
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.

---

dygraphs-exports *dygraph exported operators and S3 methods*

Description

The following functions are imported and then re-exported from the dygraphs package to avoid listing the magrittr and zoo packages as Depends of dygraphs.

dyGroup *dygraph series group*

Description

Add a data series group to a dygraph plot. Note that options will use the default global setting (as determined by dyOptions) when not specified explicitly. Importantly, any dySeries options passed can be passed as a vector of values and will be replicated across all series named as part of the group. If arguments differ in length than the number of series named, then the argument vector will be cycled across the named series.

Usage

```
dyGroup(dygraph, name = NULL, label = NULL, color = NULL, axis = "y",
  stepPlot = NULL, stemPlot = NULL, fillGraph = NULL, drawPoints = NULL,
  pointSize = NULL, pointShape = NULL, strokeWidth = NULL,
  strokePattern = NULL, strokeBorderWidth = NULL,
  strokeBorderColor = NULL, plotter = NULL)
```

Arguments

dygraph	Dygraph to add a series definition to
name	character vector of the series within data set. If no name is specified then series are bound to implicitly based on their order within the underlying time series object. This parameter can also be a character vector of length 3 that specifies a set of input column names to use as the lower, value, and upper for a series with a shaded bar drawn around it.
label	Labels to display for series (uses name if no label defined)
color	Color for series. These can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow", etc. If not specified then the global colors option (typically based on equally-spaced points around a color wheel).
axis	Y-axis to associate the series with ("y" or "y2")
stepPlot	When set, display the graph as a step plot instead of a line plot.
stemPlot	When set, display the graph as a stem plot instead of a line plot.
fillGraph	Should the area underneath the graph be filled? This option is not compatible with error bars.
drawPoints	Draw a small dot at each point, in addition to a line going through the point. This makes the individual data points easier to see, but can increase visual clutter in the chart.
pointSize	The size of the dot to draw on each point in pixels. A dot is always drawn when a point is "isolated", i.e. there is a missing point on either side of it. This also controls the size of those dots.
pointShape	The shape of the dot to draw. Can be one of the following: "dot" (default), "triangle", "square", "diamond", "pentagon", "hexagon", "circle", "star", "plus" or "ex".

`strokeWidth` The width of the lines connecting data points. This can be used to increase the contrast or some graphs.

`strokePattern` A predefined stroke pattern type ("dotted", "dashed", or "dotdash") or a custom pattern array where the even index is a draw and odd is a space in pixels. If `NULL` then it draws a solid line. The array should have an even length as any odd length array could be expressed as a smaller even length array.

`strokeBorderWidth`
Draw a border around graph lines to make crossing lines more easily distinguishable. Useful for graphs with many lines.

`strokeBorderColor`
Color for the line border used if `strokeBorderWidth` is set.

`plotter` A function which plots the series group. See the dygraphs documentation for additional details on plotting functions.

Details

NOTE: dyGroup will turn off `stackedGraph`, as the option will calculated cumulatives using all series in the underlying dygraph, not just a subset.

The dyGroup function can also replicated similar arguments across multiple series, or be used to apply a grouped custom plotter - i.e., multi-column plotter - to a subset of the dygraph data series.

Value

Dygraph with additional series

Note

See the online documentation for additional details and examples.

Examples

```
## Not run:
library(dygraphs)

lungDeaths <- cbind(ldeaths, mdeaths, fdeaths)

dygraph(lungDeaths, main = "Deaths from Lung Disease (UK)") %>%
  dySeries("fdeaths", stepPlot = TRUE, color = "red") %>%
  dyGroup(c("mdeaths", "ldeaths"), drawPoints = TRUE, color = c("blue", "green"))

## End(Not run)
```

dyHighlight *dygraph series mouse-over highlighting*

Description

Configure options for data series mouse-over highlighting. Note that highlighting is always enabled for dygraphs so this function is used to customize rather than enable highlighting.

