# Package 'RAdamant'

July 2, 2011

**Type** Package

**Title** Financial Technical Analysis and Risk Management

**Version** 0.7.1

**Date** 2011-07-02

**Author** RAdamant Development Team

**Maintainer** RAdamant Development Team <team@r-adamant.org>

**Depends** R (>= 2.10.0), utils, grDevices

**Description** R-Adamant is a collection of functions and algorithms for processing of Financial Time Series, Risk Management and Econometrics.

**License** GPL>=2

**LazyLoad** yes

## R topics documented:

---

`3dptelem` *3D Plot Elements*

---

## Description

Add elements to 3D Plot

## Usage

```
lines3d(x, y, z, pmat = getProjectionMatrix(), ...)
points3d(x, y, z, pmat = getProjectionMatrix(), ...)
rect3d(xrange, yrange, z, pmat = getProjectionMatrix(), ...)
text3d(x, y, z, pmat = getProjectionMatrix(), ...)
```

## Arguments

| | |
|---|---|
| `x` | X axis |
| `y` | Y axis |
| `z` | Z axis |
| `pmat` | pamt |
| `...` | Further arguments to or from other methods |
| `xrange` | xrange |
| `yrange` | yrange |

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`3dptpars` *3D Plot Axis Formatting*

## Description

Add and format labels for 3D Plot

## Usage

```
x.axis3d(xlim = NULL, ylim = NULL, zlim = NULL,
pmat = getProjectionMatrix(), at = NULL,
labels = NULL, theme.params = getCurrentTheme(),
show.labels = TRUE, grid = theme.params[["xgrid"]],...)

y.axis3d(xlim = NULL, ylim = NULL, zlim = NULL,
pmat = getProjectionMatrix(), at = NULL,
labels = NULL, theme.params = getCurrentTheme(),
show.labels = TRUE, grid = theme.params[["ygrid"]],...)

z.axis3d(xlim = NULL, ylim = NULL, zlim = NULL,
pmat = getProjectionMatrix(), at = NULL, labels =NULL,
theme.params = getCurrentTheme(), show.labels = TRUE,
grid = theme.params[["zgrid"]], ...)

x.title3d(xlim = NULL, ylim = NULL, zlim = NULL,
pmat = getProjectionMatrix(), title = "",
theme.params = getCurrentTheme(), ...)

y.title3d(xlim = NULL, ylim = NULL, zlim = NULL,
pmat = getProjectionMatrix(), title = "",
theme.params = getCurrentTheme(), ...)

z.title3d(xlim = NULL, ylim = NULL, zlim = NULL,
pmat = getProjectionMatrix(), title = "",
theme.params = getCurrentTheme(), ...)
```

## Arguments

| | |
|---|---|
| `xlim` | xlim |
| `ylim` | ylim |
| `zlim` | zlim |
| `pmat` | pmat |
| `at` | at |
| `labels` | labels |
| `title` | title |
| `theme.params` | theme.params |
| `show.labels` | show.labels |

| | |
|---|---|
| grid | grid |
| ... | Further arguments to or from other methods |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

abi *Absolute Breath Index - ABI*

---

### Description

Compute Absolute Breath Index (Technical Analysis)

### Usage

```
Abi(X, lag = 5, plot=FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Input numerical series |
| lag | Number of lags |
| plot | LOGICAL. Return plot. |
| ... | Further arguments to or from other methods |

### Details

The absolute breadth index (ABI) measures the amount of volatile on the New York Stock Exchange. The indicator is truly a momentum indicator since it only tracks the movement of the issues on the exchange and not their direction. The overall purpose of the indicator is to determine when the market is most volatile, thus providing the greatest opportunity for larger profits due to increased volatility.
Calculation:

$$ABI = |Advancing\ Issues - Declining\ issues|$$

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## absavdi *Absolute average discard*

### Description

Calculate Absolute average discard for model evaluation

### Usage

```
abs_avdi(actual, predicted, pc = FALSE)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |
| pc | pc |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## absrs *Absolute Relative Strenght*

### Description

Compute Absolute Relative Strenght (Technical Analysis)

### Usage

```
absrs(X, lag = 14, na.rm = FALSE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| na.rm | na.rm |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## accuracy *Accuracy*

### Description

Get accuracy measure from the results of a classification model.

### Usage

```
accuracy(x, ...)
## S3 method for class 'scorecard'
accuracy(x, th, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| th | th |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## acdi *Acceleration Deceleration*

### Description

Acceleration Deceleration Technical Indicator

### Usage

```
acdi(Close, High = NULL, Low = NULL, Vol = NULL, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Vol | VECTOR. Asset traded Volume. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

Acceleration/-Deceleration Technical Indicator (AC) measures acceleration and deceleration of the current driving force.
This indicator will change direction before any changes in the driving force, which, it its turn, will change its direction before the price.
The nought line is basically the spot where the driving force is at balance with the acceleration.
If Acceleration-Deceleration is higher than nought, then it is usually easier for the acceleration to continue the upward movement (and vice versa in cases when it is below nought).

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## adi *Advance-Decline Indicator*

---

### Description

Advance-Decline Indicator (Technical Analysis)

### Usage

```
ADind(close, high, low, lag = 5)
```

### Arguments

| | |
|---|---|
| close | VECTOR. Close price. |
| high | VECTOR. high price. |
| low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |

### Details

A technical analysis tool that represents the total difference between the number of advancing and declining security prices.
The advance/decline index can provide much more insight into the movements of the market.
In general, rising values of the advance/decline can be used to confirm the likelihood that an upward trend will continue. If the market is up but there are more declining issues than advancing ones, it is usually a sign that the market is losing its breadth and may be getting ready to change direction.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## adrating *Average Directional Rating*

### Description

Compute Average Directional Rating index (Technical Analysis)

### Usage

```
ADrating(close, high, low, lag)
```

### Arguments

| | |
|---|---|
| close | VECTOR. Close price. |
| high | VECTOR. high price. |
| low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## adratio *Advance Decline ratio*

### Description

Compute Advance Decline ratio (Technical Analysis)

### Usage

```
ADratio(X, lag, plot, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Advance Decline Ratio is a market breadth indicator that calculates the ratio of advancing amount to declining amount.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`advdec` *Advance Decline issues*

---

### Description

Compute Advance Decline issues (Technical Analysis)

### Usage

```
AdvDec(X, lag = 5, ret.idx = TRUE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `lag` | INTEGER. Number of lag periods. |
| `ret.idx` | ret.idx |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`ama` *General Adaptive Moving Average*

---

### Description

General Adaptive Moving Average, computed on each column of the input data X.

### Usage

```
ama(X, ar.ord = 1, ma.ord = 1, func = NULL, padding = 0, type = "AMA",
plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| ar.ord | ar.ord |
| ma.ord | ma.ord |
| func | func |
| padding | padding |
| type | type |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## apo *Apo - Absolute price indicator*

---

### Description

Apo - Absolute price indicator

### Usage

```
apo(X, fast.lag = 10, slow.lag = 30, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| fast.lag | fast.lag |
| slow.lag | slow.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

An indicator based on the difference between two exponential moving averages, expressed in absolute terms.
Also known as the MACD indicator, the APO is calculated by subtracting the longer exponential moving average from the shorter exponential moving average.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## apprais *Appraisal ratio*

### Description

`Appraisal`: Calculate Jensen index for a portfolio
`Appraisal.Capm`: Get Jensen index from an object of class "Capm".

### Usage

```
Appraisal(PTF, ...)
## Default S3 method:
Appraisal(PTF, PTF_M, rf = NULL, rfr = 0, ...)
## S3 method for class 'Capm'
Appraisal(PTF, rfr = 0, ...)
```

### Arguments

| | |
|---|---|
| `PTF` | Input portfolio or an object of class "Capm" |
| `PTF_M` | Market/benchmark portfolio |
| `rfr` | risk free rate |
| `rf` | risk free asset |
| `...` | Further arguments to or from other methods |

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

### See Also

`Sharpe`, `Treynor`, `Jensen`

## archlm *ARCH-LM test*

### Description

Compute ARCH-LM test

### Usage

```
Archlm(x, lags, std=FALSE, plot.acf=FALSE)
```

### Arguments

| | |
|---|---|
| `x` | x |
| `lags` | lags |
| `std` | std |
| `plot.acf` | plot.acf |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## armaspc *Arma spectral representation*

---

### Description

Spectral representation based on ARMA models

### Usage

```
Arma.Spec(X, ar_ord = 1, ma_ord = 1, vfreq = NULL)
```

### Arguments

| | |
|---|---|
| X | X |
| ar_ord | ar_ord |
| ma_ord | ma_ord |
| vfreq | vfreq |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## arms *Arms index*

---

### Description

Compute Arms index (Technical Analysis)

### Usage

```
Arms(X, Volume, lag, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| Volume | VECTOR. Asset traded Volume. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## arodown *Aroon Down oscillator*

---

### Description

Compute Aroon Down oscillator (Technical Analysis)

### Usage

```
arodown(X, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## aroon *Aroon oscillator*

### Description

Compute Aroon oscillator (Technical Analysis)

### Usage

```
aroon(X, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## aroud *Aroon Down oscillator*

### Description

Compute Aroon Down oscillator (Technical Analysis)

### Usage

```
aroud(X, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | lag |
| plot | plot |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## aroup *Aroon Up oscillator*

### Description

Compute Aroon Up oscillator (Technical Analysis)

### Usage

```
aroup(X, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## asfs *Convert Yahoo! Data into Financial Series object*

### Description

Convert Yahoo! Data into Financial Series object

### Usage

```
as.fs(X, SName = "", Symbol = "")
```

### Arguments

| | |
|---|---|
| X | X |
| SName | SName |
| Symbol | Symbol |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## aver *Average error*

### Description

Calculate Average error for model evaluation

### Usage

```
av_er(actual, predicted, pc = FALSE)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |
| pc | pc |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bincoef *Binomial coefficient*

### Description

Calculate binomial coefficient

### Usage

```
BinCoef(N, n)
```

### Arguments

| | |
|---|---|
| N | N |
| n | n |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bolband *Bollinger Bands*

### Description

Compute Bollinger Bands (Technical Analysis)

### Usage

```
BolBand(Close, High, Low, fact = 2, win.size = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| fact | fact |
| win.size | win.size |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bolbandb *Bollinger Bands Bandwidth*

### Description

Compute Bollinger Bands Bandwidth (Technical analysis)

### Usage

```
BolBandB(Close, High, Low, fact=2, win.size=5, plot=FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| fact | fact |
| win.size | win.size |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

Bollinger Bandwidth is an important indicator derived from John Bollingers original Bollinger Bands indicator.
Bandwidth is a relative measure of the width of the Bollinger Bands.
The Bollinger Bandwidth equation can be constructed as:
Volatility is high when the Bollinger Bands are farther apart and low when the Bollinger Bands are closer together. Based on the assumption that price (and volatility) generally operates in cycles - periods of low volatility inevitably followed by periods of high volatility and so on and so forth - traders can learn to take advantage of these cycles.
The Bollinger Band Width indicator is the distance between the upper and lower Bollinger Bands. It is a measure of volatility. The Band Width value is higher when volatility is high, and lower when volatility is low. High Band Width values indicate that the current trend may be about to end. Low Band Width values indicate that a new trend may be about to start.
Volatility is high when the Bollinger Bandwidth moves to its high levels and volatility. It is low when Bandwidth moves closer to zero. As a rule, a stocks price and volatility moves in cycles - the price moves up and down and periods of low volatility are replaced by periods of high volatility. In general, periods of high volatility can be noted during down-trends and corrections downward. Periods of low volatility can be observed during up-trends and recoveries. With this knowledge a trader can build a trading system that generates signals based on the volatility technical analysis.
The Bollinger Bandwidth can be used to identify the Squeeze - when the Bandwidth is at its lowest low value within n-periods. Bollinger states that Squeeze could occur before a trend reversal, like the calm before the storm. In this case, trading Buy/Sell signals can be generated on the price breakouts following the Squeeze.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## bolfib *Bollinger Bands - Fibonacci ratio*

---

### Description

Compute Bollinger Bands - Fibonacci ratio (Technical Analysis)

### Usage

```
Bol.Fib(Close, High, Low, win.size = 5, fibo = c(1.618, 2.618, 4.236),
plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| win.size | win.size |

| | |
|---|---|
| `fibo` | fibo |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

---

## `boot` *General bootstrapping function*

---

### Description

General bootstrapping function

### Usage

```
boot(X, nboots = 100, func = NULL, init = NULL,
message = "Bootstrapping...", ...)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `nboots` | nboots |
| `func` | func |
| `init` | init |
| `message` | message |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`bop` *Balance of Power*

## Description

Compute Balance of Power (Technical Analysis)

## Usage

```
Bop(Close, Open, High, Low, smoothed = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `Open` | VECTOR. Open price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `smoothed` | smoothed |
| `...` | Further arguments to or from other methods. |

## Details

The Balance of Power (BOP) indicator measures the strength of the bulls vs. bears by assessing the ability of each to push price to an extreme level.
The Balance of Power indicator is calculated by the formula:
The resulting Balance of Power value is then typically smoothed by a moving average.
The Balance Of Power indicator can be used to identify the systematic buying (accumulation) and systematic selling (distribution) of stock shares.
BOP does it is a good idea to conceptualize how much and what type of trading activities take place in the marketplace. More specifically think of institutional and individual trading activities and the volume of trading that they do.

