

Package ‘rtsplot’

September 24, 2023

Type Package

Title Time Series Plot

Version 0.1.5

Description A fast and elegant time series visualization package. In addition to the standard R plot types, this package supports candle sticks, open-high-low-close, and volume plots. Useful for visualizing any time series data, e.g., stock prices and technical indicators.

License MIT + file LICENSE

Imports xts, quantmod, zoo, RColorBrewer

Suggests TTR

URL <https://bitbucket.org/rtsvizteam/rtsplot>

BugReports <https://bitbucket.org/rtsvizteam/rtsplot/issues>

LazyLoad yes

RoxygenNote 7.2.3

NeedsCompilation no

Author RTSVizTeam [aut, cph],
Irina Kapler [cre]

Maintainer Irina Kapler <irkapler@gmail.com>

Repository CRAN

Date/Publication 2023-09-24 00:10:02 UTC

R topics documented:

| | |
|-----------------------------------|---|
| register.theme | 2 |
| rtsplot | 3 |
| rtsplot.candle | 6 |
| rtsplot.candle.col | 7 |
| rtsplot.corner.label | 7 |
| rtsplot.fake.stock.data | 8 |
| rtsplot.format | 9 |

| | |
|--------------------------------|----|
| rtsplot.grid | 10 |
| rtsplot.hl | 10 |
| rtsplot.layout | 11 |
| rtsplot.legend | 11 |
| rtsplot.lines | 13 |
| rtsplot.matplot | 13 |
| rtsplot.ohlc | 14 |
| rtsplot.polygon | 15 |
| rtsplot.scale.volume | 16 |
| rtsplot.stacked | 16 |
| rtsplot.text | 17 |
| rtsplot.volume | 18 |
| rtsplot.x.highlight | 18 |
| rtsplot.y.highlight | 19 |
| rtsplot2Y | 20 |

| | |
|--------------|-----------|
| Index | 21 |
|--------------|-----------|

| | |
|----------------|--------------|
| register.theme | <i>Theme</i> |
|----------------|--------------|

Description

Setup theme

Usage

```
register.theme(
  grid.color = "gray90",
  colors = "Set1",
  col.border = "black",
  col.up = "green",
  col.dn = "red",
  col.x.highlight = "orange",
  col.y.highlight = "orange",
  cex = 1,
  legend.bg.col = grDevices::adjustcolor("white", 200/255)
)
```

```
rtsplot.theme()
```

```
rtsplot.theme.set(...)
```

```
rtsplot.colors(n)
```

Arguments

| | |
|-----------------|--|
| grid.color | color for grid lines, defaults to 'gray90' |
| colors | RColorBrewer set to generate colors, defaults to "Set1" in RColorBrewer |
| col.border | border color for drawing candles, defaults to 'black' |
| col.up | up color for drawing candles, defaults to 'green' |
| col.dn | down color for drawing candles, defaults to 'red' |
| col.x.highlight | color for highlighting along x axis, defaults to 'orange' |
| col.y.highlight | color for highlighting along y axis, defaults to 'orange' |
| cex | font size, defaults to 1 |
| legend.bg.col | background legend color, defaults to grDevices::adjustcolor('white', 200/255) |
| ... | additional settings |
| n | number of colors to generate |

Value

None

| | |
|---------|---|
| rtsplot | <i>'rtsplot' - Time series plot with base R Graphics.</i> |
|---------|---|

Description

Plot time series data with base R Graphics.

The 'rtsplot' package is **fast** time series plot package with base R Graphics.

Usage

```

rtsplot(
  y,
  main = NULL,
  plotX = TRUE,
  LeftMargin = 0,
  grid = "xy",
  x.highlight = NULL,
  y.highlight = NULL,
  y.highlight.col = NULL,
  las = 1,
  type = "l",
  xlab = "",
  ylab = "",
  ylim = NULL,

```

```

log = "",
skip.breaks = FALSE,
xaxis.map = rtsplot.create.xaxis.map,
...
)

