Package 'rpnf'

October 14, 2022

Type Package

Title Point and Figure Package

Version 1.0.5

Date 2016-08-26

Author Sascha Herrmann

Maintainer Sascha Herrmann <sascha.herrmann.consulting@gmail.com>

Description A set of functions to analyze and print the development of a commodity using the Point and Figure (P&F) approach. A P&F processor can be used to calculate daily statistics for the time series. These statistics can be used for deeper investigations as well as to create plots. Plots can be generated as well known X/O Plots in plain text format, and additionally in a more graphical format.

License GPL-3

Depends R (>= 3.0.0)

Suggests testthat

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-08-26 14:30:43

R topics documented:

`rpnf-package` *rpnf - The R Point & Figure Package*

Description

rpnf is a tool set to create and analyze Point & Figure Charts for given time series or data frame objects.

Author(s)

Sascha Herrmann <sascha.herrmann.consulting@gmail.com>

References

Project Home Page http://rpnf.r-forge.r-project.org

Dorsey, Thomas J. Point and Figure Charting: The Essential Application for Forecasting and Tracking Market Prices. 3rd ed. Wiley Trading. Hoboken, N.J: John Wiley & Sons, 2007.

German version, which is the base for the package: Dorsey, Thomas. Sicher anlegen mit point & figure: klare Signale mit einfachen Methoden. Munich: FinanzBuch-Verl., 2000.

See Also

`pnfprocessor`

`pnfplot`

`pnfplottxt`

Examples

```
# Load rpnf library
library(rpnf)
# Load free available sample data
data(DOW)
# Determine point and figure informations for a linear chart with boxsize of 1 point
pnfdata <- pnfprocessor(
  high=DOW$High,
  low=DOW$Low,
  date=DOW$Date,
  boxsize=1L,
  log=FALSE)
# Show the object obtained
str(pnfdata)
# Show the data obtained
pnfdata
# Create a TXT based plot with X and O's
pnfplottxt(pnfdata,boxsize=1L,log=FALSE)
# Create a more graphical plot
pnfplot(pnfdata)
## Not run:
### Second example: logarithmc example
# For most stocks and indices it is useful
# to do the analysis on a logarithmic scale.
# This can be done with pnfprocessor, too.
# Ensure to make use of the getLogBoxsize() function
# for an appropriate boxsize of a logarithmic chart.
# Determine point and figure informations for a logarithmic chart with boxsize 2\%
symbol.pnf <- pnfprocessor(
  high=DOW$High,
  low=DOW$Low,
  date=DOW$Date,
  boxsize=getLogBoxsize(2),
  log=TRUE)

# View the result
tail(symbol.pnf)
#View(symbol.pnf)

# or plot it as a modern chart
pnfplot(symbol.pnf,main="P&F Plot DOW (log)")
# Or in the old traditional TXT style
pnfplottxt(symbol.pnf,boxsize=getLogBoxsize(2),log=TRUE,main="P&F Plot DOW (log)")

### Additional examples
# Examples for additional uses cases like
# - relative strength vs index
# - bullish percent of an index
# - and many others
# can be found in your local package library directory.
# Search for rpnf-example1.R, rpnf-example2.R and so on.
```

```
## End(Not run)
```

---

box2lower *Returns the lower bound value for a given boxnumber*

Description

Returns the lower bound value for a given boxnumber

Usage

```
box2lower(boxnumber, boxsize = 1, log = FALSE)
```

Arguments

boxnumber	An integer boxnumber
boxsize	single numeric value, used as the boxsize
log	TRUE, if logarithmic scales should be used

---

box2upper *Returns the upper bound value for a given boxnumber*

Description

Returns the upper bound value for a given boxnumber

Usage

```
box2upper(boxnumber, boxsize = 1, log = FALSE)
```

Arguments

boxnumber	An integer boxnumber
boxsize	single numeric value, used as the boxsize
log	TRUE, if logarithmic scales should be used

`bp.signalprocessor` *This function identifies chart signals in an [0,100]-Points Bullish Percent Chart*

Description

This function identifies chart signals in an [0,100]-Points Bullish Percent Chart

Usage

```
bp.signalprocessor(data)
```

Arguments

`data`	Input data

`currentVPOBreakoutMethod` *Identifiy for a given P&F Table the current vertical price objective triggered by the last signal reversal.*

Description

Identifiy for a given P&F Table the current vertical price objective triggered by the last signal reversal.