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

`box3d` *3D box*

## Description

Plotting tools

## Usage

```
box3d(x, y, z, pmat = getProjectionMatrix(), half = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | X axis |
| `y` | Y axis |
| `z` | Z axis |
| `pmat` | pamt |
| `half` | half |
| `...` | Further arguments to or from other methods |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`bpdlind` *BPDL indicator*

## Description

Compute BPDL indicator (Technical Analysis)

## Usage

```
BPDLind(Close, lag = 1, smoothed = TRUE, slag = 5)
```

## Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `lag` | INTEGER. Number of lag periods. |
| `smoothed` | smoothed |
| `slag` | slag |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`breadth` *Breadth trusth indicator*

---

### Description

Compute Breadth trusth indicator (Technical Analysis)

### Usage

```
Breadth(X, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `lag` | INTEGER. Number of lag periods. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Details

The Breadth Thrust indicator is a ten period simple moving average of the Advance/Decline Line Breadth. A Breadth Thrust formation is defined as when the Breadth Thrust indicator moves from below 40 above 61.5 percent within a ten day period. This indicates that the market is rapidly moving from an oversold condition to a strong up trend. A Breadth Thrust formations are rare and are a very good indicator of a bull market.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bromot *Browniam motion*

### Description

Simulate a standard Brownian motion

### Usage

```
BroMot(nsim, T, S0 = 0, mi = 0, sigma = 1,
geom = TRUE, same.rnd = TRUE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| nsim | nsim |
| T | T |
| S0 | S0 |
| mi | mi |
| sigma | sigma |
| geom | geom |
| same.rnd | same.rnd |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bromot2d *2-dimensional Browniam motion*

### Description

Simulate a bi-dimensional standard Brownian motion

### Usage

```
BroMot2D(nsim, T, S0, mi, sigma, geom = TRUE,
same.rnd = FALSE, laydisp = NULL, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| nsim | nsim |
| T | T |
| S0 | S0 |
| mi | mi |
| sigma | sigma |
| geom | geom |
| same.rnd | same.rnd |
| laydisp | laydisp |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## bsfml *Black & Scholes formula*

---

### Description

Black & Scholes analytical formula

### Usage

```
BS.formula(type = c("call", "put"))
```

### Arguments

| | |
|---|---|
| type | type |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bsgreeks *Black & Scholes greeks*

### Description

Calculate analytically Black & Scholes greeks

### Usage

```
BS.greeks(X = NULL, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## bslmpvol *Black & Scholes Implied volatility*

### Description

Calculate Black & Scholes Implied volatility

### Usage

```
BS.ImpVol(P, under, strike, rfr, sigma, maty,
yield, interval = c(-20, 20),
calc.type =c("standard", "lognorm", "gammarec"),
opt.type = c("call", "put"))
```

### Arguments

| | |
|---|---|
| P | P |
| under | under |
| strike | strike |
| rfr | rfr |
| sigma | sigma |
| maty | maty |
| yield | yield |
| interval | interval |
| calc.type | calc.type |
| opt.type | opt.type |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## bsmomt *Black & Scholes moments*

---

### Description

Calculate first four moments for Black & Scholes

### Usage

```
BS.moments(BS = NULL, under, rfr, sigma, yield, maty)
```

### Arguments

| | |
|---|---|
| BS | BS |
| under | under |
| rfr | rfr |
| sigma | sigma |
| yield | yield |
| maty | maty |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## bsprice *Black & Scholes price generic*

---

### Description

Generic method for Black & Scholes price

### Usage

```
BS.price(under, ...)
## Default S3 method:
BS.price(under, strike, rfr, sigma, maty, yield, calc.type =c("standard",
"lognorm","gammarec"), opt.type = c("call", "put"), ...)
```

### Arguments

| | |
|---|---|
| `under` | Underlying asset price. |
| `strike` | Strike price. |
| `rfr` | Risk free rate. |
| `sigma` | Volatility. |
| `maty` | Maturity. |
| `yield` | Yield |
| `calc.type` | Calculation type. |
| `opt.type` | Option type. |
| `...` | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

---

## `buypre` *Buying pressure indicator*

---

### Description

Compute Buying pressure indicator (Technical Analysis)

### Usage

```
buypre(Close, Low, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `Low` | VECTOR. Low price. |
| `lag` | INTEGER. Number of lag periods. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## calcprs *Calculated Pairs*

### Description

Calculated the number of pairs based on the results of a classification model.

### Usage

```
CalcPairs(target, pred, segm_fact = 0.002)
```

### Arguments

| | |
|---|---|
| target | target |
| pred | pred |
| segm_fact | segm_fact |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## capm *Capm - default method*

### Description

Default method for CAPM

### Usage

```
Capm(PTF, ...)
## Default S3 method:
Capm(PTF, PTF_M, rf = NULL, rfr = NULL, ...)
```

### Arguments

| | |
|---|---|
| PTF | Prtfolio. |
| PTF_M | Market portfolio. |
| rf | Risk free asset. |
| rfr | Risk free rate. |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`cci` *Commodity channel index*

---

### Description

Compute Commodity channel index (Technical Analysis)

### Usage

```
cci(High, Low, Close, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Close` | VECTOR. Close price. |
| `lag` | INTEGER. Number of lag periods. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`cciv2` *Commodity channel index v02*

---

### Description

Compute Commodity channel index v02 (Technical Analysis)

### Usage

```
cci.v2(High, Low, Close, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Close` | VECTOR. Close price. |
| `lag` | INTEGER. Number of lag periods. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## chaikin *Chaikin oscillator*

---

### Description

Compute Chaikin oscillator (Technical Analysis)

### Usage

```
chaikin(Close, High = NULL, Low = NULL,
Vol = NULL, fast.lag = 3, slow.lag = 10,
plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Vol | VECTOR. Asset traded Volume. |
| fast.lag | fast.lag |
| slow.lag | slow.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## chaosacc *Chaos Accelerator oscillator*

### Description

Compute Chaos Accelerator oscillator (Technical Analysis)

### Usage

```
chaosAcc(X)
```

### Arguments

| | |
|---|---|
| X | X |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## chist *Custom histogram function*

### Description

Custom histogram function

### Usage

```
chist(x, nclass = min(max(round(NROW(x)/10), 10),
NROW(x)), density = c("kernel", "normal"),
kernel = c("gaussian", "epanechnikov", "rectangular",
"triangular", "biweight", "cosine", "optcosine"),
theme.params = getCurrentTheme(), main = "Histogram and Kernel Density Estimatic
xtitle = NULL, ytitle = NULL, legend = NULL,
show.legend = TRUE, normalised = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| nclass | nclass |
| density | density |
| kernel | kernel |
| theme.params | theme.params |
| main | main |

| | |
|---|---|
| xtitle | xtitle |
| ytitle | ytitle |
| legend | legend |
| show.legend | show.legend |
| normalised | normalised |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## chvol *Chaikin volatility indicator*

---

### Description

Compute Chaikin volatility indicator (Technical Analysis)

### Usage

```
Ch.vol(High, Low, Close, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Close | VECTOR. Close price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Chaikin Volatility indicator is the rate of change of the trading range.
The indicator defines volatility as a increasing of the difference between the high and low. A rapid increases in the Chaikin Volatility indicate that a bottom is approaching. A slow decrease in the Chaikin Volatility indicates that a top is approaching.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

cleanup *Clean memory*

## Description

Cleanup environment and (optionally) performs Garbage Collection

## Usage

```
cleanup(keep = c(), env = parent.frame(), gc = FALSE)
```

## Arguments

| | |
|---|---|
| keep | CHARACTER. Vector of variables to keep in memory. |
| env | Environment from which objects are removed. Defaults to the environment from which this function is called. |
| gc | LOGICAL. If TRUE, garbage collection is performed to release memory. (Default = TRUE) |

## Value

VOID

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

clust *Time series clusters*

## Description

Create a simple cluster partition of a time series

## Usage

```
TSClust(x, ...)

## Default S3 method:
TSClust(x, y=NULL, n_clust=5,
bk.type=c("quantile","volatility","uniform","custom"),
pc_vol=0.1, win.size=10,  custom_breaks=NULL,
lab.dig=0, ...)

## S3 method for class 'TSClust'
summary(object, funs = summary, ...)
```

```
## S3 method for class 'TSClust'
plot(x, smooth=FALSE, ...)
```

### Arguments

| | |
|---|---|
| x, object | Univariate time series or an object of class "TSClust" |
| y | y |
| n_clust | number of cluster |
| bk.type | Breaks type |
| custom_breaks | custom_breaks |
| lab.dig | lab.dig |
| funs | function to run inside summary.TSClust |
| smooth | smooth |
| pc_vol | pc_vol |
| win.size | win.size |
| ... | further arguments accepted by "funs" |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## clv *Close Location value oscillator*

---

### Description

Compute Close Location value oscillator (Technical Analysis)

### Usage

```
clv(Close, High = NULL, Low = NULL, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`cmf` *Chaikin Money Flow*

---

### Description

Compute Chaikin Money Flow (Technical Analysis)

### Usage

```
cmf(Close, Low, High, Volume, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `Low` | VECTOR. Low price. |
| `High` | VECTOR. High price. |
| `Volume` | Volume |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods |

### Details

Chaikin Money Flow measures the amount of Money Flow Volume over a specific period.
Money Flow Volume forms the basis for the Accumulation Distribution Line. Chaikin Money Flow sums Money Flow Volume for a specific look-back period, typically 20 or 21 days. The resulting indicator fluctuates above/below the zero line just like an oscillator. Chartists weigh the balance of buying or selling pressure with the absolute level of Chaikin Money Flow. Chartists can also look for crosses above or below the zero line to identify changes on money flow. The Chaikin Money Flow compares the total volume over the last n time periods to the total of volume times the Closing Location Value (CLV) over the last n time periods. The CLV calculates where the issue closes within its trading range. When the Chaikin Money Flow is above 0.25 it is a bullish signal, when it is below -0.25, it is a bearish signal. If the Chaikin Money Flow remains below zero while the price is rising, it indicates a probable reversal.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## cmof *Chande Momentum Oscillator*

### Description

Compute Chande Momentum Oscillator (Technical Analysis)

### Usage

```
cmof(X, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Chande Momentum Oscillator is a modified RSI. Where the RSI divides the upward movement by the net movement (up / (up + down)), the CMO divides the total movement by the net movement ((up - down) / (up + down)). There are several ways to interpret the CMO. Values over 50 indicate overbought conditions, while values under -50 indicate oversold conditions. High CMO values indicate strong trends. When the CMO crosses above a moving average of the CMO, it is a buy signal, crossing down is a sell signal.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## cofit *Cornish Fisher Transformation*

### Description

Cornish Fisher Transformation

### Usage

```
cofit(X, p, k = NULL, s = NULL)
```

### Arguments

| | |
|---|---|
| X | Input matrix/sequence. Sequences are treated as one column matrices. |
| p | vector of probability threshold (interval [0, 1]) |
| k | kurtosis (DEFAULT = NULL -> becomes kurt(X)) |
| s | skewness (DEFAULT = NULL -> becomes skew(X)) |

### Value

A matrix length(trsh) by NCOL(X) of computed quantiles

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## colinprs *Co-Linearity analysis*

---

### Description

This function performs a Co-Linearity analysis between the columns of X: Correlation factors between columns are computed,and pairs of columns with a correlation factor higher than a specified threshold are returned.

### Usage

```
colin.pairs(X, trsh = 0.8)
```

### Arguments

| | |
|---|---|
| X | X |
| trsh | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## colinred *Co-Linearity reduction*

### Description

Perform a cross Co-Linearity analysis between the columns of Y and X, and for each Yi returns a reduced set of columns of X obtained after removing those columns of X that are too correlated (one for each co-linear pair). In the removal process, those columns of X that are most correlated to Yi are kept.

### Usage

```
colin.reduce(Y, X, max.iter = 100, trsh = 0.85)
```

### Arguments

| | |
|---|---|
| Y | Y |
| X | X |
| max.iter | max.iter |
| trsh | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## combfs *Combine Multiple Financial Series*

### Description

Combine Multiple Financial Series

### Usage

```
combine.fs(..., which = "Close")
```

### Arguments

| | |
|---|---|
| ... | ... |
| which | which |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## confmat *Confusion matrix*

### Description

Build confusion matrix based on the results of a classification model.

### Usage

```
confusionM(target, ...)
## Default S3 method:
confusionM(target, pred, th, ...)
## S3 method for class 'scorecard'
confusionM(target, th, ...)
```

### Arguments

| | |
|---|---|
| target | target |
| pred | pred |
| th | th |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## cplot *2-Dimensional Plotting*

### Description

Workhorse function for automatic plotting

### Usage

```
cplot(X, base = NULL, xrange = NULL, yrange = NULL, theme.params =
getCurrentTheme(), xtitle = "" , xlabels = NULL , ytitle = "" , ylabels =
NULL , ytitle2 = "" , ylabels2 = NULL, show.xlabels = TRUE , show.ylabels =
TRUE , main = "" , legend = NULL , legend.col = theme.params[["col"]],
show.legend = TRUE , shaded = FALSE, grid = TRUE , overrides = list(...),
new.device = FALSE,append = FALSE, multicolor = FALSE , ...)
```

### Arguments

| | |
|---|---|
| X | x coordinates of the plot |
| base | y coordinates of the plot |
| xrange | x axis range |
| yrange | y axis range |
| theme.params | Retrieve RAdamant graphical theme |
| xtitle | title for the x axis |
| xlabels | labels for x tick marks |
| ytitle | title for the y axis |
| ylabels | labels for y tick marks |
| ytitle2 | title for the right-y axis |
| ylabels2 | labels for right-y tick marks |
| show.xlabels | LOGICAL. Show labels on the x axis |
| show.ylabels | LOGICAL. Show labels on the y axis |
| main | Main title for the plot |
| legend | Add the legend |
| legend.col | Colors for the elements in the legend |
| show.legend | LOGICAL. Display the legend in the plot |
| shaded | LOGICAL. Insert shaded under the plot |
| grid | LOGICAL. Draw a grid |
| overrides | overrides list |
| new.device | open new.device window |
| append | append to existing plot |
| multicolor | multiple colors |
| ... | additional arguments for generic funciotn "plot" |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## cplot3d *3-Dimensional plotting*

---

### Description

Workhorse function for 3D automatic plotting

## Usage

```
cplot3d(x, y, z, fill = c("simple", "colormap", "gradient"),
main = "", xtitle = "", ytitle = "", ztitle = "",
xlim = range(x) + 0.1*diff(range(x))*c(-1, 1),
ylim = range(y) + 0.1*diff(range(y))*c(-1, 1),
zlim = range(z, na.rm = TRUE) + 0.1*diff(range(z, na.rm = TRUE))*c(-1, 1),
xlabels = NULL, ylabels = NULL,
zlabels = NULL, pre = NULL, post = NULL,
theme.params = getCurrentTheme(),
overrides = list(...), new.device = FALSE,
append = FALSE, axis = TRUE,
show.labels = TRUE, show.xlabels = TRUE,
show.ylabels = TRUE, show.zlabels = TRUE
, ...)
```

## Arguments

| | |
|---|---|
| x | x coordinates for the plot |
| y | y coordinates for the plot |
| z | z coordinates for the plot |
| fill | fill |
| main | main |
| xtitle | xtitle |
| ytitle | ytitle |
| ztitle | ztitle |
| xlim | xlim |
| ylim | ylim |
| zlim | zlim |
| xlabels | xlables |
| ylabels | ylabels |
| zlabels | zlabels |
| pre | pre |
| post | post |
| theme.params | theme.params |
| overrides | overrides |
| new.device | new.device |
| append | append |
| axis | axis |
| show.labels | show.labels |
| show.xlabels | show.xlabels |
| show.ylabels | show.ylabels |
| show.zlabels | show.zlabels |
| ... | Further arguments to or from other methods |

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## cramv *Cramers V*

### Description

Calculate Cramers V

### Usage

```
cramv(x, y)
```

### Arguments

| | |
|---|---|
| x | x |
| y | y |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## crbtree *CRR Binomial Tree*