```

Arguments

| | |
|------------------------------|--|
| <code>y</code> | <code>xts</code> object |
| <code>main</code> | plot title |
| <code>plotX</code> | flag to display X axis |
| <code>LeftMargin</code> | to plot second Y axis, set <code>LeftMargin=3</code> , defaults to 0 |
| <code>grid</code> | which grid lines to draw, defaults to 'xy' |
| <code>x.highlight</code> | segments to highlight along X axis, defaults to NULL |
| <code>y.highlight</code> | segments to highlight along Y axis, defaults to NULL |
| <code>y.highlight.col</code> | color to highlight segments Y axis, defaults to NULL |
| <code>las</code> | rotation of Y axis labels, defaults to 1 , for more info see par |
| <code>type</code> | plot type, defaults to 'l' , for more info see plot also support 'ohlc', 'hl', 'candle', 'volume' types |
| <code>xlab</code> | X label, defaults to "" , for more info see plot |
| <code>ylab</code> | Y label, defaults to "" , for more info see plot |
| <code>ylim</code> | range on Y values, defaults to NULL |
| <code>log</code> | log scale x, y, xy axes, defaults to "" |
| <code>skip.breaks</code> | flag to skip plotting missing date/times (i.e. nights and weekends), defaults to FALSE |
| <code>xaxis.map</code> | xaxis map function used if <code>skip.breaks</code> is TRUE, defaults to rtsplot.create.xaxis.map |
| <code>...</code> | additional parameters to the plot |

Value

nothing

Author(s)

Maintainer: Irina Kapler <irkapler@gmail.com>

Authors:

- RTSVizTeam <rtsvizteam@gmail.com> [copyright holder]

See Also

Useful links:

- <https://bitbucket.org/rtsvizteam/rtsplot>
- Report bugs at <https://bitbucket.org/rtsvizteam/rtsplot/issues>

Examples

```
# generate time series data
y = rtspplot.fake.stock.data(1000)
symbol = 'Test'

sma = TTR::SMA(y, 250)
rsi = TTR::RSI(y, 20)

# plot candles and RSI charts
layout(c(1,1,1,2))
cols = rtspplot.colors(2)

rtspplot(y, type = 'l', plotX = FALSE, col=cols[1],lwd=1.5)
rtspplot.lines(sma, col=cols[2], lwd=1.5)
rtspplot.legend(c(symbol, 'SMA(250)'), cols[1:2], list(y,sma))

# plot rsi
rtspplot(rsi, type = 'l', ylim=c(0,100),
y.highlight = c(c(0,30), c(70,100)),
y.highlight.col = grDevices::adjustcolor(c('green','red'), 50/255)
)
rtspplot.legend('RSI(20)', 'black', rsi)

y = rtspplot.fake.stock.data(1000)
symbol = 'SPY'

# simple example
highlight = which(y < 10)

# plot
layout(1)
rtspplot.theme.set(col.x.highlight=grDevices::adjustcolor('orange', 200/255))

rtspplot(y, type = 'l', main = symbol, x.highlight = highlight)

# 'skip.breaks' example with daily data
y = rtspplot.fake.stock.data(10, remove.non.trading = TRUE)

layout(1:2)
rtspplot(y, type='b')
rtspplot.legend('skip.breaks=FALSE', text.col='red')
rtspplot(y, type='b', skip.breaks=TRUE)
rtspplot.legend('skip.breaks=TRUE', text.col='red')

# 'skip.breaks' example with intra-day data
y = rtspplot.fake.stock.data(5*24*60, period = 'minute', remove.non.trading = TRUE)

layout(1:2)
rtspplot(y, type='l')
```

```
rtsplot.legend('skip.breaks=FALSE', text.col='red')
rtsplot(y, type='l', skip.breaks=TRUE)
rtsplot.legend('skip.breaks=TRUE', text.col='red')
```

rtsplot.candle *Create Candle Plot*

Description

Plot candles if dx is sufficient otherwise ohlc or bars

Usage

```
rtsplot.candle(
  y,
  col = rtsplot.candle.col(y),
  border = rtsplot.theme()$col.border
)
```

Arguments

| | |
|--------|--|
| y | xts object |
| col | color for bars, defaults to rtsplot.candle.col |
| border | border color, defaults to rtsplot.theme()\$col.border |