Usage

```
dyHighlight(dygraph, highlightCircleSize = 3,
  highlightSeriesBackgroundAlpha = 0.5, highlightSeriesOpts = list(),
  hideOnMouseOut = TRUE)
```

Arguments

dygraph	Dygraph to configure highlighting behavior for.
highlightCircleSize	The size in pixels of the dot drawn over highlighted points.
highlightSeriesBackgroundAlpha	Fade the background while highlighting series. 1=fully visible background (disable fading), 0=hidden background (show highlighted series only).
highlightSeriesOpts	When set, the options from this list are applied to the series closest to the mouse pointer for interactive highlighting. Example: list(strokeWidth = 3). See the documentation on dySeries for additional details on options that can be set.
hideOnMouseOut	Whether to hide the highlighting effects when the mouse leaves the chart area. Note that this also affects the hiding of the dyLegend on mouse out.

Value

A dygraph with customized highlighting options

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)
lungDeaths <- cbind(ldeaths, mdeaths, fdeaths)
dygraph(lungDeaths, main = "Deaths from Lung Disease (UK)") %>%
  dyHighlight(highlightCircleSize = 5,
              highlightSeriesBackgroundAlpha = 0.2,
              hideOnMouseOut = FALSE)
```

dyLegend *dygraph legend*

Description

Configure options for the dygraph series legend.

Usage

```
dyLegend(dygraph, show = c("auto", "always", "onmouseover", "follow",
  "never"), width = 250, showZeroValues = TRUE, labelsDiv = NULL,
  labelsSeparateLines = FALSE, hideOnMouseOut = TRUE)
```

Arguments

dygraph	Dygraph to configure legend options for.
show	When to display the legend. Specify "always" to always show the legend. Specify "onmouseover" to only display it when a user mouses over the chart. Specify "follow" to have the legend show as overlay to the chart which follows the mouse. The default behavior is "auto", which results in "always" when more than one series is plotted and "onmouseover" when only a single series is plotted.
width	Width (in pixels) of the div which shows the legend.
showZeroValues	Show zero value labels in the legend.
labelsDiv	Show data labels in an external div, rather than on the graph. This value should be a div element id.
labelsSeparateLines	Put a between lines in the label string. Often used in conjunction with labelsDiv.
hideOnMouseOut	Whether to hide the legend when the mouse leaves the chart area. This option applies when show is set to "onmouseover". Note that this also affects the hiding of the dyHighlight on mouse out.

Value

A dygraph with customized legend options

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dySeries("V1", label = "Temperature (F)") %>%
  dyLegend(show = "always", hideOnMouseOut = FALSE)
```

`dyLimit` *dygraph limit line*

Description

Add a horizontal limit line to a dygraph

Usage

```
dyLimit(dygraph, limit, label = NULL, labelLoc = c("left", "right"),
  color = "black", strokePattern = "dashed")
```

Arguments

`dygraph`	Dygraph to add limit line to
`limit`	Numeric position of limit.
`label`	Label for limit. Defaults to blank.
`labelLoc`	Location for label (left or right).
`color`	Color of limit line. This can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow". Defaults to black.
`strokePattern`	A predefined stroke pattern type ("dotted", "dashed", "dotdash", or "solid") or a custom pattern array where the even index is a draw and odd is a space in pixels. Defaults to dashed.

Value

A dygraph with the specified limit line.

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(presidents, main = "Presidential Approval") %>%
  dyAxis("y", valueRange = c(0, 100)) %>%
  dyLimit(max(presidents, na.rm = TRUE), "Max",
          strokePattern = "solid", color = "blue")
```

dyOptions *dygraph options*

Description

Add options to a dygraph plot.

Usage