### Description

Option evaluation with Cox, Rossand and Rubinstein Binomial Tree

### Usage

```
CRR.BinTree(Nsteps, under, strike, rfr,
sigma, maty, yield, life, ret.steps = FALSE)
```

### Arguments

| | |
|---|---|
| Nsteps | Nsteps |
| under | under |
| strike | strike |
| rfr | rfr |
| sigma | sigma |
| maty | maty |
| yield | yield |
| life | life |
| ret.steps | ret.steps |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`croscf` *Cross correlation function*

---

### Description

Compute the cross correlation function for each pairs of variables (Yi Xj)

### Usage

```
cross.ccf(Y, X, ...)
## Default S3 method:
cross.ccf(Y, X, lag.max = 10, ci = 0.95, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| `Y` | Matrix of data series (one column per variable) |
| `X` | Matrix of data series (one column per variable) |
| `lag.max` | Max lag to be computed by the cross correlation function (DEFAULT: 10) |
| `ci` | Confidence Interval (DEFAULT: 0.95) |
| `plot` | LOGICAL. If TRUE, results are plotted. |
| `...` | additional parameters accepted by the function plot.cross.ccf. |

### Value

A list of Ny*Nx cross correlation objects of the class "cross.acf"

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## crosplot *Cross Plot*

### Description

Plot the input dependent variable Y versus each input independent variable X

### Usage

```
cross.plot(Y, X, theme.params = getCurrentTheme(),
xlabels = NULL, two.axis = TRUE,
shaded.first = FALSE, overrides = NULL)
```

### Arguments

| | |
|---|---|
| Y | serie of the dependent variable |
| X | Matrix containing all independent variables (one column per variable) |
| theme.params | theme parameters (DEFAULT: getCurrentTheme()) |
| xlabels | serie of the lables associated to the rows of X (i.e. Time libels)(DEFAULT: NULL) |
| two.axis | LOGICAL. If TRUE, series are plotted on two axis (two scales). |
| shaded.first | LOGICAL. If TRUE, the variable Y is shaded. |
| overrides | list of parameters to override the theme. Must match by name the parameters defined by the theme (DEFAULT: NULL) |

### Value

VOID

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## crscolin *Cross collinearity*

### Description

Perform a cross Co-Linearity analysis between the columns of Y and X: Correlation factors between each column Yi and all columns of X are calculated for different time lags.Also pairs of columns of X with a correlation factor higher than a specified threshold are returned.

### Usage

```
cross.colin(Y, X, max.lag = 8, trsh = 0.8)
```

### Arguments

| | |
|---|---|
| Y | Y |
| X | X |
| max.lag | max.lag |
| trsh | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## cumfun *Cumulative functions*

---

### Description

Cumulative max / min / Mean / Standard Deviation / Variance / sum on each column of the input matrix.

### Usage

```
cumMax(X, lag = 0, padding = NA, na.rm = FALSE)
```

### Arguments

| | |
|---|---|
| X | Input matrix/sequence |
| lag | vector of integer lags. If lag >= 0 data are shifted to the right, else to the left. (DEFAULT = 0) |
| padding | value used to initialise the output matrix (DEFAULT = NA) |
| na.rm | LOGICAL. If TRUE, N-lag entries are removed from the output. Also NA in the input are replaced by -Inf (DEFAULT = FALSE) |

### Details

Sequences are treated as one-column matrices

### Value

A matrix of cumulative maximums of X. Number of rows depends on the na.rm parameter. Number of columns is NCOL(X)

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

lew

## decimals *Count decimal*

### Description

Count decimal

### Usage

```
decimals(x, max.digits = 10, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| max.digits | max.digits |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## decscal *Decimal scale*

### Description

Compute decimal scale of a vector

### Usage

```
Decscal(x, scale = 0.1)
```

### Arguments

| | |
|---|---|
| x | x |
| scale | scale |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

dema *Double EMA*

## Description

Compute multiple Double EMA on the input data, one for each column of X[, i] and window size win.size[j]

## Usage

```
dema(X, win.size = NROW(X), plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| X | X |
| win.size | win.size |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.
DEMA is a weighted combination of EMA: 2*EMA(X) - EMA(EMA(X)).
Smoothing factor: lambda = 2/(win.size+1).

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## See Also

ema

demark *DeMark indicator*

## Description

Compute DeMark indicator (Technical Analysis)

## Usage

```
demark(High, Low, Close, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Close` | VECTOR. Close price. |
| `lag` | INTEGER. Number of lag periods. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

---

## `dgev` *Generalised Extreme Value (GEV)*

---

### Description

Generalised Extreme Value (GEV) - Density function

### Usage

```
dgev(X, mu = 0, xi = 0.1, sigma = 1)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `mu` | mu |
| `xi` | xi |
| `sigma` | sigma |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## dgpd *Generalised Pareto Distribution (GPD)*

### Description

Generalised Pareto Distribution (GPD) - Density function

### Usage

```
dgpd(X, xi = 0.1, sigma = 1, trsh = 0)
```

### Arguments

| | |
|---|---|
| X | X |
| xi | xi |
| sigma | sigma |
| trsh | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## dma *Derivative Moving Averages*

### Description

Compute multiple Derivative Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
dma(X, fast.win = 5, slow.win = 28, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| fast.win | fast.win |
| slow.win | slow.win |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
Formula: 100 * ( movMax(SMA(X, fast.win), slow.win) - movMin(SMA(X, fast.win), slow.win)) / X.

### Value

A object of class 'ma' with attributes type = "DMA" and 'win.size' as from the corresponding input parameters [fast.win,slow.win]:
- matrix of size NROW(X) by NCOL(X) where each column is the moving average of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

sma

---

## dpo *Detrended price oscillator*

---

### Description

Compute Detrended price oscillator (Technical Analysis)

### Usage

```
dpo(Close, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Detrended Price Oscillator removes the trend in prices by subtracting a moving average of the price from the price.
The Detrended Price shows cycles and overbought/oversold conditions. Note that the calculation shifts the results (shift = term / 2 + 1) periods, so the last shift periods will be zero.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`drawdown` *Financial Drawdown*

## Description

Drawdown risk analysis

## Usage

```
drawdown(x, ...)
## Default S3 method:
drawdown(x, FUN = max, relative = FALSE, plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | x |
| `FUN` | FUN |
| `relative` | realtive |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

---

`dropn` *Drop N Possible Terms to a Linear Regression Model*

## Description

Drop N Possible Terms to a Linear Regression Model

## Usage

```
dropn(mod, N = 1, ...)
```

## Arguments

| | |
|---|---|
| `mod` | mod |
| `N` | N |
| `...` | Further arguments to or from other methods. |

## Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`edgefact` *Edge Factor (B&S)*

---

### Description

Edgeworth adaption factors

### Usage

```
EdgeFact(x, s, k)
```

### Arguments

| | |
|---|---|
| x | x |
| s | s |
| k | k |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`edwdist` *Edgeworth distribution*

---

### Description

Simulate empirical Edgeworth distribution

### Usage

```
EdgeWorthDist(init, Nsteps, p=0.5)
```

### Arguments

| | |
|---|---|
| init | init |
| Nsteps | Nsteps |
| p | p |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## edwprice *Edgeworth option price*

### Description

Option evaluation with Edgeworth adapted Binomial Tree

### Usage

```
Edgeworth.price(init, under, strike, rfr, sigma, maty, yield)
```

### Arguments

| | |
|---|---|
| init | init |
| under | under |
| strike | strike |
| rfr | rfr |
| sigma | sigma |
| maty | maty |
| yield | yiels |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## ema *Exponential Moving Average*

### Description

Compute multiple Exponential Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
ema(X, win.size = NROW(X), plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = 10). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by the function Mmovav. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
Smoothing factor: lambda = 2/(win.size+1).

### Value

A object of class 'ma' with attributes type = "EMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## emat *Trend corrected Exponential Moving Averages*

---

### Description

Compute multiple Trend corrected Exponential Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
emat(X, win.size = NROW(X), alpha = 0.1, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| alpha | weight for the trend correction (DEFAULT: 0.1) |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function ema. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
EMAT is a dynamic model regulated by the smoothing factors lambda = 2/(win.size+1) and alpha.

### Value

A object of class 'ma' with attributes type = "EMAT", 'lambda' and 'alpha':
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

eom *Ease of Movement oscillator*

---

### Description

Compute Ease of Movement oscillator (Technical Analysis)

### Usage

```
eom(Close, High = NULL, Low = NULL, Vol = NULL, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Vol | VECTOR. Asset traded Volume. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

epma *end Point Moving Averages*

---

### Description

Computes multiple End-Points Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
epma(X, win.size = 10, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable) |
| win.size | Vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by the function Movav |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
EPMA Weights are given by a win.size-long line with angular coefficient = -3 and intercept = 2*win.size-1

### Value

A object of class 'Movav' with attributes type = "EPMA" and 'win.size' as from the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

Movav

---

## erf *Elder Ray force*

---

### Description

Compute Elder Ray force (Technical Analysis)

### Usage

```
erf(Close, High = NULL, Low = NULL, lag = 13, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## erfi *Elder Ray force index*

---

### Description

Compute Elder Ray force index (Technical Analysis)

### Usage

```
erfi(X, Volume, lag = 13, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| Volume | VECTOR. Asset traded Volume. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## extrbrk *Extract breaks*

---

### Description

extract specificied break from an object of class Factorise

### Usage

```
extrBreak(var, Factors)
```

### Arguments

| | |
|---|---|
| var | var |
| Factors | Factors |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## extrdd *Maximum / Minimum drawdown*

---

### Description

Calculate Mximum / Minimum DrawDown

### Usage

```
ExtremeDD(DD, FUN, lag = 1, rolling = FALSE, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| DD | OBJECT of class "drawdown" |
| FUN | FUN |
| lag | INTEGER. Number of lag periods. |
| rolling | rolling |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## fact2woe *Factor to Weight of Evidence*

---

### Description

Transform factorise data to weight of evidence

### Usage

```
factor2woe(segm, woe)
```

### Arguments

| | |
|---|---|
| segm | INTEGER / VECTOR. Number of segments to factorise numerical variables. |
| woe | woe |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## factor *Factorise variable*

---

### Description

Factorise numerical variables according with defined bins

### Usage

```
Factorise(X, nseg,
seg.type = c("freq_equal", "width_equal"),
na.replace = NULL)
```

### Arguments

| | |
|---|---|
| X | X |
| nseg | INTEGER / VECTOR. Number of segments to factorise numerical variables. |
| seg.type | CHARACTER. Type of segments to create. (Default = "equal frequencies") |
| na.replace | CHARACTER / NUMERIC. Value to replace missing.<br>If NULL missing values are not considered in the computation. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`fft` *Custom Fast Fourier transformation*

## Description

Compute Custom Fast Fourier transformation

## Usage

```
FFT(x, ...)

## Default S3 method:
FFT(x, Fs = 1, half = FALSE,
window = NULL, plot = TRUE, optimised = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `x` | x |
| `Fs` | Fs |
| `half` | half |
| `window` | window |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `optimised` | optimised |
| `...` | Further arguments to or from other methods |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`finplot` *Plot financial time series*

## Description

Plot financial time series

### Usage

```
fin.plot(X, top.vars = c("Close", "High", "Low"),
bottom.vars = "Volume", style = c("default", "candlestick"),
snames = attr(X, "SName"), xlabels = rownames(X),
main = "", main2 = "",
ytitle = "", ytitle2 = "",
theme.top = getCurrentTheme(),
overrides = list(...),
theme.bottom = getCurrentTheme(),
overrides2 = NULL, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| top.vars | top.vars |
| bottom.vars | bottom.vars |
| style | style |
| snames | snames |
| xlabels | xlabels |
| main | main |
| main2 | main2 |
| ytitle | ytitle |
| ytitle2 | ytitle2 |
| theme.top | theme.top |
| overrides | overrides |
| theme.bottom | theme.bottom |
| overrides2 | overrides2 |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## firsthit *First Hit time barrier (Brownian motion)*

---

### Description

Calculate expected time of the First Hitting for a Brownian motion

### Usage

```
FirstHit(B, S0, mi, geom = FALSE, sigma = NULL)
```

### Arguments

| | |
|---|---|
| B | B |
| S0 | S0 |
| mi | mi |
| geom | geom |
| sigma | sigma |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## fmeas *Four Measures indexes*

---

### Description

Calculate the Four Measures indexes

### Usage

```
FourMeasures(PTF, ...)
## Default S3 method:
FourMeasures(PTF, PTF_M, rf = NULL, rfr = 0, ...)
## S3 method for class 'Capm'
FourMeasures(PTF, rfr = 0, ...)
```

### Arguments

| | |
|---|---|
| PTF | Input portfolio or an object of class "Capm" |
| PTF_M | Market/benchmark portfolio |
| rfr | risk free rate |
| rf | risk free asset |
| ... | Further arguments to or from other methods |

### Value

Return a matrix containing the values for the following indexes: Sharpe, Treynor, Jensen and Appraisal

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

Sharpe, Treynor, Jensen, Appraisal

`fmlmreg` *Extract formula from regression object*

## Description

Extract formula from regression ("reg" / "mreg") object

## Usage

```
## S3 method for class 'reg'
formula(x, ...)
## S3 method for class 'mreg'
formula(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class "reg" / "mreg" |
| `...` | Further arguments passed to or from other methods |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`forcidx` *Force index*

## Description

Compute Force index (Technical Analysis)

## Usage

```
forcidx(X, Volume, lag = 5, sth = TRUE,
sth.lag = 13, mov = sma, plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `X` | X |
| `Volume` | Volume |
| `lag` | INTEGER. Number of lag periods. |
| `sth` | sth |
| `sth.lag` | sth.lag |
| `mov` | mov |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from othermethods |

### Details

The Force Index measures the force of bulls behind every rally and of bears behind every decline. The Force Index attempts to quantify the force of every price move according to its direction, distance, and trading volume.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## frama *Fractal Moving Average*

---

### Description

Fractal Moving Average, computed on each column of the input data X and for each pair (fast.win[i], slow.win[i]).

### Usage

```
frama(X, win.size = 10, tau = 4.6,
keep.lambda = FALSE, keep.ER = FALSE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of window sizes (lags) (DEFAULT = 10). |
| tau | controls how the smoothing factor lambda is calculated (lambda = exp(tau*log(ER))) (DEFAULT = 4.6). |
| keep.lambda | LOGICAL. If TRUE, adaptive smoothing factor lambda is returned as an attribute (DEFAULT = FALSE). |
| keep.ER | LOGICAL. If TRUE, adaptive Efficiency Ratio ER is returned as an attribute (DEFAULT = FALSE). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters for future development. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A object of class 'Movav' with attributes type = "FRAMA", 'lambda' and 'ER' as required and 'win.size' and 'tau' given by the corresponding input parameters:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