Value

nothing

Examples

```
y = rtsplot.fake.stock.data(50, ohlc=TRUE)
symbol = 'SPY'

# plot
layout(1)
rtsplot(y, type = 'n')
rtsplot.candle(y)
rtsplot.legend(symbol, 'black', y)
```

`rtspplot.candle.col` *Bar Colors for Candle and Volume plots*

Description

Bar Colors for Candle and Volume plots

Usage

```
rtspplot.candle.col(y)
```

```
rtspplot.volume.col(y)
```

Arguments

`y` `xts` object

Value

colors

`rtspplot.corner.label` *Plot corner label*

Description

Plot corner label, based on the [text at the upper left corner outside of the plot region](<http://r.789695.n4.nabble.com/text-at-the-upper-left-corner-outside-of-the-plot-region-td885675.html>)

Usage

```
rtspplot.corner.label(  
  label = NULL,  
  col = "black",  
  x = -1,  
  y = 1,  
  xoffset = NA,  
  yoffset = NA,  
  space = c("plot", "figure"),  
  cex = 1,  
  border = NA  
)
```

Arguments

| | |
|---------|--|
| label | label |
| col | label color |
| x | x location, defaults to -1 |
| y | y location, defaults to 1 |
| xoffset | x offset, defaults to NA |
| yoffset | y offset, defaults to NA |
| space | coordinate space, can be "plot" or "figure", defaults to "plot" |
| cex | font size, defaults to 1 |
| border | border color, defaults to NA - no color |

Value

nothing

Examples

```
rtsplot.theme.set(legend.bg.col=grDevices::adjustcolor('orange', 200/255))
plot(rnorm(20), rnorm(20))
```

```
rtsplot.corner.label('test1', y=-1, space='figure')
rtsplot.corner.label('test2', y=1, space='figure')
rtsplot.corner.label('test3', x=1, space='figure')
rtsplot.corner.label('test4', x=1, y=-1, space='figure')
rtsplot.theme.set(legend.bg.col=grDevices::adjustcolor('white', 50/255))
```

```
rtsplot.fake.stock.data
      Generate fake stock data
```

Description

Generate fake stock data for use in rtsplot examples

Usage

```
rtsplot.fake.stock.data(
  n,
  y0 = 10,
  stdev = 0.1,
  ohlc = FALSE,
  method = c("normal", "adhoc"),
  period = c("day", "minute"),
  remove.non.trading = FALSE
)
```


Arguments

| | |
|--------------------|--|
| n | number of points to generate |
| y0 | starting price, defaults to 10 |
| stdev | standard deviation, defaults to 0.1 |
| ohlc | generate ohlc data, defaults to FALSE |
| method | method to generate fake stock data, defaults to 'normal' two methods are implemented: * 'normal' - generate fake stock data assuming returns are normally distributed with zero drift * 'uniform' - generate fake stock data assuming returns are uniformly distributed with zero drift |
| period | frequency to generate fake stock data, (possible values: "day", "minute"), defaults to "day" |
| remove.non.trading | flag to remove non trading periods(i.e. weekends and non-trading hours). Note, this flag likely will cause function return less than 'n' observation, defaults to FALSE |

Value

`xts` object with fake stock data

Examples

```
rtsplot.fake.stock.data(10)
```

| | |
|----------------|--|
| rtsplot.format | <i>Format numbers using 1000 separator</i> |
|----------------|--|

Description

Format numbers using 1000 separator

Usage

```
rtsplot.format(temp, nround = 2, sprefix = "", eprefix = "")
```

Arguments

| | |
|---------|---|
| temp | numbers |
| nround | number of rounding digits, defaults to '2' |
| sprefix | start prefix string, defaults to "" |
| eprefix | end postfix string, defaults to "" |

Value

numbers formatted using 1000 separator

| | |
|--------------|-------------------------------------|
| rtsplot.grid | <i>Add grid to time series plot</i> |
|--------------|-------------------------------------|

Description

Add grid to time series plot

Usage

```
rtsplot.grid(grid, xaxis.ticks, col = rtsplot.theme()$grid.color)
```

Arguments

| | |
|-------------|--|
| grid | which grid lines to draw, defaults to 'xy' |
| xaxis.ticks | location of x axis ticks |
| col | grid color, defaults to rtsplot.theme()\$grid.color |