```
dyOptions(dygraph, stackedGraph = FALSE, fillGraph = FALSE,
  fillAlpha = 0.15, stepPlot = FALSE, stemPlot = FALSE,
  drawPoints = FALSE, pointSize = 1, pointShape = c("dot", "triangle",
  "square", "diamond", "pentagon", "hexagon", "circle", "star", "plus", "ex"),
  drawGapEdgePoints = FALSE, connectSeparatedPoints = FALSE,
  strokeWidth = 1, strokePattern = NULL, strokeBorderWidth = NULL,
  strokeBorderColor = "white", plotter = NULL, colors = NULL,
  colorValue = 0.5, colorSaturation = 1, drawXAxis = TRUE,
  drawYAxis = TRUE, includeZero = FALSE, drawAxesAtZero = FALSE,
  logscale = FALSE, axisTickSize = 3, axisLineColor = "black",
  axisLineWidth = 0.3, axisLabelColor = "black", axisLabelFontSize = 14,
  axisLabelWidth = 60, drawGrid = TRUE, gridLineColor = NULL,
  gridLineWidth = 0.3, titleHeight = NULL, rightGap = 5,
  digitsAfterDecimal = 2, labelsKMB = FALSE, labelsKMG2 = FALSE,
  labelsUTC = FALSE, maxNumberWidth = 6, sigFigs = NULL,
  panEdgeFraction = NULL, animatedZooms = FALSE,
  mobileDisableYTouch = TRUE, timingName = NULL, useDataTimezone = FALSE,
  retainDateWindow = FALSE, disableZoom = FALSE)
```

Arguments

dygraph	Dygraph to add options to
stackedGraph	If set, stack series on top of one another rather than drawing them independently. The first series specified in the input data will wind up on top of the chart and the last will be on bottom.
fillGraph	Should the area underneath the graph be filled? This option is not compatible with error bars. This option can also be set on a per-series basis.
fillAlpha	Transparency for filled regions of the plot. A value of 0.0 means that the fill will not be drawn, whereas a value of 1.0 means that the fill will be as dark as the line of the series itself.
stepPlot	When set, display the graph as a step plot instead of a line plot. This option can also be set on a per-series basis.
stemPlot	When set, display the graph as a stem plot instead of a line plot. This option can also be set on a per-series basis.
drawPoints	Draw a small dot at each point, in addition to a line going through the point. This makes the individual data points easier to see, but can increase visual clutter in the chart. This option can also be set on a per-series basis.

`pointSize` The size of the dot to draw on each point in pixels. A dot is always drawn when a point is "isolated", i.e. there is a missing point on either side of it. This also controls the size of those dots. This option can also be set on a per-series basis.

`pointShape` The shape of the dot to draw. Can be one of the following: "dot" (default), "triangle", "square", "diamond", "pentagon", "hexagon", "circle", "star", "plus" or "ex". This option can also be set on a per-series basis.

`drawGapEdgePoints`
Draw points at the edges of gaps in the data. This improves visibility of small data segments or other data irregularities.

`connectSeparatedPoints`
Usually, when dygraphs encounters a missing value in a data series, it interprets this as a gap and draws it as such. If, instead, the missing values represents an x-value for which only a different series has data, then you'll want to connect the dots by setting this to true.

`strokeWidth` The width of the lines connecting data points. This can be used to increase the contrast or some graphs. This option can also be set on a per-series basis.

`strokePattern` A custom pattern array where the even index is a draw and odd is a space in pixels. If null then it draws a solid line. The array should have a even length as any odd length array could be expressed as a smaller even length array. This is used to create dashed lines. This option can also be set on a per-series basis.

`strokeBorderWidth`
Draw a border around graph lines to make crossing lines more easily distinguishable. Useful for graphs with many lines. This option can also be set on a per-series basis.

`strokeBorderColor`
Color for the line border used if `strokeBorderWidth` is set. This option can also be set on a per-series basis.

`plotter` A function (or array of functions) which plot each data series on the chart. May also be set on a per-series basis. See the dygraphs documentation for additional details on plotting functions.

`colors` Character vector of colors for the data series. These can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow", etc. If not specified, equally-spaced points around a color wheel are used. This option can also be set on a per-series basis. Note that in both global and per-series specification of custom colors you must provide a color for all series being displayed. Note also that global and per-series color specification cannot be mixed.

`colorValue` If custom colors are not specified, value of the data series colors, as in hue/saturation/value (0.0-1.0, default 0.5).

`colorSaturation`
If custom colors are not specified, saturation of the automatically-generated data series colors (0.0-1.0, default 0.5).

`drawXAxis` Whether to draw the x-axis. Setting this to false also prevents x-axis ticks from being drawn and reclaims the space for the chart grid/lines.

`drawYAxis` Whether to draw the y-axis. Setting this to false also prevents y-axis ticks from being drawn and reclaims the space for the chart grid/lines.