`freqwind` *Smoothing windows*

---

### Description

`barthann`: Compute Barthann window
`bartlet`: Compute Bartlet window
`blackman`: Compute Blackman window
`gauss`: Compute Gauss window
`lanczos`: Compute Lanczos window
`triangle`: Compute Triange window
`hamming`: Compute Hamming window
`hann`: Compute Hann window
`cosine`: Compute Cosine window

### Usage

```
barthann(N, normalized = TRUE, alpha = 0.38)
bartlet(N, normalized = TRUE)
blackman(N, normalized = TRUE, alpha = 0.16)
cosine(N, normalized = TRUE)
gauss(N, normalized = TRUE, sigma = 0.5)
hamming(N, normalized = TRUE)
hann(N, normalized = TRUE)
kaiser(N, normalized = TRUE, alpha = 3)
lanczos(N, normalized = TRUE)
triangle(N, normalized = TRUE)
```

### Arguments

| | |
|---|---|
| `N` | Window length |
| `normalized` | LOGICAL. If TRUE, window is normalised to have unitary norm |
| `alpha` | alpha |
| `sigma` | sigma |

### Value

The window samples

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## See Also

FFT

---

fresvar *Fiited / Residual for VAR*

---

## Description

Get Fitted values and Residuals from a VAR model

## Usage

```
## S3 method for class 'VecAr'
fitted(object, Coefs, ar.lags, ...)
```

## Arguments

| | |
|---|---|
| object | object |
| Coefs | Coefs |
| ar.lags | ar.lags |
| ... | Further arguments to or from other methods |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

---

fsevecar *VAR Forecast Standard Error*

---

## Description

Compute forecast standard error for VAR model

## Usage

```
FSE.VecAr(X, steps, ...)
```

## Arguments

| | |
|---|---|
| X | X |
| steps | steps |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## fulp *Full price*

---

### Description

Compute Full price (Technical Analysis)

### Usage

```
fullP(Close, Open, High, Low, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| Open | VECTOR. Open price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## funcomx *Function comment*

---

### Description

Given an input file, this functions created an index based commented version of the file.

### Usage

```
func.comment.idx(control.df =
data.frame(FNAME = c(), FCODE = c(),
AREA = c(), SECTION = c(), CLASS = c()),
infile = NULL, incode = NULL, outfile = NULL, max.dgt = 3)
```

### Arguments

| | |
|---|---|
| control.df | List of function names. See Details |
| infile | Input file (Full path: Mandatory). |
| incode | Input code array (Alternative to infile: Mandatory). Each entry is considered to be a line of code. |
| outfile | Output commented file (Full path: Optional). If provided, an output file is generated. |
| max.dgt | Controls the number of digits to be used on each section of the comment. |

### Details

This data frame is a list of function names:

- FNAME = Name of the function
- FCODE = code identifier for the function. (a-Z)(0-9).
- AREA = Macro area (Description) classification for the function.
- SECTION = Section (Description) classification for the function (Sub-AREA)
- CLASS = The class of the returned object.

### Value

String array where every entry is a line of code. Each original line of the input code is preceded by a special comment.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### Examples

```
#### EXAMPLE #####
tst = data.frame(FNAME = c("sd", "lm")
, FCODE = c("SD", "LM")
, AREA = c("s5", "s2")
, SECTION = c("s1", "s1")
, CLASS = c("c1", "c2")
);

incode = rbind(paste("sd =", as.character(deparse(args(sd)))[1])
, as.matrix(deparse(body(sd)))
, ""
, ""
, paste("lm =", as.character(deparse(args(lm)))[1])
, as.matrix(deparse(body(lm)))
   )
func.comment.idx(tst, incode = incode, max.dgt=3)
```

`funlcnt` *Function line counting*

## Description

Given a package name or a list of functions, for each function X in the package or the list it counts the lines of code, the number of subcalls made to any other function Y of the list/package and the number of other functions that make calls to the function X.
Results are plotted if requested.

## Usage

```
func.line.cnt(package = NULL, plot = TRUE,
qtz.type = "NONE", qtz.nbins = 10, qtz.cutoff = 30)
```

## Arguments

| | |
|---|---|
| `package` | CHARACTER. Single name of the package to load or array list of function names. |
| `plot` | LOGICAL. If TRUE, results are plotted on bar charts. |
| `qtz.type` | CHARACTER. qtz.type = "NONE" \| "LINEAR" \| "LOG". |
| `qtz.nbins` | INTEGER. Number of bins to be computed. Used only when qtz.type != "NONE". (Default = 10) |
| `qtz.cutoff` | Used only when qtz.type = "LOG". (Default = 30) |

## Details

Parameter "qtz.type" is Case Insensitive. It states the ypype of quantizzation to be used to set bin size for the barchart plotting the distribution of lines of code. Values:

- If "NONE", bin size is set to 1.
- If "LINEAR", qtz.nbins equispaced intervarls are computed.
- If "LOG", qtz.nbins log-spaced intervals are computed based on qtz.cutoff.

Parameter "qtz.bins": qtz.nbins equispaced intervals are computed on a log(x/qtz.cutoff) scale. This creates more intervals in the range 0 < x < qtz.cutoff.

## Value

Data frame containing the stats for each function in the input list/package:

- fcn.name = Name of the function
- fcn.lines = Number of lines of code
- fcn.subcalls = Calls made to other functions
- fcn.called = Number of function calling the function

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## fwmovav *Front Weighted Moving Averages*

### Description

fw1: Computes multiple Front Weighted 32 Day Moving Averages on the input data, one for each column X[, i].
fw2: Computes multiple Front Weighted 18 Day Moving Averages on the input data, one for each column X[, i].
fw3: Computes multiple Front Weighted 2 Day Moving Averages on the input data, one for each column X[, i].

### Usage

```
fw1(X, plot = FALSE, ...)
fw2(X, plot = FALSE, ...)
fw3(X, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function movav. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A object of class 'ma' with attributes type = "FW1/2/3" and 'weights' given by the FW1/2/3 filter weights:
- matrix of size NROW(X) by NCOL(X) where each column is the moving average of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## garch *Garch*

### Description

Estimate Garch models

### Usage

```
Garch(x, ...)
## Default S3 method:
Garch(x, Y=NULL, order=c(alpha=1,beta=1), phi=0, delta=0,
type=c("garch","mgarch","tgarch","egarch"), prob=c("norm","ged","t"), ...)
```

### Arguments

| | |
|---|---|
| x | Univariate time series, usaully returns |
| Y | Exogenous regressors for the Mean Equation |
| order | Garch order |
| type | Garch type. |
| prob | Probability density for the innovations. |
| phi | Phi pars |
| delta | Delta pars |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gdema *Generalised Double EMA*

---

### Description

Compute multiple Generalised Double EMA on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
gdema(X, win.size = NROW(X), alpha = 0.7, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| alpha | weight in the interval [0, 1]. (DEFAULT: 0.7) |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function ema. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
GDEMA is a weighted combination of EMA and DEMA: alpha*DEMA(X) + (1-alpha) * EMA(X).
Smoothing factor: lambda = 2/(win.size+1).

### Value

A object of class 'ma' with attributes type = "GDEMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

## getacfci *Normal confidence intervals for correlation*

---

### Description

Compute the Normal confidence intervals for correlation and partial autocorrelation data

### Usage

```
get.acf.ci(X, ci = 0.95)
```

### Arguments

| | |
|---|---|
| X | X |
| ci | ci |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`getfs` *Download Financial Series data from Yahoo!*

## Description

Download Financial Series data from Yahoo!

## Usage

```
get.fs(symbol = NULL, SName = NULL,
from = as.Date("1950-01-01"), to = Sys.Date(),
strip.spaces = TRUE, strip.char = ".")
```

## Arguments

| | |
|---|---|
| `symbol` | symbol |
| `SName` | SName |
| `from` | from |
| `to` | to |
| `strip.spaces` | strip.spaces |
| `strip.char` | strip.char |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`getlmwgh` *Extract weights percentages of the coefficients of a linear model*

## Description

Extract weights percentages of the coefficients of a linear model

## Usage

```
get.lm.weights(mod, pct = FALSE)
```

## Arguments

| | |
|---|---|
| `mod` | mod |
| `pct` | pct |

## Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`getpred` *Extract the column names of the regression terms of a linear model*

---

### Description

Extract the column names of the regression terms of a linear model

### Usage

```
get.predictors(mod)
```

### Arguments

| | |
|---|---|
| `mod` | mod |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gevar` *GEV - VaR calculation*

---

### Description

GEV - VaR calculation

### Usage

```
gev.VaR(Xbmax, mu = NULL, xi = NULL, sigma = NULL, prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| `Xbmax` | Xbmax |
| `mu` | mu |
| `xi` | xi |
| `sigma` | sigma |
| `prob` | prob |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gevarci` *GEV - VaR calculation and Confidence Intervals*

---

### Description

GEV - VaR calculation and Confidence Intervals

### Usage

```
gev.VaR.ci(Xbmax, VaR = sum(gev.VaR.constraint(parms = c(0, xi, sigma),
type = "both", Xbmax = Xbmax, prob = prob))/2, xi = 0.1,
sigma = 1, alpha = 0.01, df = 3, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| `Xbmax` | Xbmax |
| `VaR` | VaR |
| `xi` | xi |
| `sigma` | sigma |
| `alpha` | alpha |
| `df` | df |
| `prob` | prob |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gevarcnt — *GEV - VaR Joint Confidence Intervals by Profile Likelihood*

### Description

GEV - VaR Joint Confidence Intervals by Profile Likelihood

### Usage

```
gev.VaR.contour(Xbmax,
VaR = sum(gev.VaR.constraint(parms = c(0, xi, sigma),
type = "both", Xbmax = Xbmax, prob = prob))/2, xi = 0.1,
sigma = 1, alpha = 0.01, df = 3, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| Xbmax | Xbmax |
| VaR | VaR |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gevarcst — *GEV - Domain range for the VaR parameter*

### Description

GEV - Domain range for the VaR parameter

### Usage

```
gev.VaR.constraint(parms, type = c("left", "right", "both"),
Xbmax, prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| type | type |
| Xbmax | Xbmax |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gevarg *GEV - VaR range grid for contour calculation*

---

### Description

GEV - VaR range grid for contour calculation

### Usage

```
gev.VaR.range(Xbmax,
VaR = sum(gev.VaR.constraint(parms = c(0, xi, sigma),
type = "both", Xbmax = Xbmax, prob = prob))/2, xi = 0.1,
sigma = 1, alpha = 0.01, df = 3, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| Xbmax | Xbmax |
| VaR | VaR |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gevark *GEV - VaR Log Likelihood*

### Description

GEV - VaR Log Likelihood

### Usage

```
gev.VaR.like(parms, Xbmax, prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| Xbmax | Xbmax |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gevci *GEV - Distribution fitting and Confidence Intervals*

### Description

GEV - Distribution fitting and Confidence Intervals

### Usage

```
gev.ci(Xbmax, mu = 0, xi = 0.1, sigma = 1, alpha = 0.01, df = 3, ...)
```

### Arguments

| | |
|---|---|
| Xbmax | Xbmax |
| mu | mu |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| ... | Further arguments to or from other methods. |

**Note**

TO BE COMPLETED

**Author(s)**

RAdamant Development Team <team@r-adamant.org>

---

`gevcont` *GEV - Joint Confidence Intervals by Profile Likelihood*

---

**Description**

GEV - Joint Confidence Intervals by Profile Likelihood

**Usage**

```
gev.contour(Xbmax, mu = 0, xi = 0.1, sigma = 1, alpha = 0.01, df = 3, ...)
```

**Arguments**

| | |
|---|---|
| `Xbmax` | Xbmax |
| `mu` | mu |
| `xi` | xi |
| `sigma` | sigma |
| `alpha` | alpha |
| `df` | df |
| `...` | Further arguments to or from other methods. |

**Note**

TO BE COMPLETED

**Author(s)**

RAdamant Development Team <team@r-adamant.org>

## gevlike *GEV - Log Likelihood*

### Description

GEV - Log Likelihood

### Usage

```
gev.like(parms, Xbmax, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| Xbmax | Xbmax |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gevmcst *GEV - Domain range for the mu parameter*

### Description

GEV - Domain range for the mu parameter

### Usage

```
gev.mu.constraint(parms, type = c("left", "right", "both"), Xbmax, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| type | type |
| Xbmax | Xbmax |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

gevml *GEV - Maximum Likelihood Parameters Estimation*

## Description

GEV - Maximum Likelihood Parameters Estimation

## Usage

```
gev.ml(Xbmax, init = c(0, 0.1, 1), ...)
```

## Arguments

| | |
|---|---|
| Xbmax | Xbmax |
| init | init |
| ... | Further arguments to or from other methods. |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

gevrng *GEV - Parameters range grid for contour calculation*

## Description

GEV - Parameters range grid for contour calculation

## Usage

```
gev.range(Xbmax, mu = 0, xi = 0.1, sigma = 1, alpha = 0.01, df = 3, ...)
```

## Arguments

| | |
|---|---|
| Xbmax | Xbmax |
| mu | mu |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gevsicst` *GEV - Domain range for the sigma parameter*

---

### Description

GEV - Domain range for the sigma parameter

### Usage

```
gev.sigma.constraint(parms, type = c("left", "right", "both"), Xbmax, parm.type
"VaR", "ES"), prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| `parms` | parms |
| `type` | type |
| `Xbmax` | Xbmax |
| `parm.type` | parm.type |
| `prob` | prob |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`gevxicst` *GEV - Domain range for the xi parameter*

## Description

GEV - Domain range for the xi parameter

## Usage

```
gev.xi.constraint(parms, type = c("left", "right", "both"),
Xbmax, parm.type = c("mu", "VaR","ES"), prob = 0.01, ...)
```

## Arguments

| | |
|---|---|
| `parms` | parms |
| `type` | type |
| `Xbmax` | Xbmax |
| `parm.type` | parm.type |
| `prob` | prob |
| `...` | ... |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`gini` *Gini index*

## Description

Calculate Gini index based on the results of a classification model.

## Usage

```
Gini(x, ...)
## Default S3 method:
Gini(x, ...)
## S3 method for class 'scorecard'
Gini(x, glob = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class "scorecard" |
| `glob` | global |
| `...` | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gkgamma *GK Gamma*

---

### Description

GK Gamma statistic

### Usage

```
GKgamma(target, pred, ...)
```

### Arguments

| | |
|---|---|
| target | target |
| pred | pred |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gmma *Guppy's Multiple EMA*

---

### Description

Compute Guppy's Multiple EMA on the input data, one for each column of X[, i].