Value

nothing

| | |
|------------|-----------------------|
| rtsplot.hl | <i>Create HL Plot</i> |
|------------|-----------------------|

Description

Create HL Plot

Usage

```
rtsplot.hl(y, col = rtsplot.volume.col(y), border = rtsplot.theme()$col.border)
```

Arguments

| | |
|--------|--|
| y | xts object |
| col | color for bars, defaults to rtsplot.volume.col |
| border | border color, defaults to rtsplot.theme()\$col.border |

Value

nothing

Examples

```
y = rtspplot.fake.stock.data(50, ohlc=TRUE)
symbol = 'SPY'

# plot
layout(1)
rtspplot(y, type = 'n')
rtspplot.hl(y)
rtspplot.legend(symbol, 'black', y)
```

| | |
|---|----------------------|
| <code>rtsp<code>plot.layout</code></code> | <i>Create layout</i> |
|---|----------------------|

Description

Create layout

Usage

```
rtspplot.layout(ilayout, delim = ",")
```

Arguments

| | |
|----------------------|-----------------------------------|
| <code>ilayout</code> | matrix stored as a string |
| <code>delim</code> | delimiter, defaults to ',' |

Value

nothing

| | |
|---|--|
| <code>rtsp<code>plot.legend</code></code> | <i>Plot legend - shortcut to the legend function</i> |
|---|--|

Description

Plot legend - shortcut to the [legend](#) function

Usage

```
rtsplot.legend(  
  labels,  
  fill = NULL,  
  lastobs = NULL,  
  x = "topleft",  
  merge = FALSE,  
  bty = "n",  
  border = NA,  
  yformat = rtsplot.format,  
  cex = 1,  
  ...  
)
```

Arguments

| | |
|---------|---|
| labels | legend labels |
| fill | fill colors, defaults to NULL |
| lastobs | list of last observations, defaults to NULL |
| x | location of legend, defaults to 'topleft' |
| merge | merge, defaults to FALSE , see legend function for more info |
| bty | box, defaults to 'n' , see legend function for more info |
| border | border color, defaults to NA - no color |
| yformat | format Y values function, defaults to rtsplot.format |
| cex | font size, defaults to 1 |
| ... | other parameters to legend, see legend function for more info |

Value

nothing

Examples

```
y = rtsplot.fake.stock.data(1000)  
symbol = 'SPY'  
  
# plot  
layout(1)  
rtsplot(y, type = 'l', col='black')  
rtsplot.legend(symbol, 'black', y)
```

| | |
|----------------|--------------------------------------|
| rtspplot.lines | <i>Add lines to time series plot</i> |
|----------------|--------------------------------------|

Description

Add lines to time series plot

Usage

```
rtspplot.lines(y, type = "l", col = graphics::par("col"), ...)
```

Arguments

| | |
|------|---|
| y | xts object |
| type | line type, defaults to 'l' , for more info see lines |
| col | color, defaults to par('col') |
| ... | additional parameters to the lines |

Value

nothing

Examples

```
y = rtspplot.fake.stock.data(1000)
symbol = 'SPY'

# moving average
sma = TTR::SMA(y, 250)

# plot
layout(1)
rtspplot(y, type = 'l', col='black')
rtspplot.lines(sma, col='blue', lwd=1.5)
rtspplot.legend(c(symbol, 'SMA(250)'), 'black,blue', list(y,sma))
```

| | |
|------------------|--|
| rtspplot.matplot | matplot version for xts object |
|------------------|--|

Description

[matplot](#) version for [xts](#) object

Usage

```

rtsplot.matplot(
  y,
  dates = NULL,
  ylim = NULL,
  type = "l",
  cols = rtsplot.colors(ncol(y)),
  ...
)

```

Arguments

| | |
|-------|--|
| y | xts object |
| dates | subset of dates defaults to NULL |
| ylim | range on Y values, defaults to NULL |
| type | plot type, defaults to 'l' , see plot for details |
| cols | colors |
| ... | additional parameters to the matplot |