`includeZero` Usually, dygraphs will use the range of the data plus some padding to set the range of the y-axis. If this option is set, the y-axis will always include zero, typically as the lowest value. This can be used to avoid exaggerating the variance in the data.

`drawAxesAtZero` When set, draw the X axis at the Y=0 position and the Y axis at the X=0 position if those positions are inside the graph's visible area. Otherwise, draw the axes at the bottom or left graph edge as usual.

`logscale` When set the graph shows the y-axis in log scale. Any values less than or equal to zero are not displayed.

`axisTickSize` The spacing between axis labels and tick marks.

`axisLineColor` Color of the x- and y-axis lines. Accepts any value which the HTML canvas strokeStyle attribute understands, e.g. 'black' or 'rgb(0, 100, 255)'. This can also be set on a per-axis basis.

`axisLineWidth` Thickness (in pixels) of the x- and y-axis lines. This can also be set on a per-axis basis.

`axisLabelColor` Color for x- and y-axis labels. This is a CSS color string. This may also be set on a per-axis basis.

`axisLabelFontSize`
Size of the font (in pixels) to use in the axis labels, both x- and y-axis. This may also be set on a per-axis basis.

`axisLabelWidth` Width (in pixels) of the containing divs for x- and y-axis labels.

`drawGrid` Whether to display grid lines in the chart. This may be set on a per-axis basis to define the visibility of each axis' grid separately. Defaults to `TRUE` for x and y, and `FALSE` for y2.

`gridLineColor` The color of the grid lines. This option can also be set on a per-series basis.

`gridLineWidth` Thickness (in pixels) of the grid lines drawn under the chart. This option can also be set on a per-series basis.

`titleHeight` Height of the chart title, in pixels. This also controls the default font size of the title. If you style the title on your own, this controls how much space is set aside above the chart for the title's div.

`rightGap` Number of pixels to leave blank at the right edge of the Dygraph. This makes it easier to highlight the right-most data point.

`digitsAfterDecimal`
Unless it's run in scientific mode (see the `sigFigs` option), dygraphs displays numbers with `digitsAfterDecimal` digits after the decimal point. Trailing zeros are not displayed, so with a value of 2 you'll get '0', '0.1', '0.12', '123.45' but not '123.456' (it will be rounded to '123.46'). Numbers with absolute value less than 0.1^digitsAfterDecimal (i.e. those which would show up as '0.00') will be displayed in scientific notation.

`labelsKMB` Show K/M/B for thousands/millions/billions on y-axis.

`labelsKMG2` Show k/M/G for kilo/Mega/Giga on y-axis. This is different than `labelsKMB` in that it uses base 2, not 10.

`labelsUTC` Show date/time labels according to UTC (instead of local time). Note that this option cannot is incompatible with `useDataTimezone` (you must use one or the other).

`maxNumberWidth` When displaying numbers in normal (not scientific) mode, large numbers will be displayed with many trailing zeros (e.g. 100000000 instead of 1e9). This can lead to unwieldy y-axis labels. If there are more than maxNumberWidth digits to the left of the decimal in a number, dygraphs will switch to scientific notation, even when not operating in scientific mode. If you'd like to see all those digits, set this to something large, like 20 or 30.

`sigFigs` By default, dygraphs displays numbers with a fixed number of digits after the decimal point. If you'd prefer to have a fixed number of significant figures, set this option to that number of significant figures. A value of 2, for instance, would cause 1 to be display as 1.0 and 1234 to be displayed as 1.23e+3.

`panEdgeFraction`
A value representing the farthest a graph may be panned, in percent of the display. For example, a value of 0.1 means that the graph can only be panned 10 null means no bounds.

`animatedZooms` Set this option to animate the transition between zoom windows. Applies to programmatic and interactive zooms. Note that if you also set a drawCallback, it will be called several times on each zoom. If you set a zoomCallback, it will only be called after the animation is complete.

`mobileDisableYTouch`
Set this option to automatically disable touch events on the Y axis for mobile devices (since this interferes with swiping/scrolling on mobile devices).

`timingName` Set this option to log timing information. The value of the option will be logged along with the timing, so that you can distinguish multiple dygraphs on the same page.

`useDataTimezone`
Whether to use the time zone of the underlying xts object for display. Defaults to `FALSE` which uses the time zone of the client workstation. Note that this option is incompatible with `labelsUTC` (you must use one or other other).

`retainDateWindow`
Whether to retain the user's current date window (zoom level) when updating an existing dygraph with new data and/or options.

`disableZoom` Set this option to disable click and drag zooming.

Value

dygraph with additional options

Note

See the online documentation for additional details and examples.

dyPlotter *Include a dygraph plotter*

Description

Include a dygraph plotter

Usage