### Usage

```
gmma(X, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function ema. |

### Details

GMMA is two sets (short and long window sizes) of six EMA:
- Short Windows: 3, 5, 8, 10, 12, 15
- Long Windows: 30, 35, 40, 45, 50, 60.

### Value

A object of class 'ma' with attributes type = "GMMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*12 with twelve moving averagesfor each column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

## gpdboot *GPD - parameters bootstrapping*

---

### Description

GPD - parameters bootstrapping

### Usage

```
gpdboot(Xtail, trsh = 0, xi = NULL, sigma = NULL, nboots = 100, ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| xi | xi |
| sigma | sigma |
| nboots | nboots |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdci — *GPD - Distribution fitting and Confidence Intervals*

### Description

GPD - Distribution fitting and Confidence Intervals

### Usage

```
gpd.ci(Xtail, trsh = 0, xi = 0.1, sigma = 1, alpha = 0.01, df = 2, ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdcnt — *GPD - Joint Confidence Intervals by Profile Likelihood*

### Description

GPD - Joint Confidence Intervals by Profile Likelihood

### Usage

```
gpd.contour(Xtail, trsh = 0, xi = 0.1, sigma = 1, alpha = 0.01, df = 2, ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| ... | Further arguments to or from other methods. |

**Note**

TO BE COMPLETED

**Author(s)**

RAdamant Development Team <team@r-adamant.org>

---

`gpdes` *GPD - Expected Shortfall (ES) calculation*

---

**Description**

GPD - Expected Shortfall (ES) calculation

**Usage**

```
gpd.ES(Xtail, trsh = 0, xi = NULL, sigma = NULL, N, prob = 0.01, ...)
```

**Arguments**

| | |
|---|---|
| `Xtail` | Xtail |
| `trsh` | trsh |
| `xi` | xi |
| `sigma` | sigma |
| `N` | N |
| `prob` | prob |
| `...` | ... |

**Note**

TO BE COMPLETED

**Author(s)**

RAdamant Development Team <team@r-adamant.org>

## gpdesci — *GPD - ES calculation and Confidence Intervals*

### Description

GPD - ES calculation and Confidence Intervals

### Usage

```
gpd.ES.ci(Xtail, trsh = 0, ES = trsh + 10^-5, xi = 0.1,
alpha = 0.01, df = 2, N, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| ES | ES |
| xi | xi |
| alpha | alpha |
| df | df |
| N | N |
| prob | prob |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdescnt — *GPD - ES Joint Confidence Intervals by Profile Likelihood*

### Description

GPD - ES Joint Confidence Intervals by Profile Likelihood

### Usage

```
gpd.ES.contour(Xtail, trsh = 0, ES = trsh + 10^-5,
xi = 0.1, alpha = 0.01, df = 2, N, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| ES | ES |
| xi | xi |
| alpha | alpha |
| df | df |
| N | N |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gpdescst *GPD - Domain range for the ES parameter*

---

### Description

GPD - Domain range for the ES parameter

### Usage

```
gpd.ES.constraint(parms, type = c("left", "right", "both"), trsh = 0, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| type | type |
| trsh | trsh |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdesfce — *GPD - Log Likelihood 3D surface as a function of Expected Shortfall*

### Description

GPD - Log Likelihood 3D surface as a function of Expected Shortfall

### Usage

```
gpd.ES.surface(ES = NULL, xi = NULL, Xtail,
trsh = 0, N, prob = 0.01, grid.size = 100, alpha = 0.01, ...)
```

### Arguments

| | |
|---|---|
| ES | ES |
| xi | xi |
| Xtail | Xtail |
| trsh | trsh |
| N | N |
| prob | prob |
| grid.size | grid.size |
| alpha | alpha |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdesk — *GPD - ES Log Likelihood*

### Description

GPD - ES Log Likelihood

### Usage

```
gpd.ES.like(parms, Xtail, trsh = 0, N, prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| Xtail | Xtail |
| trsh | trsh |
| N | N |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gpdesml *GPD - Maximum Likelihood ES Estimation*

---

### Description

GPD - Maximum Likelihood ES Estimation

### Usage

```
gpd.ES.ml(Xtail, trsh = 0, N, init = c(1, 0.1), ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| N | N |
| init | init |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdesrng — *GPD - ES range grid for contour calculation*

### Description

GPD - ES range grid for contour calculation

### Usage

```
gpd.ES.range(Xtail, trsh = 0, ES = trsh + 10^-5,
xi = 0.1, alpha = 0.01, df = 2, N, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| ES | ES |
| xi | xi |
| alpha | alpha |
| df | df |
| N | N |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdlk — *GPD - Log Likelihood*

### Description

GPD - Log Likelihood

### Usage

```
gpd.like(parms, Xtail, trsh = 0, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| Xtail | Xtail |
| trsh | trsh |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gpdml` *GPD - Maximum Likelihood Parameters Estimation*

---

### Description

GPD - Maximum Likelihood Parameters Estimation

### Usage

```
gpd.ml(Xtail, trsh = 0, init = c(0.1, 1), ...)
```

### Arguments

| | |
|---|---|
| `Xtail` | Xtail |
| `trsh` | trsh |
| `init` | init |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gpdrng` *GPD - Parameters range grid for contour calculation*

---

### Description

GPD - Parameters range grid for contour calculation

### Usage

```
gpd.range(Xtail, trsh = 0, xi = 0.1, sigma = 1, alpha = 0.01, df = 2, ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| xi | xi |
| sigma | sigma |
| alpha | alpha |
| df | df |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## gpdsfc *GPD - Log Likelihood 3D surface*

---

### Description

GPD - Log Likelihood 3D surface

### Usage

```
gpd.surface(xi = NULL, sigma = NULL, Xtail,
trsh = 0, grid.size = 100, alpha = 0.01, ...)
```

### Arguments

| | |
|---|---|
| xi | xi |
| sigma | sigma |
| Xtail | Xtail |
| trsh | trsh |
| grid.size | grid.size |
| alpha | alpha |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdsgcnt *GPD - Domain range for the sigma parameter*

### Description

GPD - Domain range for the sigma parameter

### Usage

```
gpd.sigma.constraint(parms, type = c("left", "right", "both"),
Xtail, trsh = 0, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| type | type |
| Xtail | Xtail |
| trsh | trsh |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdvar *GPD - VaR calculation*

### Description

GPD - VaR calculation

### Usage

```
gpd.VaR(Xtail, trsh = 0, xi = NULL, sigma = NULL, N, prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| xi | xi |
| sigma | sigma |
| N | N |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gpdvarci` *GPD - VaR calculation and Confidence Intervals*

---

### Description

GPD - VaR calculation and Confidence Intervals

### Usage

```
gpd.VaR.ci(Xtail, trsh = 0, VaR = trsh + 10^-5,
xi = 0.1, alpha = 0.01, df = 2, N, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| `Xtail` | Xtail |
| `trsh` | trsh |
| `VaR` | VaR |
| `xi` | xi |
| `alpha` | alpha |
| `df` | df |
| `N` | N |
| `prob` | prob |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`gpdvarcn` *GPD - VaR Joint Confidence Intervals by Profile Likelihood*

## Description

GPD - VaR Joint Confidence Intervals by Profile Likelihood

## Usage

```
gpd.VaR.contour(Xtail, trsh = 0, VaR = trsh + 10^-5,
xi = 0.1, alpha = 0.01, df = 2, N, prob = alpha[1], ...)
```

## Arguments

| | |
|---|---|
| `Xtail` | Xtail |
| `trsh` | trsh |
| `VaR` | VaR |
| `xi` | xi |
| `alpha` | alpha |
| `df` | df |
| `N` | N |
| `prob` | prob |
| `...` | Further arguments to or from other methods. |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`gpdvarct` *GPD - Domain range for the VaR parameter*

## Description

GPD - Domain range for the VaR parameter

## Usage

```
gpd.VaR.constraint(parms, type = c("left", "right", "both"), trsh = 0, ...)
```

## Arguments

| | |
|---|---|
| `parms` | parms |
| `type` | type |
| `trsh` | trsh |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gpdvarg` *GPD - VaR range grid for contour calculation*

---

### Description

GPD - VaR range grid for contour calculation

### Usage

```
gpd.VaR.range(Xtail, trsh = 0, VaR = trsh + 10^-5,
xi = 0.1, alpha = 0.01, df = 2, N, prob = alpha[1], ...)
```

### Arguments

| | |
|---|---|
| `Xtail` | Xtail |
| `trsh` | trsh |
| `VaR` | VaR |
| `xi` | xi |
| `alpha` | alpha |
| `df` | df |
| `N` | N |
| `prob` | prob |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdvarlk *GPD - VaR Log Likelihood*

### Description

GPD - VaR Log Likelihood

### Usage

```
gpd.VaR.like(parms, Xtail, trsh = 0, N, prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| Xtail | Xtail |
| trsh | trsh |
| N | N |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdvarml *GPD - Maximum Likelihood VaR Estimation*

### Description

GPD - Maximum Likelihood VaR Estimation

### Usage

```
gpd.VaR.ml(Xtail, trsh = 0, N, init = c(1, 0.1), ...)
```

### Arguments

| | |
|---|---|
| Xtail | Xtail |
| trsh | trsh |
| N | N |
| init | init |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`gpdvarsf` *GPD - Log Likelihood 3D surface as a function of VaR*

---

### Description

GPD - Log Likelihood 3D surface as a function of VaR

### Usage

```
gpd.VaR.surface(VaR = NULL, xi = NULL, Xtail,
trsh = 0, N, prob = 0.01, grid.size = 100, alpha = 0.01, ...)
```

### Arguments

| | |
|---|---|
| `VaR` | VaR |
| `xi` | xi |
| `Xtail` | Xtail |
| `trsh` | trsh |
| `N` | N |
| `prob` | prob |
| `grid.size` | grid.size |
| `alpha` | alpha |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## gpdxicst — *GPD - Domain range for the xi parameter*

### Description

GPD - Domain range for the xi parameter

### Usage

```
gpd.xi.constraint(parms, type = c("left", "right", "both"),
Xtail, trsh = 0, N, parm.type = c("sigma", "VaR", "ES"),
prob = 0.01, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| type | type |
| Xtail | Xtail |
| trsh | trsh |
| N | N |
| parm.type | parm.type |
| prob | prob |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## grad — *Compute numerical gradient of a function*

### Description

Plotting tools

### Usage

```
grad(func = NULL, x, scalar = TRUE, eps = sqrt(.Machine$double.neg.eps), ...)
```

### Arguments

| | |
|---|---|
| func | func |
| x | x |
| scalar | scalar |
| eps | eps |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## grangcas *Granger Causality test*

---

### Description

Perform Granger causality test for parameters of VAR model

### Usage

```
## S3 method for class 'VecAr'
GrangCas(X, cause = NULL, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| cause | cause |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## grautil *RAdamant Graphical utilities*

---

### Description

Graphical utilities used by the plotting functions

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`heas` *Heikin - Ashi techniques*

## Description

Compute Heikin - Ashi techniques (Technical Analysis)

## Usage

```
he_as(Close, Open, High, Low, plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `Open` | VECTOR. Open price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

## Details

A type of candlestick chart that shares many characteristics with standard candlestick charts, but differs because of the values used to create each bar.
The Heikin-Ashi technique is used by technical traders to identify a given trend more easily. Hollow candles with no lower shadows are used to signal a strong uptrend, while filled candles with no higher shadow are used to identify a strong downtrend.

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

---

`hhv` *Highest high*

## Description

Compute Highest high (Technical Analysis)

## Usage

```
hhv(X, lag, na.rm = TRUE)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| na.rm | na.rm |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## hill *Hill function*

---

### Description

Hill function: Approximated gamma parameter of the Generalised Pareto distribution

### Usage

```
Hill(X, trsh)
```

### Arguments

| | |
|---|---|
| X | Input matrix/sequence. Sequences are treated as one column matrices. |
| trsh | vector of probability threshold (interval [0, 1]) |

### Value

A matrix length(trsh) by NCOL(X) of computed quantiles

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

hma *Hull Moving Averages*

## Description

Compute multiple Hull Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

## Usage

```
hma(X, win.size = NROW(X), plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable) |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| plot | LOGICAL. Return plot. |
| ... | Further arguments to or from other methods |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.
HMA is a combination of WMA: WMA(2*WMA(X, win.size/2) - wma(X, win.size), sqrt(win.size)).

## Value

A object of class 'ma' with attributes type = "HMA" and 'win.size' as from the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## See Also

wma

## hroi *Historical Returns on Investments*

### Description

Historical Returns on Investments

### Usage

```
hroi(X, lag = 1, mode = c("auto", "range", "selected"),
autolag.start = 1, range.step = 1, log = TRUE,
VaR.type = "norm", p = 0.05, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| mode | mode |
| autolag.start | autolag.start |
| range.step | range.step |
| log | log |
| VaR.type | VaR.type |
| p | p |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## hvar *Historical Value at Risk*

### Description

Compute historical VaR on each column of the input matrix

### Usage

```
hVaR(X, p = 0.05, centered = FALSE)
```

### Arguments

| | |
|---|---|
| X | Input matrix/sequence. Sequences are treated as one column matrices. |
| p | vector of probabilities (DEFAULT = 0.05) |
| centered | LOGICAL. If TRUE, input data are standardised |

### Value

A matrix length(p) by NCOL(X) of computed quantiles

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## ichkh *Ichimoku Kinko Hyo*

---

### Description

Compute Ichimoku Kinko Hyo (Technical Analysis)

### Usage

```
Ichkh(Close, High, Low, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | close |
| High | high |
| Low | low |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## impulse *Unitary impulse*

### Description

Generates an impulse sequence of specified length

### Usage

```
impulse(N, value = 1)
```

### Arguments

| | |
|---|---|
| N | Length of the impulse |
| value | value of the impulse (Default = 1) |

### Value

Impulse sequence of specified length

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## in2woe *Data to Weight of Evidence*

### Description

Transform input data according to weight of evidence

### Usage

```
input2woe(data, nseg, woe, na.replace,
seg.type = c("freq_equal", "width_equal"))
```

### Arguments

| | |
|---|---|
| data | data |
| nseg | nseg |
| woe | woe |
| na.replace | na.replace |
| seg.type | seg.type |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## inertia *Inertia oscillator*

### Description

Compute Inertia oscillator (Technical Analysis)

### Usage

```
Inertia(X, lag, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## invlogit *Inverse Logit transformation*

### Description

Inverse Logit transformation

### Usage

```
inv.logit(y)
```

### Arguments

| | |
|---|---|
| y | y |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## invp *Peizer-Pratt Inversion formula*

### Description

Peizer-Pratt Inversion formula

### Usage

```
InvPP(z, n)
```

### Arguments

| | |
|---|---|
| z | z |
| n | n |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## irsvecar *VAR Impulse response*