Usage

```
currentVPOBreakoutMethod(data, reversal, boxsize, log)
```

Arguments

`data`	Input data
`reversal`	Number of boxes for reversal
`boxsize`	Size of one box
`log`	Use logarithmic scale

`currentVPOReversalMethod` *Identifiy for a given P&F Table the current vertical price objective triggered by the last signal reversal.*

Description

Identifiy for a given P&F Table the current vertical price objective triggered by the last signal reversal.

Usage

```
currentVPOReversalMethod(data, reversal, boxsize, log)
```

Arguments

`data`	Input data
`reversal`	Number of boxes for reversal
`boxsize`	Size of one box
`log`	Use logarithmic scale

`doubleBottom` *returns true if given column c matches exactly previous column of same type (this is always column c-2)*

Description

returns true if given column c matches exactly previous column of same type (this is always column c-2)

Usage

```
doubleBottom(redData, column)
```

Arguments

`redData`	Data to consider
`column`	Column to consider

`doubleTop` *Returns true if given column c matches exactly previous column of same type (this is always column c-2)*

Description

Returns true if given column c matches exactly previous column of same type (this is always column c-2)

Usage

```
doubleTop(redData, column)
```

Arguments

`redData`	Data to consider
`column`	Column to consider

`DOW` *This is some free available quote data for the DOW Chemical Company.*

Description

End of day open, high, low, close and volume, dividends and splits, and split/dividend adjusted open, high, low close and volume for Dow Chemical Company (The) (DOW). Data are freely available at https://www.quandl.com/data/WIKI/DOW, and may be copy, distribute, disseminate or include the data in other products for commercial and/or noncommercial purposes. This data is part of Quandl's Wiki initiative to get financial data permanently into the public domain. Quandl relies on users like you to flag errors and provide data where data is wrong or missing. Get involved: connect@quandl.com

Author(s)

Sascha Herrmann <sascha.herrmann.consulting@gmail.com>

References

https://www.quandl.com/data/WIKI/DOW

`fallingBottom` *Returns true if given column c drops below prevois column of same type (this is always column c-2)*

Description

Returns true if given column c drops below prevois column of same type (this is always column c-2)

Usage

```
fallingBottom(redData, column)
```

Arguments

`redData`	Data to consider
`column`	Column to consider

`fallingTop` *returns true if given column c drops below previous column of same type (this is always column c-2)*

Description

returns true if given column c drops below previous column of same type (this is always column c-2)

Usage

```
fallingTop(redData, column)
```

Arguments

`redData`	Data to consider
`column`	Column to consider

getLogBoxsize *Determine an appropriate boxsize, if you want to use logarithmic scale.*

Description

This function returns an appropriate boxsize if you want to do your point and figure analysis with an logarithmic scale.

Usage

```
getLogBoxsize(percent)
```

Arguments

percent	a numeric value defining the percent

Value

a numeric value which is equivalent to the percental change given on a logarithmic scale

Examples

```
# apply it with pnfprocessor
library(rpnf) # Load rpnf library
data(DOW) # Load some example data

# return appropriate value for 1% boxsize
getLogBoxsize(percent=1)

pnfprocessor(
 high=DOW$High,
 low=DOW$Low,
 date=DOW$Date,
 boxsize=getLogBoxsize(percent=1),
 log=TRUE)
```

maxBox *Returns the maximum box number in given column*

Description

Returns the maximum box number in given column

Usage

```
maxBox(redData, column)
```

Arguments

redData	Data to consider
column	Column to consider

---

minBox *Returns the minimum box number in given column*

Description

Returns the minimum box number in given column

Usage

```
minBox(redData, column)
```

Arguments

redData	Data to consider
column	Column to consider

---

nextBox *Determine the next box frontier for current quote(s) given a recent XO-status.*