```
dyPlotter(dygraph, name, path, version = "1.0")
```

Arguments

dygraph	Dygraph to add plotter to
name	Name of plotter
path	Path to plotter JavaScript file
version	Plotter version (e.g. version of package which provides the plotter)

Value

A dygraph with the specified plotter enabled.

dyPlugin *Include a dygraph plugin*

Description

Include a dygraph plugin

Usage

```
dyPlugin(dygraph, name, path, options = list(), version = "1.0")
```

Arguments

dygraph	Dygraph to add plugin to
name	Name of plugin
path	Path to plugin JavaScript file
options	Named list of options to pass to plugin constructor
version	Plugin version (e.g. version of package which provides the plugin)

Details

You can use dygraphs plugins to customize the appearance of dygraphs as well as add new interactive behaviors. For additional information on creating dygraphs plugins see the dygraphs plugins documentation. Once you've created a dygraphs plugins you can use the dyPlugin function to create an R wrapper for it. See https://rstudio.github.io/dygraphs/gallery-plugins.html for details on how to do this.

Value

A dygraph with the specified plugin enabled.

`dyRangeSelector` *dygraph interactive range selection and zooming*

Description

Add a range selector to the bottom of the chart that allows users to pan and zoom to various date ranges.

Usage

```
dyRangeSelector(dygraph, dateWindow = NULL, height = 40,
  fillColor = " #A7B1C4", strokeColor = "#808FAB", keepMouseZoom = TRUE,
  retainDateWindow = FALSE)
```

Arguments

`dygraph`	Dygraph to add range selector to
`dateWindow`	Initially zoom in on a section of the graph. Is a two element vector [earliest, latest], where earliest/latest objects convertible via `as.POSIXct`.
`height`	Height, in pixels, of the range selector widget.
`fillColor`	The range selector mini plot fill color. This can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow". You can also specify "" to turn off fill.
`strokeColor`	The range selector mini plot stroke color. This can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow". You can also specify "" to turn off stroke.
`keepMouseZoom`	Keep mouse zoom when adding a range selector
`retainDateWindow`	Whether to retain the user's current date window (zoom level) when updating an existing dygraph with new data and/or options.

Value

A dygraph that displays a range selector

Note

See the online documentation for additional details and examples.

Shiny applications can respond to changes in the `dateWindow` via a special date window shiny input value. For example, if the output id of a dygraph is 'series' then the current date window can be read from `input$series_date_window` as an array of two date values (from and to).

Examples

```
library(dygraphs)

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector()

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector(dateWindow = c("1920-01-01", "1960-01-01"))

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector(height = 20, strokeColor = "")
```

---

dyRebase *Rebase data handler for straw broom charts with Dygraph*

Description

Draw a straw broom chart.

Usage

```
dyRebase(dygraph, value = 100, percent = FALSE)
```

Arguments

`dygraph`	Dygraph to draw chart on
`value`	Value to rebase to
`percent`	If true, ingnore value argument and rebase to percentage difference

Value

Dygraph with specified straw broom chart

Examples