### Description

Compute and plot Impulse response function calculated for VAR model

### Usage

```
IRS.VecAr(X, imp, resp = NULL, steps = 5, cum = TRUE, ortho = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| imp | imp |
| resp | resp |
| steps | steps |
| cum | cum |
| ortho | ortho |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`isfs` *Check for inheritance from Financial Series class*

---

### Description

Check for inheritance from Financial Series class

### Usage

```
is.fs(X)
```

### Arguments

| | |
|---|---|
| `X` | X |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`jbtest` *Jaques-Brera normality test*

---

### Description

Compute Jaques-Brera normality test for each column of X

### Usage

```
JB.test(X, plot.hist=FALSE)
```

### Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable) |
| `plot.hist` | LOGICAL. Return histogram. |

### Value

Matrix of Jaques-Brera scores and P-Value

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

`kurt`, `skew`

## jensen *Jensen index*

### Description

`Jensen`: Calculate Jensen index for a portfolio
`Jensen.Capm`: Get Jensen index from an object of class "Capm".

### Usage

```
Jensen(PTF, ...)
## Default S3 method:
Jensen(PTF, PTF_M, rf = NULL, rfr = 0, ...)
## S3 method for class 'Capm'
Jensen(PTF, rfr = 0, ...)
```

### Arguments

| | |
|---|---|
| `PTF` | Input portfolio or an object of class "Capm" |
| `PTF_M` | Market/benchmark portfolio |
| `rfr` | risk free rate |
| `rf` | risk free asset |
| `...` | Further arguments to or from other methods |

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

### See Also

`Sharpe`, `Treynor`, `Appraisal`

## jrbtree *JR Binomial Tree*

### Description

Option evaluation with Jarrow and Rudd Binomial Tree

### Usage

```
JR.BinTree(Nsteps, p, under, strike, rfr, sigma,
maty, yield, life, ret.steps = FALSE)
```

### Arguments

| | |
|---|---|
| Nsteps | Nsteps |
| p | p |
| under | under |
| strike | strike |
| rfr | rfr |
| sigma | sigma |
| maty | maty |
| yield | yield |
| life | life |
| ret.steps | ret.steps |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## kama *Kauffman Adaptive Moving Average*

---

### Description

Kauffman Adaptive Moving Average, computed on each column of the input data X and for each pair (fast.win[i], slow.win[i]).

### Usage

```
kama(X, fast.win = 2, slow.win = 30, lag = 5,
keep.lambda = FALSE, keep.ER = FALSE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| fast.win | vector of fast window sizes (fast lags) (DEFAULT = 2) |
| slow.win | vector of slow window sizes (slow lags) (DEFAULT = 30) |
| lag | vector of lags used to compute Kauffman efficiency ratio (DEFAULT = 5). Recycled to be of equal length as fast and slow lags if necessary |
| keep.lambda | LOGICAL. If TRUE, adaptive smoothing factor lambda is returned as an attribute (DEFAULT = FALSE) |
| keep.ER | LOGICAL. If TRUE, adaptive Efficiency Ratio ER is returned as an attribute (DEFAULT = FALSE) |
| plot | LOGICAL. Return plot. |
| ... | Further arguments to or from other methods. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

An object of class 'Movav' with attributes type = "KAMA", 'lambda' and 'ER' as required and 'fast.win', 'slow.win' and 'lag' given by the corresponding input parameters:
- matrix of size NROW(X) by NCOL(X)*length(fast.win) where each column is the moving average of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ama

---

## kelt *Keltner channel*

---

### Description

Compute Keltner channel (Technical Analysis)

### Usage

```
kelt(Close, High, Low, mult = 2, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| mult | mult |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`kendalt` *KendallTau*

## Description

Calculate KendallTau statistic

## Usage

```
KendallTau(target, pred, ...)
```

## Arguments

| | |
|---|---|
| `target` | target |
| `pred` | pred |
| `...` | Further arguments to or from other methods. |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

`kri` *Kairi Relative Index*

## Description

Compute Kairi Relative Index (Technical Analysis)

## Usage

```
kri(X, lag1 = 10, lag2 = 20, plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `X` | X |
| `lag1` | lag1 |
| `lag2` | lag2 |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## kurtskew *Kurtosis and Skewness*

### Description

`kurt`: Compute the excess kurtosis for each column of X
`skew`: Compute the skewness for each column of X

### Usage

```
kurt(X, pval = FALSE)
skew(X, pval = FALSE)
```

### Arguments

| | |
|---|---|
| `X` | Matrix of numeric data series (one column per variable). |
| `pval` | LOGICAL. Return P-Value. |

### Value

Matrix of Excess Kurtosis / Skewness and P-Value

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

### See Also

`JB.test`

## kvo *Klinger oscillator*

### Description

Compute Klinger oscillator (Technical Analysis)

### Usage

```
kvo(Close, High = NULL, Low = NULL,
Vol = NULL, cumulative = FALSE, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Vol` | VECTOR. Asset traded Volume. |
| `cumulative` | cumulative |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Details

The Klinger Oscillator is a volume- and price-based oscillator intended to measure both short- and long-term money flows into and out of a security.
The Klinger Volume Oscillator measures trends of money flows based on volume.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## lagret *Time Series Operators*

---

### Description

`Ret`: Compute N-points Returns on each column of the input matrix.
`Lag`: Compute lag on each column of the input matrix.
`Diff`: Compute lagged difference on each column of the input matrix.
`MDiff`: Compute Multiple lagged differences on each column of the input matrix. \ cr `MLag`: Compute Multiple lags on each column of the input matrix

### Usage

```
Ret(X, lag = 1, log = FALSE, na.rm = FALSE, plot = FALSE, ...)

Lag(X, lag = 1, na.rm = FALSE, padding = NA)

Diff(X, lag = 1, padding = NA, na.rm = FALSE)

MDiff(X, lag = 1, padding = NA,
mode = c("auto", "range", "selected"), na.rm = FALSE)

MLag(X, lag = 1, na.rm = FALSE, padding = NA,
mode = c("auto", "range", "selected"), autolag.start = 1)
```

### Arguments

| | |
|---|---|
| `X` | Input data (i.e. matrix/vector of prices) |
| `lag` | number of lags |
| `log` | BOOLEAN: compute log-returns |
| `na.rm` | BOOLEAN: remove NAs |
| `plot` | BOOLEAN: return plot |
| `padding` | value to replace removed observations |
| `mode` | mode |
| `autolag.start` | autolag.start |
| `...` | Further arguments to or from other methods |

### Details

Sequences are treated as one-column matrices.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## lew *Moving window*

---

### Description

Apply a given function to an extending window of the lagged data series of the input matrix, each column separately.

### Usage

```
lew(X, lag = 0, padding = NA, na.rm = FALSE,
func = NULL, is.cumulative = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | Input matrix/sequence |
| lag | vector of integer lags. If lag >= 0 data are shifted to the right, else to the left. (DEFAULT = 0) |
| padding | value used to initialise the output matrix (DEFAULT = NA) |
| na.rm | LOGICAL. If TRUE, N-lag entries are removed from the output (DEFAULT = FALSE) |
| func | function applied to the extending data window (DEFAULT = NULL) |
| is.cumulative | LOGICAL. If TRUE it the function provided must be cumulative by itself (like cummax, cummin, etc..) (DEFAULT = TRUE) |
| ... | Additional parameters accepted by the function 'func' |

### Details

Sequences are treated as one-column matrices

### Value

A matrix where func has been applied on increasing data windows for each column of X. Number of rows depends on the na.rm parameter. Number of columns is NCOL(X)

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

cumSum, cumMin, cumMax, cumSd, cumVar

## liftgain *Lift and Gain plot*

### Description

Plot cumulative Gain and Lift chart for a classification model

### Usage

```
Gain(x, ...)
Lift(x, ...)
## S3 method for class 'scorecard'
Gain(x, pc = 0.1, ...)
## S3 method for class 'scorecard'
Lift(x, pc = 0.1, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| pc | pc |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## ljbgarch *Ljung-Box test*

### Description

Perform Ljung-Box test for residual correlation

### Usage

```
LjungBox(x, lags, plot.acf = FALSE)
```

### Arguments

| | |
|---|---|
| x | Residual series or object of class "Garch" |
| lags | Number of lags to calculate the autocorrelation function |
| plot.acf | LOGICAL. Plot ACF. |

## lkegarch *EGARCH likelihood function*

### Description

Calculate EGARCH likelihood function

### Usage

```
like.egarch(theta, ee, x, Y,
order = c(alpha = 1, beta = 1), prob = c("norm", "ged", "t"))
```

### Arguments

| | |
|---|---|
| theta | theta |
| ee | ee |
| x | x |
| Y | Y |
| order | order |
| prob | prob |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## lkgarch *GARCH likelihood function*

### Description

Calculate GARCH likelihood function

### Usage

```
like.garch(theta, ee, x, Y, order, prob = c("norm", "ged", "t"), r)
```

### Arguments

| | |
|---|---|
| theta | theta |
| ee | ee |
| x | x |
| Y | Y |
| order | order |
| prob | prob |
| r | r |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## lkmgarch *MGARCH likelihood function*

---

### Description

Calculate MGARCH likelihood function

### Usage

```
like.mgarch(theta, x, Y, order, prob=c("norm","ged","t"))
```

### Arguments

| | |
|---|---|
| theta | theta |
| x | x |
| Y | Y |
| order | order |
| prob | prob |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## lktgarch *TGARCH likelihood function*

---

### Description

Calculate TGARCH likelihood function

### Usage

```
like.tgarch(theta, ee, x, Y, order, prob = c("norm","ged", "t"))
```

### Arguments

| | |
|---|---|
| `theta` | theta |
| `ee` | ee |
| `x` | x |
| `Y` | Y |
| `order` | order |
| `prob` | prob |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

---

## `llv` *Lowest low*

---

### Description

Compute Lowest low (Technical Analysis)

### Usage

```
llv(X, lag, na.rm = TRUE)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `lag` | INTEGER. Number of lag periods. |
| `na.rm` | na.rm |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## logger *Main logging function*

### Description

Create Log for the functions contained in the package

### Usage

```
Logger(message = "", from = deparse(sys.call(sys.parent())),
level = 1, line = NA, env = getOption("RAdamant"),
console = .getConsoleLogging(env = env),
logfile = .getLogFile(env = env))
```

### Arguments

| | |
|---|---|
| message | Message printed. |
| from | from |
| level | Log depth level, minimum = 1 |
| line | line |
| env | environment |
| console | console logging |
| logfile | log file |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## logit *Logit transformation*

### Description

Logit transformation

### Usage

```
logit(x, adjust = 5e-05)
```

### Arguments

| | |
|---|---|
| x | x |
| adjust | adjust |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## lrbtree *LR Binomial Tree*

---

### Description

Option evaluation with Leinsen and Reimer Binomial Tree

### Usage

```
LR.BinTree(Nsteps, under, strike, rfr,
sigma, maty, yield, life, ret.steps = FALSE)
```

### Arguments

| | |
|---|---|
| Nsteps | Nsteps |
| under | under |
| strike | strike |
| rfr | rfr |
| sigma | sigma |
| maty | maty |
| yield | yield |
| life | life |
| ret.steps | ret.steps |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## macd *Moving Average Convergence / Divergence*

### Description

Compute Moving Average Convergence / Divergence (Technical Analysis)

### Usage

```
macd(X, fast.lag = 12, slow.lag = 26, signal.lag = 14, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| fast.lag | fast.lag |
| slow.lag | slow.lag |
| signal.lag | signal.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mass *Mass indicator*

### Description

Compute Mass indicator (Technical Analysis)

### Usage

```
mass(High, Low, Close = NULL, lag = 9, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Close | VECTOR. Close price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## masscum *Mass indicator cumulative*

---

### Description

Compute Mass indicator cumulative (Technical Analysis)

### Usage

```
mass.cum(High, Low, Close = NULL, lag = 9, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Close | VECTOR. Close price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mcf *Auto-Correlation and Partial Auto-Correlation*

---

### Description

Compute auto-correlation and partial auto-correlation function on a matrix

### Usage

```
mcf(X, lag.max = 10, ci = 0.95, plot=TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable) |
| lag.max | Max lag to be computed by the cross correlation function (DEFAULT: 10) |
| ci | Confidence Interval (DEFAULT: 0.95) |
| plot | LOGICAL. If TRUE, results are plotted. |
| ... | additional parameters accepted by the function plot.cross.ccf. |

### Value

A list with two entries:
- ACF: list of Auto-Correlation Functions (one for each column of X)
- PACF: list of Partil Auto-Correlation Functions (one for each column of X)

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

cross.ccf

---

## mcgind *McGinley Dynamic Indicator*

---

### Description

Compute McGinley Dynamic Indicator (Technical Analysis)

### Usage

```
mcgind(X, lag = 12, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mcosc *McClellan Oscillator*

### Description

Compute McClellan Oscillator (Technical Analysis)

### Usage

```
mcosc(X, fast.lag = 19, slow.lag = 39, hist.lag = 9, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| fast.lag | fast.lag |
| slow.lag | slow.lag |
| hist.lag | hist.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mcplot *Muliple correlation plot*

### Description

Multiple correlation plot

### Usage

```
mcplot(X, hist.nclass = 10, theme.params = getCurrentTheme(),
coLin = TRUE,
main = ifelse(coLin, "Co-Linearity Analysis", "Multi-Correlation Analysis"),
new.device = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| hist.nclass | hist.nclass |
| theme.params | theme.params |
| coLin | coLin |
| main | main |
| new.device | new.device |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mcsi *McClellan Summation Index*

---

### Description

Compute McClellan Summation Index (Technical Analysis)

### Usage

```
mcsi(matr, nr, nc, lag1, lag2, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| matr | matr |
| nr | nr |
| nc | nc |
| lag1 | lag1 |
| lag2 | lag2 |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## means — *Geometric and Harmonic means*

### Description

`gmean`: Compute the geometric mean for each column of X
`hmean`: Compute the harmonic mean for each column of X

### Usage

```
gmean(X, ...)
hmean(X, ...)
```

### Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable) |
| `...` | Additional parameters accepted by the function sum (i.e. na.rm) |

### Value

Matrix of harmonic / geometric means

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mfind — *Money flow indicator*

### Description

Compute Money flow indicator (Technical Analysis)

### Usage

```
Mflow.ind(Close, High, Low, Volume, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Volume` | VECTOR. Asset traded Volume. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`mflow` *Money flow*

---

### Description

Compute Money flow (Technical Analysis)

### Usage

```
Mflow(Close, High, Low, Volume, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Volume` | VECTOR. Asset traded Volume. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`mfratio` *Money flow ratio*

---

### Description

Compute Money flow ratio (Technical Analysis)