Description

Note: offset should only be used for reversal calculation

Usage

```
nextBox(quote, status, boxsize = 1, log = FALSE)
```

Arguments

quote	A single quote or a vector of quotes.
status	A single character indicating the current XO-status.
boxsize	A single numeric value, indicating the boxsize to be considered.
log	TRUE, if logarithmic scales should be used.

nextReversal — *Determine the next reversal frontier for current quote(s) given a recent XO-status.*

Description

Determine the next reversal frontier for current quote(s) given a recent XO-status.

Usage

```
nextReversal(quote, status, reversal = 3L, boxsize = 1, log = FALSE)
```

Arguments

quote	A single quote or a vector of quotes.
status	A single character indicating the current XO-status.
reversal	number of boxes needed to make a reversal
boxsize	A single numeric value, indicating the boxsize to be considered.
log	TRUE, if logarithmic scales should be used.

pnfplot — *Generate a modern point and figure plot*

Description

Generate a modern point and figure plot

Usage

```
pnfplot(data, reversal = 3, boxsize = 1, log = FALSE, ...)
```

Arguments

data	a data frame object containing point and figure informations to be plotted
reversal	number of boxes used in pnfprocessor
boxsize	the boxsize used in pnfprocessor
log	are calculations done in logarithmic mode
...	any additional options for the plot command

References

http://rpnf.r-forge.r-project.org

See Also

pnfprocessor

pnfplottxt

Examples

```
library(rpnf) # Load rpnf library
data(DOW) # (Offline) Load free available sample data from https://www.quandl.com/data/WIKI/DOW
pnfdata <- pnfprocessor(
  high=DOW$High,
  low=DOW$Low,
  date=DOW$Date,
  boxsize=1L,
  log=FALSE)
pnfplot(pnfdata,boxsize=1L,log=FALSE)
```

---

pnfplottxt *Generate a classical TXT point and figure plot.*

Description

THIS FUNCTION IS STILL UNDER DEVELOPMENT, THEREFORE IT MIGHT BE SUBJECT TO CHANGE!

Usage

```
pnfplottxt(data, reversal = 3, boxsize = 1, log = FALSE, main = NULL,
  sub = NULL)
```

Arguments

data	a data frame object containing point and figure informations to be plotted
reversal	number of boxes used in pnfprocessor
boxsize	the boxsize used in pnfprocessor
log	are calculations done in logarithmic mode
main	a string used as a main title of the chart
sub	a string used as a sub title of the chart

References

http://rpnf.r-forge.r-project.org

See Also

pnfprocessor

pnfplot

Examples

```
library(rpnf) # Load rpnf library
data(DOW) # (Offline) Load free available sample data from https://www.quandl.com/data/WIKI/DOW
pnfdata <- pnfprocessor(
  high=DOW$High,
  low=DOW$Low,
  date=DOW$Date,
  boxsize=1L,
  log=FALSE)
pnfplottxt(pnfdata,boxsize=1L,log=FALSE)
```

`pnfprocessor` *Generate all point and figure informations for a given time series.*

Description

Please ensure that high, low and date are all ordered according to the Date column.

Usage

```
pnfprocessor(high, low = high, date, reversal = 3L, boxsize = 1L,
  log = FALSE, style = "xo")
```

Arguments

`high`	a vector containing the high quotes
`low`	a (optional) vector containing the low quotes
`date`	a vector of dates the quotes belong
`reversal`	number of boxes needed to make a reversal
`boxsize`	the boxsize to be used
`log`	should we do the calculations on a logarithmic scale
`style`	the style the pnfprocessor is working with. Can be {xo,rs,bp}.