```
## Not run:
library(quantmod)
tickers <- c("AAPL", "MSFT")
getSymbols(tickers)
closePrices <- do.call(merge, lapply(tickers, function(x) Cl(get(x))))
dateWindow <- c("2008-01-01", "2009-01-01")
dygraph(closePrices, main = "Value", group = "stock") %>%
  dyRebase(value = 100) %>%
  dyRangeSelector(dateWindow = dateWindow)
dygraph(closePrices, main = "Percent", group = "stock") %>%
  dyRebase(percent = TRUE) %>%
  dyRangeSelector(dateWindow = dateWindow)
dygraph(closePrices, main = "None", group = "stock") %>%
  dyRangeSelector(dateWindow = dateWindow)

## End(Not run)
```

---

dyRibbon *dyRibbon plugin adds a horizontal band of colors that runs through the chart. It can be useful to visualize categorical variables (http://en.wikipedia.org/wiki/Categorical_variable) that change over time (along the x-axis).*

Description

dyRibbon plugin adds a horizontal band of colors that runs through the chart. It can be useful to visualize categorical variables (http://en.wikipedia.org/wiki/Categorical_variable) that change over time (along the x-axis).

Usage

```
dyRibbon(dygraph, data = NULL, palette = NULL, parser = NULL, top = 1,
  bottom = 0)
```

Arguments

dygraph	Dygraph to add plugin to
data	Vector of numeric values in the range from 0 to 1
palette	Vector with colors palette
parser	JavaScrip function (function (data, dygraph)) returning the array of numeric values. Parser is used if no data was provided
top	Vertical position of the top edge of ribbon relative to chart height.
bottom	Vertical position of the bottom edge of ribbon relative to chart height.

Examples

```
## Not run:
library(quantmod)

getSymbols("SPY", from = "2016-12-01", auto.assign=TRUE)

difference <- SPY[, "SPY.Open"] - SPY[, "SPY.Close"]
decreasing <- which(difference < 0)
increasing <- which(difference > 0)

dyData <- SPY[, "SPY.Close"]

ribbonData <- rep(0, nrow(dyData))
ribbonData[decreasing] <- 0.5
ribbonData[increasing] <- 1

dygraph(dyData) %>%
    dyRibbon(data = ribbonData, top = 0.1, bottom = 0.02)

## End(Not run)
```

`dyRoller` *dygraph rolling average period text box*

Description

Add a rolling average period text box to the bottom left of the plot. Y values are averaged over the specified number of time scale units.

Usage

```
dyRoller(dygraph, showRoller = TRUE, rollPeriod = 1)
```

Arguments

`dygraph`	Dygraph to add roller to
`showRoller`	Whether to show the roller
`rollPeriod`	Number of time scale units (e.g. days, months, years) to average values over.

Value

A dygraph that displays a range selector

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(discoveries, main = "Important Discoveries") %>%
  dyRoller(rollPeriod = 5)
```

---

dySeries *dygraph data series*

Description

Add a data series to a dygraph plot. Note that options will use the default global setting (as determined by dyOptions) when not specified explicitly. When no dySeries is specified for a plot then all series within the underlying data are plotted.

Usage

```
dySeries(dygraph, name = NULL, label = NULL, color = NULL, axis = "y",
  stepPlot = NULL, stemPlot = NULL, fillGraph = NULL, drawPoints = NULL,
  pointSize = NULL, pointShape = NULL, strokeWidth = NULL,
  strokePattern = NULL, strokeBorderWidth = NULL,
  strokeBorderColor = NULL, plotter = NULL)
```

Arguments

dygraph	Dygraph to add a series definition to
name	Name of series within data set. If no name is specified then series are bound to implicitly based on their order within the underlying time series object. This parameter can also be a character vector of length 3 that specifies a set of input column names to use as the lower, value, and upper for a series with a shaded bar drawn around it.
label	Label to display for series (uses name if no label defined)
color	Color for series. These can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow", etc. If not specified then the global colors option (typically based on equally-spaced points around a color wheel).
axis	Y-axis to associate the series with ("y" or "y2")
stepPlot	When set, display the graph as a step plot instead of a line plot.
stemPlot	When set, display the graph as a stem plot instead of a line plot.
fillGraph	Should the area underneath the graph be filled? This option is not compatible with error bars.
drawPoints	Draw a small dot at each point, in addition to a line going through the point. This makes the individual data points easier to see, but can increase visual clutter in the chart.