### Usage

```
Mflow.ratio(Close, High, Low, Volume, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `Volume` | VECTOR. Asset traded Volume. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## minmaxs *Mini/Max Scale*

---

### Description

Compute minimum / maximum scale of a vector

### Usage

```
Minmaxscal(x, tmin = 0, tmax = 1)
```

### Arguments

| | |
|---|---|
| x | x |
| tmin | tmin |
| tmax | tmax |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mma *Modified EMA*

---

### Description

Compute multiple Modified EMA on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
mma(X, win.size = NROW(X), plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function ema. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
MMA is a EMA with smoothing factor: lambda = 1/win.size.

### Value

A object of class 'ma' with attributes type = "MMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

## mndma *Modified N-Day Moving Averages*

---

### Description

Computes multiple Modified N-Day Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
mndma(X, win.size = 50, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable) |
| win.size | Vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by the function sma |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A object of class 'Movav' with attributes type = "MNDMA" and 'win.size' as from the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

sma

---

## mom *Momentum oscillator*

---

### Description

Compute Momentum oscillator (Technical Analysis)

### Usage

```
mom(X, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## moments *Main Moments*

---

### Description

Calculate sample moments on each columns of X

### Usage

```
moments(X)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable) |

### Value

Matrix of moments

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

JB.test , skew , kurt

---

## movapply *Moving Apply function*

---

### Description

Applies a given function to a sliding window of the input data

### Usage

```
movApply(X, win.size = 1, padding = NA, rm.transient = FALSE, func = NULL, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of data window sizes that will be passed to the given function "func" (DEFAULT = 1). |
| padding | Padding value to fill transient of result (output data rows from 1 to win.size-1). (DEFAULT = NA) |
| rm.transient | transient: LOGICAL. If TRUE, transient is removed, otherwise func is applied to the transient. (DEFAULT = FALSE) |
| func | Function to be run |
| ... | Additional parameters accepted by the function func |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A matrix of size NROW(X) by NCOL(X)*length(win.size). func is applied to each sliding window SWi (given by win.size[i]) and each column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`movav` *Generic 8Multiple) Moving Average*

## Description

Generic Multiple Moving Average (MA filter). Compute multiple FIR filtering on each column of the input data

## Usage

```
Movav(X, ...)
## Default S3 method:
Movav(X, win.size = NULL,
func = NULL, type = "MA", desc = "Moving Average",
plot= FALSE, ...)
```

## Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable). |
| `win.size` | vector of lengths of the FIR filters to be applied on the data X. (DEFAULT = NULL). |
| `func` | function accepting an integer N and returning an N-long set of filter coefficients. |
| `type` | Charachter attribute attached to the result (DEFAULT: "MA"). |
| `desc` | desc |
| `plot` | LOGICAL. Return plot. |
| `...` | Further arguments to or from other methods |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.

## Value

A object of class 'Movav' with attributes 'type' and 'win.size' as given by the corresponding input parameters:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## movfunc *Moving Base Functions*

### Description

Applies the function "Max", "Min", "Standard Deviation" or "Variance" to a sliding window of the input data

### Usage

```
movMax(X, win.size = 1, ...)
movMin(X, win.size = 1, ...)
movSd(X, win.size = 1, ...)
movVar(X, win.size = 1, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of data window sizes that will be used for the calculations (DEFAULT = 1). |
| ... | Additional parameters accepted by the function movApply |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A matrix of size NROW(X) by NCOL(X)*length(win.size). max is applied to each sliding window SWi (given by win.size[i]) and

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

movApply

## mqt *Multiple t quantile*

### Description

Compute quantiles from Students T distribution for multiple degrees of freedom values

### Usage

```
mqt(p, df, ...)
```

### Arguments

| | |
|---|---|
| p | Vector of probabilities (DEFAULT = 0.05) |
| df | Vector of degrees of freedom |
| ... | Further arguments to and from other methods |

### Value

A matrix length(p) by length(df) of computed quantiles

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mreg *Multiple regressions*

---

### Description

Multiple regressions

### Usage

```
mreg(Y, X, xlabels = NULL, tick.step = 1, backtest = 0,
stress.idx = c(), type = "simple",
model = "lm", ci = 0.95, max.vars = NCOL(X),
intercept = TRUE, family = gaussian, weights = NULL,
plot = TRUE, scope = NULL, trace = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Y | Y |
| X | X |
| xlabels | xlabels |
| tick.step | tisck.step |
| backtest | backtest |
| stress.idx | stress.idx |
| type | type |
| model | model |
| ci | ci |
| max.vars | max.vars |
| intercept | intercepts |
| family | family |
| weights | weights |
| plot | LOGICAL. If TRUE plot is returned. |
| trace | trace |
| scope | scope |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mse *Mean squared error*

---

### Description

Calculate Mean squared error for model evaluation

### Usage

```
mse(actual, predicted)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mstyles *Multiple Style analysis (Portfolio)*

---

### Description

Perform Style analysis for multiple time periods

### Usage

```
Multi.Styles(FUND, IND, W, n_clust = 5, lower = NULL, upper = NULL, ...)
```

### Arguments

| | |
|---|---|
| FUND | FUND |
| IND | IND |
| W | W |
| n_clust | n_clust |
| lower | lower |
| upper | upper |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mtacf *Cool.Acf methods*

---

### Description

Plot and Print methods for class 'cool.acf'

### Usage

```
## S3 method for class 'cool.acf'
print(x, ...)
## S3 method for class 'cool.acf'
plot(x, theme.params = getCurrentTheme(), xtitle = "Lag", ytitle =
expression(rho), overrides = list(...), ...)
```

### Arguments

| | |
|---|---|
| x | Instance of class 'cool.acf' |
| theme.params | Theme parameters (DEFAULT: getCurrentTheme()) |
| xtitle | Title for the x-axis (DEFAULT: "Lag") |
| ytitle | Title for the y-axis (DEFAULT: expression(rho)) |
| overrides | List of parameters to override the theme. Must match by name the parameters defined by the theme (DEFAULT: NULL) |
| ... | Further arguments to or fromother methods |

### Value

Void

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mtccf *Cross.ccf functions*

### Description

Methods for class 'cross.ccf'

### Usage

```
## S3 method for class 'cross.ccf'
print(x, ...)
## S3 method for class 'cross.ccf'
plot(x, theme.params = getCurrentTheme(), xtitle = "Lag", ytitle =
expression(rho), overrides = list(...), ...)
```

### Arguments

| | |
|---|---|
| x | Instance of class 'cross.ccf' |
| theme.params | Theme parameters (DEFAULT: getCurrentTheme()) |
| xtitle | Title for the x-axis (DEFAULT: "Lag") |
| ytitle | Title for the y-axis (DEFAULT: expression(rho)) |
| overrides | List of parameters to override the theme. Must match by name the parameters defined by the theme (DEFAULT: NULL) |
| ... | Further arguments to or fromother methods |

### Value

Void

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mtfft *Plot and Print FFT*

### Description

Plot and Print methods for class 'FFT'

### Usage

```
## S3 method for class 'FFT'
print(x, ...)

## S3 method for class 'FFT'
plot(x, theme.params = getCurrentTheme(),
overrides = list(...), shaded = TRUE,
show.periodicity = FALSE, show.legend = FALSE,
zoom = 100, semilog = FALSE, new.device = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| theme.params | theme.params |
| overrides | overrides |
| shaded | shaded |
| show.periodicity | show.periodicity |
| show.legend | show.legend |
| zoom | zoom |
| semilog | semilog |
| new.device | new.device |
| ... | Further arguments to or fromother methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mtfs *Methods for Financial Series class*

---

### Description

Plot and Print methods for Financial Series class

### Usage

```
## S3 method for class 'fs'
plot(x, ...)
## S3 method for class 'fs'
plot(x, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mtmcf *Mcf methods*

### Description

Plot and Print method for class 'mcf'

### Usage

```
## S3 method for class 'mcf'
print(x, ...)

## S3 method for class 'mcf'
plot(x, theme.params = getCurrentTheme(),
xtitle = "Lag", ytitle = expression(rho),
overrides = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | Instance of class 'mcf' |
| theme.params | Theme parameters (DEFAULT: getCurrentTheme()) |
| xtitle | Title for the x-axis (DEFAULT: "Lag") |
| ytitle | Title for the y-axis (DEFAULT: expression(rho)) |
| overrides | List of parameters to override the theme. Must match by name the parameters defined by the theme (DEFAULT: NULL) |
| ... | Further arguments to or fromother methods |

### Value

Void

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## mtoscil *Plot function for Oscillators*

### Description

Plot and Print method for Oscillators (Technical Analysis)

### Usage

```
## S3 method for class 'oscil'
print(x, digits = 5, ...)

## S3 method for class 'oscil'
plot(x, Y = NULL, main = "",
show.trsh = NULL, xlabels = rownames(Y),
theme.params =getTheme(1), overrides = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| Y | Y |
| main | main |
| show.trsh | show treshold |
| xlabels | xlabels |
| theme.params | them.params |
| overrides | overrides |
| digits | digits |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mtreg *Methods for reg*

---

### Description

Plot, Print ND Summary method for "reg"

### Usage

```
## S3 method for class 'reg'
print(x, ...)
## S3 method for class 'reg'
summary(object, ...)
## S3 method for class 'reg'
plot(x, mode = c("response", "link"),
title = ifelse(x$model.type == "lm", "LS Regression","GLM Regression"),
theme.params = getCurrentTheme(),
overrides = list(...), ...)
```

### Arguments

| | |
|---|---|
| x, object | x |
| mode | mode |
| title | title |
| theme.params | theme.params |
| overrides | overrides |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## mtunivar *Methods for univariate analysis*

---

### Description

Print, Plot and Summary methods for class 'univar'

### Usage

```
## S3 method for class 'univar'
summary(object, ...)
## S3 method for class 'univar'
plot(x, theme.params = getCurrentTheme(), overrides = NULL, ...)
## S3 method for class 'univar'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x, object | Instance of class 'univar' |
| theme.params | params: Theme parameters (DEFAULT: getCurrentTheme()) |
| overrides | list of parameters to override the theme. Must match by name the parameters defined by the theme (DEFAULT: NULL) |
| ... | Further arguments to or fromother methods |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

univar

namutil *Get column and row names*

## Description

Retrieve column / row names from a matrix.

## Usage

```
get.col.names(X, default = "X")
get.row.names(X, default = "")
```

## Arguments

| | |
|---|---|
| X | Input matrix. |
| default | LOGICAL vector. Each entry determines the sort direction of the respective column of X. Recycled if necessary. (DEFAULT = FALSE). |

## Details

Sequences are treated as one column matrices.
Default names are given if input has missing names.

## Value

A character sequence containing the column names of X, or a default set of names if X has no column names

## Author(s)

RAdamant Development Team <team@r-adamant.org>

---

newsimp *News impact curve*

## Description

Compute News impact curve

## Usage

```
newsimp(x, ...)
## S3 method for class 'Garch'
newsimp(x, plot = TRUE, ...)
## Default S3 method:
newsimp(x, theta, order,
type=c("garch","mgarch", "egarch","tgarch"),
plot=FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| theta | theta |
| order | order |
| type | type |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## normfit *Fit normal distribution*

---

### Description

Fit normal distribution

### Usage

```
norm.fit(x, n = 200, range = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| n | n |
| range | range |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## normlike *Normal Distribution - Log Likelihood function*

### Description

Normal Distribution - Log Likelihood function

### Usage

```
norm.like(parms, X, ...)
```

### Arguments

| | |
|---|---|
| parms | parms |
| X | X |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## objgarch *Garch objects*

### Description

Extract objects from Garch model (class "Garch")

### Usage

```
## S3 method for class 'Garch'
coef(object, names=TRUE, ...)
## S3 method for class 'Garch'
logLik(object, ...)
## S3 method for class 'Garch'
vcov(object, ...)
```

### Arguments

| | |
|---|---|
| object | An object of class "Garch" |
| names | Return names |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`obv` *On Balance Volume oscillator*

---

### Description

Compute On Balance Volume oscillator (Technical Analysis)

### Usage

```
Obv(Close, Volume)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `Volume` | VECTOR. Asset traded Volume. |

### Details

The On Balance Volume (OBV) is a cumulative total of the up and down volume. When the close is higher than the previous close, the volume is added to the running total, and when the close is lower than the previous close, the volume is subtracted from the running total.
To interpret the OBV, look for the OBV to move with the price or precede price moves. If the price moves before the OBV, then it is a non-confirmed move. A series of rising peaks, or falling troughs, in the OBV indicates a strong trend. If the OBV is flat, then the market is not trending.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`oscil` *Oscillator default method*

---

### Description

Compute Oscillator (Technical Analysis)

### Usage

```
oscil(X, ...)
## Default S3 method:
oscil(X, Y, pc = FALSE, type = "oscil", ...)
```

### Arguments

| | |
|---|---|
| X | X |
| Y | Y |
| pc | pc |
| type | type |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## pchan *Price channel*

---

### Description

Compute Price channel (Technical Analysis)

### Usage

```
Pchan(CLose, High, Low, lag = 20, na.rm = TRUE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| CLose | CLose |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |
| na.rm | na.rm |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## pdfhit *Density of Hitting probability*

### Description

Density for the First Hitting time

### Usage

```
PDFHit(t, B = 0, S0 = 0, mi, sigma, cumul = FALSE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| t | t |
| B | B |
| S0 | S0 |
| mi | mi |
| sigma | sigma |
| cumul | cumul |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## perf *Performance indicator*

### Description

Compute Performance indicator (Technical Analysis)

### Usage

```
Perf(X, ini.per = 1, cut = TRUE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| ini.per | ini.per |
| cut | cut |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Performance indicator displays the percentage difference between the price today and the price at the start of the data series. It is also known as a normalized price. It can be useful for comparing the performance of two securities or a security and an index.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## pfactor *Print "factorise"*

---

### Description

Print method for "factorise"

### Usage

```
## S3 method for class 'Factorise'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "Factorise". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## pfe *Polarized fractal efficiency*

### Description

Compute Polarized fractal efficiency (Technical Analysis)

### Usage

```
pfe(X, lag = 9, corr_fact = 200, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| corr_fact | corr_fact |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## pgarch *Print Garch*

### Description

Print function for Garch model

### Usage

```
## S3 method for class 'Garch'
print(x, digits = 5, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| digits | digits |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`pgev` *Generalised Extreme Value (GEV) - Probability function*

---

### Description

Generalised Extreme Value (GEV) - Probability function

### Usage

```
pgev(X, mu = 0, xi = 0.1, sigma = 1)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `mu` | mu |
| `xi` | xi |
| `sigma` | sigma |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`pgpd` *Generalised Pareto Distribution (GPD) - Probability function*