Value

returns a data table with all point and figure information in it

References

http://rpnf.r-forge.r-project.org

See Also

pnfplot

pnfplottxt

Examples

```
library(rpnf) # Load rpnf library
data(DOW) # (Offline) Load free available sample data from https://www.quandl.com/data/WIKI/DOW
pnfdata <- pnfprocessor(
  high=DOW$High,
  low=DOW$Low,
  date=DOW$Date,
  boxsize=1L,
  log=FALSE)
pnfdata
```

`quote2box` *Converts a single or a vector of quotes into integer boxnumbers for P&F-Analysis.*

Description

Converts a single or a vector of quotes into integer boxnumbers for P&F-Analysis.

Usage

```
quote2box(quote, boxsize = 1, log = FALSE)
```

Arguments

`quote`	a single quote, or a vector of quotes
`boxsize`	single numeric value, used as the boxsize
`log`	TRUE, if logarithmic scales should be used

Value

a single or a vector of integer boxnumbers This function transforms a given quote into an unique integer box number

---

`quoteToBoxnumber` *Determines the boxnumber for a given tuple of quotes,status,boxsize and log.*

Description

Determines the boxnumber for a given tuple of quotes,status,boxsize and log.

Usage

```
quoteToBoxnumber(quote, status, boxsize, log)
```

Arguments

quote	a numeric vector of quotes
status	current status, either "X" or "O"
boxsize	boxsize
log	use log scale, either TRUE or FALSE

Value

a vector of integer boxnumbers

---

quoteToScale — *Scales a quote. In case log==TRUE this is logarithmic scale, original scale otherwise.*

Description

Scales a quote. In case log==TRUE this is logarithmic scale, original scale otherwise.

Usage

```
quoteToScale(x, log)
```

Arguments

x	a numeric vector of quotes
log	TRUE or FALSE

Value

scaled quote

---

raisingBottom — *returns true if given column c exceeds prevois column of same type (this is always column c-2)*

Description

returns true if given column c exceeds prevois column of same type (this is always column c-2)

Usage

```
raisingBottom(redData, column)
```

Arguments

redData	Data to consider
column	Column to consider

raisingTop — *Returns true if given column c exceeds previous column of same type (this is always column c-2)*

Description

Returns true if given column c exceeds previous column of same type (this is always column c-2)

Usage

```
raisingTop(redData, column)
```

Arguments

redData	Data to consider
column	Column to consider

rs.signal.processor — *This function analyzes a (preliminary) P&F Chart for Bullish Support Line and Bearish Resistance Line*

Description

Finding the appropriate trendlines is explained very good at http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:pnf_charts:pnf_trendlines.

Usage

```
rs.signal.processor(data)
```

Arguments

data	Input data

See Also

http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:pnf_charts:pnf_trendlines

scaleToQuote *Rescales a scaled quote to original scale.*

Description

Rescales a scaled quote to original scale.

Usage

```
scaleToQuote(x, log)
```

Arguments

x	a numeric vector of scaled quotes
log	TRUE or FALSE

Value

scaled quote

signalanalyzer *analyze transitions of signal states*

Description

analyze transitions of signal states

Usage

```
signalanalyzer(signal, probability = TRUE)
```

Arguments

signal	Signal to identify
probability	Report probability

`xo.priceobjective.processor` *This function adds Vertical Price Objectives calculated with the Bullish Breakout and Bearish Breakdown Method (BM) to an P&F Table.*

Description

Finding the appropriate price objectives has been explained very good at http://stockcharts.com/school/doku.php?id=chart_sc but this documentation is no longer available. The function adds columns vpo_bm_boxnumber and vpo_bm_price to the given P&F Table. vpo_bm_bonumber contains the boxnumber of the price objective, while vpo_bm_price contains the real price objective.

Usage

```
xo.priceobjective.processor(data, reversal, boxsize, log)
```

Arguments

`data`	Input data
`reversal`	Number of boxes for reversal
`boxsize`	Size of one box
`log`	Use logarithmic scale

`xo.signalprocessor` *Analyzes a given PNF time-series for Buy&Sell patterns*

Description

Analyzes a given PNF time-series for Buy&Sell patterns

Usage

```
xo.signalprocessor(data, reversal = 3)
```

Arguments

`data`	Input data
`reversal`	Number of boxes for reversal

Index