`pointSize` The size of the dot to draw on each point in pixels. A dot is always drawn when a point is "isolated", i.e. there is a missing point on either side of it. This also controls the size of those dots.

`pointShape` The shape of the dot to draw. Can be one of the following: "dot" (default), "triangle", "square", "diamond", "pentagon", "hexagon", "circle", "star", "plus" or "ex".

`strokeWidth` The width of the lines connecting data points. This can be used to increase the contrast or some graphs.

`strokePattern` A predefined stroke pattern type ("dotted", "dashed", or "dotdash") or a custom pattern array where the even index is a draw and odd is a space in pixels. If `NULL` then it draws a solid line. The array should have an even length as any odd length array could be expressed as a smaller even length array.

`strokeBorderWidth` Draw a border around graph lines to make crossing lines more easily distinguishable. Useful for graphs with many lines.

`strokeBorderColor` Color for the line border used if `strokeBorderWidth` is set.

`plotter` A function which plots the data series. May also be set on on a global basis using `dyOptions`. See the dygraphs documentation for additional details on plotting functions.

Value

Dygraph with additional series

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

lungDeaths <- cbind(ldeaths, mdeaths, fdeaths)

dygraph(lungDeaths, main = "Deaths from Lung Disease (UK)") %>%
  dySeries("mdeaths", drawPoints = TRUE, color = "blue") %>%
  dySeries("fdeaths", stepPlot = TRUE, color = "red")
```

---

dySeriesData *Add series data to dygraph*

Description

Add an additional column of series data to a dygraph. This is typically used in the construction of custom series types (e.g. log scaled, smoothed, etc.)

Usage

```
dySeriesData(dygraph, name, values)
```

Arguments

dygraph	Dygraph to add data to
name	Name of series
values	Data values

Value

Dygraph with additional series data

---

dyShading *dygraph region shading*

Description

Specify that a region of a dygraph be drawn with a background shading

Usage

```
dyShading(dygraph, from, to, color = "#EFEFEF", axis = "x")
```

Arguments

dygraph	Dygraph to add shading to
from	Date/time or numeric to shade from (for date/time this must be a as.POSIXct object or another object convertible via as.POSIXct).
to	Date/time or numeric to shade to (for date/time this must be a as.POSIXct object or another object convertible via as.POSIXct).
color	Color of shading. This can be of the form "#AABBCC" or "rgb(255,100,200)" or "yellow". Defaults to a very light gray.
axis	Axis to apply shading. Choices are "x" or "y".

Value

A dygraph with the specified shading

Note

See the online documentation for additional details and examples.

Examples

```
library(dygraphs)

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyShading(from = "1920-1-1", to = "1930-1-1") %>%
  dyShading(from = "1940-1-1", to = "1950-1-1")

dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyShading(from = "48", to = "52", axis = "y") %>%
  dyShading(from = "50", to = "50.1", axis = "y", color = "black")
```

---

dyUnzoom

The dyUnzoom plugin adds an "Unzoom" button to the graph when it's displaying in a zoomed state (this is a bit more discoverable than the default double- click gesture for unzooming).

Description

The dyUnzoom plugin adds an "Unzoom" button to the graph when it's displaying in a zoomed state (this is a bit more discoverable than the default double- click gesture for unzooming).

Usage

```
dyUnzoom(dygraph)
```

Arguments

dygraph	Dygraph to add plugin to

Value

Dygraph with Unzoom plugin enabled

Examples

```
library(dygraphs)
dygraph(ldeaths) %>%
  dyRangeSelector() %>%
  dyUnzoom()
```

Index