Value

nothing

 rtsplot.ohlc

Create OHLC Plot

Description

Plot ohlc if dx is sufficient otherwise bars

Usage

```

rtsplot.ohlc(y, col = rtsplot.theme()$col.border)

```

Arguments

| | |
|-----|--|
| y | xts object |
| col | color for bars, defaults to rtsplot.theme()\$col.border |

Value

nothing

Examples

```

y = rtspoly.fake.stock.data(50, ohlc=TRUE)
symbol = 'SPY'

# plot
layout(1)
rtspoly(y, type = 'n')
rtspoly.ohlc(y)
rtspoly.legend(symbol, 'black', y)

rtspoly.theme.set(legend.bg.col=grDevices::adjustcolor('blue', 25/255))
rtspoly.corner.label('Logo \uA9', x=1, y=-1, cex = 0.7, space='figure', col='blue')
rtspoly.theme.set(legend.bg.col = grDevices::adjustcolor('white', 200/255))

```

rtspoly.polygon *Add polygon to time series plot*

Description

Add polygon to time series plot

Usage

```
rtspoly.polygon(y, col = graphics::par("col"), ...)
```

Arguments

| | |
|-----|--|
| y | xts object with 2 columns |
| col | color, defaults to par('col') |
| ... | additional parameters to the lines |

Value

nothing

Examples

```

y = rtspoly.fake.stock.data(1000, ohlc=TRUE)
symbol = 'SPY'

# moving average
bbands = TTR::BBands(quantmod::HLC(y), n=200, sd=1)[,c('up', 'dn')]

# plot
layout(1)
rtspoly(y, type = 'l', col='black')
col = grDevices::adjustcolor('green', 50/255)

```

```
rtsplot.polygon(bbands, col = col)
rtsplot.legend(c(symbol, 'BBands'), c('black', col), list(y,bbands))
```

`rtsplot.scale.volume` *Scale volume*

Description

Scale volume

Usage

```
rtsplot.scale.volume(y)
```

Arguments

`y` [xts](#) object

Value

adjusted y object

`rtsplot.stacked` *Create Stacked plot*

Description

Create Stacked plot

Usage

```
rtsplot.stacked(
  x,
  y,
  xlab = "",
  cols = rtsplot.colors(ncol(y)),
  type = c("l", "s"),
  flip.legend = FALSE,
  ...
)
```


Arguments

| | |
|-------------|--|
| x | dates object |
| y | matrix with weights |
| xlab | X label, defaults to "", for more info see plot |
| cols | colors, defaults to colors rtspplot.theme |
| type | plot type: lines, step stairs c('l','s') |
| flip.legend | flag to reverse legend order, defaults to FALSE |
| ... | additional parameters to the plot |

Value

nothing

| | |
|---------------|-------------------------------------|
| rtspplot.text | <i>Add text to time series plot</i> |
|---------------|-------------------------------------|

Description

Add text to time series plot

Usage

```
rtspplot.text(y, ...)
```

Arguments

| | |
|-----|--|
| y | xts object |
| ... | additional parameters to the lines |

Value

nothing

Examples

```
y = rtspplot.fake.stock.data(1000)
symbol = 'SPY'

# plot
layout(1)
rtspplot(y, type = 'l', col='black')
rtspplot.text(y[100], 'Text', col='red')
rtspplot.legend(symbol, 'black', y)
```

`rtsplot.volume` *Plot volume*

Description

Plot volume

Usage

```

rtsplot.volume(
  y,
  col = rtsplot.volume.col(y),
  border = rtsplot.theme()$col.border
)

```

Arguments

| | |
|---------------------|------------------------------|
| <code>y</code> | <code>xts</code> object |
| <code>col</code> | color for volume bars |
| <code>border</code> | color for volume bars border |

Value

nothing

`rtsplot.x.highlight` *Highlight vertical segments*

Description

Highlight vertical segments

Usage

```

rtsplot.x.highlight(y, highlight, col = rtsplot.theme()$col.x.highlight)

```

Arguments

| | |
|------------------------|---|
| <code>y</code> | <code>xts</code> object |
| <code>highlight</code> | segments to highlight along X axis |
| <code>col</code> | highlight color, defaults to <code>rtsplot.control\$col.x.highlight</code> |

Value

nothing

rtsplot.y.highlight *Highlight horizontal segments*

Description

Highlight horizontal segments

Usage

```
rtsplot.y.highlight(highlight, col = rtsplot.theme()$col.y.highlight)
```

Arguments

| | |
|-----------|--|
| highlight | segments to highlight along Y axis |
| col | highlight color, defaults to rtsplot.control\$col.y.highlight |

Value

nothing

Examples

```
# generate time series data
y = rtsplot.fake.stock.data(1000)

rsi = TTR::RSI(y, 20)

#set up two regions for graphs candlestick price data on top 2/3 of the plot
#and rsi on the bottom 1/3 of the plot
layout(c(1,1,2))

rtsplot(y, type = 'line', plotX = FALSE)
  rtsplot.legend('SPY', 'grey70', y)

rtsplot(rsi, type = 'l')

col = grDevices::adjustcolor(c('green','red'), 80/255)
rtsplot.y.highlight(col=col[1], highlight=c(50,100))
rtsplot.y.highlight(col=col[2], highlight=c(0,50))

abline(h = 50, col = 'gray20')

rtsplot.legend('RSI(20)', 'black', rsi)
```

rtsplot2Y

*Plot time series with second Y axis***Description**

Detailed discussion for validity of dual Y axis at [Dual axes time series plots may be ok sometimes after all](<http://freerangestats.info/blog/2016/08/18/dualaxes>)

Usage

```
rtsplot2Y(y, las = 1, type = "l", col.axis = "red", ylim = NULL, log = "", ...)
```

Arguments

| | |
|----------|--|
| y | xts object |
| las | rotation of Y axis labels, defaults to 1 , for more info see par |
| type | plot type, defaults to 'l' , for more info see plot also support 'ohlc', 'hl', 'candle', 'volume' types |
| col.axis | axis color, defaults to 'red' |
| ylim | range on Y values, defaults to NULL |
| log | log scale x, y, xy axes, defaults to '' |
| ... | additional parameters to the plot |

Value

nothing

Examples

```
# generate time series data
y = rtsplot.fake.stock.data(1000)
symbol = 'SPY'

y1 = rtsplot.fake.stock.data(1000, 100)
symbol = 'IBM'

# two Y axis example
# to plot second Y axis, free some space on left side, set LeftMargin=3
layout(1)
cols = c('black', 'red')

rtsplot(y, type = 'l', LeftMargin=3, col=cols[1])

rtsplot2Y(y1, type='l', las=1, col=cols[2], col.axis=cols[2])

rtsplot.legend('SPY(rhs),IBM(lhs)', cols, list(y,y1))
```

Index

legend, [11](#), [12](#)
lines, [13](#), [15](#), [17](#)

matplot, [13](#), [14](#)

par, [4](#), [20](#)
plot, [4](#), [14](#), [17](#), [20](#)

register.theme, [2](#)
rtsplot, [3](#)
rtsplot-package (rtsplot), [3](#)
rtsplot.candle, [6](#)
rtsplot.candle.col, [7](#)
rtsplot.colors (register.theme), [2](#)
rtsplot.corner.label, [7](#)
rtsplot.fake.stock.data, [8](#)
rtsplot.format, [9](#), [12](#)
rtsplot.grid, [10](#)
rtsplot.hl, [10](#)
rtsplot.layout, [11](#)
rtsplot.legend, [11](#)
rtsplot.lines, [13](#)
rtsplot.matplot, [13](#)
rtsplot.ohlc, [14](#)
rtsplot.polygon, [15](#)
rtsplot.scale.volume, [16](#)
rtsplot.stacked, [16](#)
rtsplot.text, [17](#)
rtsplot.theme, [17](#)
rtsplot.theme (register.theme), [2](#)
rtsplot.volume, [18](#)
rtsplot.volume.col
 (rtsplot.candle.col), [7](#)
rtsplot.x.highlight, [18](#)
rtsplot.y.highlight, [19](#)
rtsplot2Y, [20](#)

xts, [4](#), [6](#), [7](#), [9](#), [10](#), [13–18](#), [20](#)