---

### Description

Generalised Pareto Distribution (GPD) - Probability function

### Usage

```
pgpd(Q, xi = 0.1, sigma = 1, trsh = 0)
```

### Arguments

| | |
|---|---|
| `Q` | Q |
| `xi` | xi |
| `sigma` | sigma |
| `trsh` | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`pgrangas` *Print Granger test*

---

### Description

Print function for Granger test

### Usage

```
## S3 method for class 'GrangCas'
print(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | OBJECT of class "GrangCas". |
| `...` | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`phivecar` *VAR - PHI*

---

### Description

Estimate PHI matrix for MA (Wold) representation of VAR model

### Usage

```
PHI.VecAr(X, steps, ortho = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | OBJECT of class "VecAR" |
| `steps` | INTEGER. Number of steps ahead. |
| `ortho` | LOGICAL. If TRUE matrix is orthogonal |
| `...` | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`plikeci` *Likelihood confidence intervals calculation*

---

### Description

General function for profile likelihood confidence intervals calculation

### Usage

```
plike.ci(ML.init = c(), flike = NULL, alpha = 0.01, df = NULL, frange = list(),
NULL, ...)
```

### Arguments

| | |
|---|---|
| `ML.init` | ML.init |
| `flike` | flike |
| `alpha` | alpha |
| `df` | df |
| `frange` | frange |
| `par.names` | par.names |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## plikecnt *Likelihood joint confidence intervals contour*

### Description

General function for profile likelihood joint confidence intervals contour

### Usage

```
plike.contour(ML.init = c(), flike = NULL,
alpha = 0.01, df = NULL, frange = list(),
par.names = NULL, grid.size = 100, ...)
```

### Arguments

| | |
|---|---|
| ML.init | ML.init |
| flike | flike |
| alpha | alpha |
| df | df |
| frange | frange |
| par.names | par.names |
| grid.size | grid.size |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## plikerng *Range grid for contour calculation*

### Description

General range grid for contour calculation

### Usage

```
plike.range(ML.init = c(), flike = NULL,
alpha = 0.01, df = NULL, frange = list(), par.names
= NULL, grid.size = 100, max.iter = 100, tol = 10^-5, ...)
```

### Arguments

| | |
|---|---|
| ML.init | ML.init |
| flike | flike |
| alpha | alpha |
| df | df |
| frange | frange |
| par.names | par.names |
| grid.size | grid.size |
| max.iter | max.iter |
| tol | tol |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## plotkit *Plotting Tools*

---

### Description

Utilities functions used for Plotting

### Usage

```
draw.grid(X, base = NULL, theme.params = getCurrentTheme())

draw.legend(legend = "", theme.params = getCurrentTheme(),
overrides = list(...), ...)

draw.projections(X, Y, Y.fit,
col = getCurrentTheme()[["projection.col"]][1],
type = getCurrentTheme()[["projection.type"]][1],
lty = getCurrentTheme()[["projection.lty"]][1])

draw.x.axis(X, base = NULL, xlabels = NULL,
theme.params = getCurrentTheme(), show.labels = TRUE)

draw.x.title(xtitle = "", theme.params = getCurrentTheme())

draw.y.axis(X, ylabels = NULL, theme.params = getCurrentTheme(),
side = 1, show.labels = TRUE)

draw.y.title(ytitle = "", theme.params = getCurrentTheme(), side = 1)
```

### Arguments

| | |
|---|---|
| X | X |
| Y | Y |
| base | base |
| theme.params | theme.params |
| overrides | overrides |
| legend | legend |
| xlabels | xlabels |
| ylabels | ylabels |
| xtitle | xtitle |
| ytitle | ytitle |
| show.labels | show.labels |
| Y.fit | Y.fit |
| col | col |
| type | type |
| lty | lty |
| side | side |
| ... | Further arguments to or from other methods. |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## plotmov *Plot Moving Average*

---

### Description

Plot method for object of class 'Movav' (Moving Average)

### Usage

```
## S3 method for class 'Movav'
plot(x, fs = NULL, main = attr(x, "desc"), ...)
```

### Arguments

| | |
|---|---|
| x | instance of class 'Movav' |
| fs | Matrix containing the original data series (one column per variable). For financial time series (class = 'fs'), only 'Close' column is processed. |
| main | Main title of the plot |
| ... | Additional parameters accepted by the functions cplot and fin.plot |

### Details

If the original data series is an instance of class 'fs', then the plot will have two panels:
- plot of fs and x on the top;
- histogram of the Volume data of the financial series X.

### Value

VOID

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

cplot

### Examples

```
# Compute Exponentioal Moving Average and plot results
x = ema(rnorm(100), 10)
plot(x)
```

---

## plotmreg *Plot function for mreg*

---

### Description

Plot function for class 'mreg'

### Usage

```
## S3 method for class 'mreg'
plot(x, ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "mreg". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## plotret *Plot Returns*

### Description

Plot function for Returns

### Usage

```
## S3 method for class 'ret'
plot(x, style = c("line", "bar"), xlabels = rownames(x), theme.params =
getCurrentTheme(), ...)
```

### Arguments

| | |
|---|---|
| x | x |
| style | style |
| xlabels | xlabels |
| theme.params | theme.params |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## plotroi *Plot Return on Investment objects*

### Description

Plotting function for Return on Investment objects

### Usage

```
## S3 method for class 'roi'
plot(x, main = "Historical Return on Investment", xtitle = "Lag", ...)
```

### Arguments

| | |
|---|---|
| x | x |
| main | main |
| xtitle | xtitle |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## plotsme *Plot Sample Mean Excess class*

---

### Description

Plotting function for Sample Mean Excess class

### Usage

```
## S3 method for class 'sme'
plot(x, main = attr(x, "desc"), xtitle = get.col.names(attr(x, "data")), ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "sme". |
| main | main |
| xtitle | xtitle |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## pmreg *Print function for mreg*

---

### Description

Print function for class 'mreg'

### Usage

```
## S3 method for class 'mreg'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "mreg". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## ppo *Percentage Price oscillator*

---

### Description

Compute Percentage Price oscillator (Technical Analysis)

### Usage

```
ppo(X, fast.lag = 10, slow.lag = 30, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| fast.lag | fast.lag |
| slow.lag | slow.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## pptfopt *Print Optimum Portfolio*

---

### Description

Print function for Optimum Ptf

### Usage

```
## S3 method for class 'PtfOpt'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "PrfOpt". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## prbsar *Parabolic Stop and Reverse (PSAR)*

---

### Description

Compute Parabolic Stop and Reverse (PSAR) (Technical Analysis)

### Usage

```
prbsar(Close, High, Low, accel = c(0.02, 0.2), plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| accel | accel |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## preder *Prediction error*

### Description

Calculate Prediction error for model evaluation

### Usage

```
pred_error(actual, predicted, pc = FALSE)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |
| pc | pc |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## predgar *Predict Garch model*

### Description

Predict Garch model

### Usage

```
## S3 method for class 'Garch'
predict(object, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| object | OBJECT of class "Garch". |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## predmreg *Predict method for Multiple regressions*

### Description

Predict function for class 'mreg'

### Usage

```
## S3 method for class 'mreg'
predict(object, ...)
```

### Arguments

| | |
|---|---|
| object | OBJECT of class "mreg". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## predreg *Predict method for regression*

### Description

Predict method for class 'reg'

### Usage

```
## S3 method for class 'reg'
predict(object, newdata = NULL, ci = 0.95, mode = c("response", "link"),
plot = FALSE, shaded = FALSE, xlabels = NULL, main = "Linear Model Prediction",
getThemeAttr("col", exact = TRUE)[c(1, 2, 2)],
shade.stripes = 1, shade.col = getThemeAttr("col", exact = TRUE)[2],
shade.density = 40, shade.angle = 30, legend = NULL, ...)
```

### Arguments

| | |
|---|---|
| object | OBJECT of class "reg". |
| newdata | newdata |
| ci | ci |
| mode | mode |
| plot | LOGICAL. If TRUE plot is returned. |
| shaded | shaded |
| xlabels | xlabels |
| main | main |
| col | color |
| shade.stripes | shade.stripes |
| shade.col | shade.col |
| shade.density | shade.density |
| shade.angle | shade.angle |
| legend | legend |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## predvear *VAR predictions*

---

### Description

Predict VAR model

### Usage

```
## S3 method for class 'VecAr'
predict(object, steps = 5, CI = 0.95, viewby = c("vars", "step"), ...)
```

### Arguments

| | |
|---|---|
| object | OBJECT of class "VecAr". |
| steps | steps |
| CI | CI |
| viewby | viewby |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## printvar *Print VaR results*

---

### Description

Print method for class 'VaR'

### Usage

```
## S3 method for class 'VaR'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | Instance of class 'VaR' |
| ... | Further arguments to and from other methods |

### Value

Void

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## pro *Price oscillator*

---

### Description

Compute Price oscillator (Technical Analysis)

### Usage

```
pro(Close, fast.lag = 5, slow.lag = 10, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| fast.lag | fast.lag |
| slow.lag | slow.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Price Oscillator Percent shows the percentage difference between two moving averages. A buy signal is generate when the Price Oscillator Percent rises above zero, and a sell signal when the it falls below zero.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## probhit *Probability of first hit (Brownian motion)*

---

### Description

Calcualte probability to Hit a barrier

### Usage

```
ProbHit(B = 0, S0 = 0, mi, sigma)
```

### Arguments

| | |
|---|---|
| B | B |
| S0 | S0 |
| mi | mi |
| sigma | sigma |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## psme *Print Sample Mean Excess class*

### Description

Printing function for Sample Mean Excess class

### Usage

```
## S3 method for class 'sme'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "sme". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## ptfbeta *Portfolio Beta*

### Description

Get portfolio Beta

### Usage

```
PtfBeta(beta, w = NULL, glob = TRUE)
```

### Arguments

| | |
|---|---|
| beta | beta |
| w | w |
| glob | glob |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## ptfopt *Mean-Variance optimum portfolio*

### Description

Calculate mean-variance efficient portfolio

### Usage

```
PtfOpt(ret = NULL, ptf = NULL, mi = NULL, SIGMA = NULL, volatility = TRUE, ...)
```

### Arguments

| | |
|---|---|
| ret | ret |
| ptf | ptf |
| mi | mi |
| SIGMA | SIGMA |
| volatility | volatility |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## ptfret *Portfolio returns*

### Description

Calculate portfolio returns

### Usage

```
PtfRet(PTF, w = NULL, glob = TRUE, calc.ret = FALSE, ...)
```

### Arguments

| | |
|---|---|
| PTF | PTF |
| w | w |
| glob | glob |
| calc.ret | calc.ret |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## ptfront *Portfolio efficient frontier*

---

### Description

Compute / Simulate portfolio mean-variance efficient frontier

### Usage

```
PtfFront(PTF = NULL, ret = NULL, SIGMA = NULL, mi = NULL, n_sim = 10,
volatility = TRUE, plot = TRUE, main = paste("Frontier Simulation:",
ifelse(is.null(mi), n_sim, length(mi)), "points"), xtitle = ifelse(volatility,
expression(sigma), expression(sigma^2)), ytitle = expression(mu), xlab.srt =
0, ytitle.srt = 0, type = "o", legend = "Mean-Variance Frontier", ...)
```

### Arguments

| | |
|---|---|
| PTF | PTF |
| ret | ret |
| SIGMA | SIGMA |
| mi | mi |
| n_sim | n_sim |
| volatility | volatility |
| plot | plot |
| main | main |
| xtitle | xtitle |
| ytitle | ytitle |
| xlab.srt | xlab.srt |
| ytitle.srt | ytitle.srt |
| type | type |
| legend | legend |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`ptfutil` *Portfolio Utility*

## Description

Calculate utility and plot for efficient portfolio

## Usage

```
PtfUtility(PTF = NULL, W, R = NULL, SIGMA = NULL,
af = 3, plot = TRUE, plot.mv = FALSE,
plot.mu = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `PTF` | PTF |
| `W` | W |
| `R` | R |
| `SIGMA` | SIGMA |
| `af` | af |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `plot.mv` | plot.mv |
| `plot.mu` | plot.mu |
| `...` | Further arguments to or from other methods. |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`ptfvar` *Portfolio variance*

## Description

Calculate portfolio variance

## Usage

```
PtfVar(PTF, w = NULL, glob = TRUE,
vol = FALSE, calc.ret = FALSE, ...)
```

### Arguments

| | |
|---|---|
| PTF | PTF |
| w | w |
| glob | glob |
| vol | vol |
| calc.ret | calc.ret |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## pvecar *Print VAR*

---

### Description

Print method for VAR

### Usage

```
## S3 method for class 'VecAr'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | OBJECT of class "VecAr". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## pvt — *Price Volume trend indicator*

### Description

Compute Price Volume trend indicator (Technical Analysis)

### Usage

```
pvt(Close, Volume, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| Volume | VECTOR. Asset traded Volume. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

Price Volume Trend combines percentage price change and volume to confirm the strength of price trends or through divergences, warn of weak price moves. Unlike other price-volume indicators, the Price Volume Trend takes into consideration the percentage increase or decrease in price, rather than just simply adding or subtracting volume based on whether the current price is higher than the previous days price.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## qgev — *Generalised Extreme Value (GEV) - Quantile function*

### Description

Generalised Extreme Value (GEV) - Quantile function

### Usage

```
qgev(P, mu = 0, xi = 0.1, sigma = 1)
```

### Arguments

| | |
|---|---|
| P | P |
| mu | mu |
| xi | xi |
| sigma | sigma |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## qgpd *Generalised Pareto Distribution (GPD) - Quantile function*

---

### Description

Generalised Pareto Distribution (GPD) - Quantile function

### Usage

```
qgpd(P, xi = 0.1, sigma = 1, trsh = 0)
```

### Arguments

| | |
|---|---|
| P | P |
| xi | xi |
| sigma | sigma |
| trsh | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## radpkg *Financial Technical Analysis and Risk Management*

---

### Description

R-Adamant is a collection of functions and algorithms for processing of Financial Time Series, Risk Management and Econometrics.

### Details

| | |
|---|---|
| Package: | RAdamant |
| Type: | Package |
| Version: | 0.7.1 |
| Date: | 2011-06-25 |
| License: | GPL>=2 |
| LazyLoad: | yes |

## Author(s)

RAdamant Development Team Maintainer: RAdamant Development Team <team@r-adamant.org>

## References

Farewell friend, I was a thousand times more evil than thou...

---

`recycle` *Recycle function for time series*

---

## Description

Recycle an input sequence X to get a new sequence of the specified length V

## Usage

```
recycle(X, V = length(X))
```

## Arguments

| | |
|---|---|
| `X` | X |
| `V` | V |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## relvol *Relative Volatility oscillator*

### Description

Compute Relative Volatility oscillator (Technical Analysis)

### Usage

```
RelVol(Close, sdlag = 9, lag = 5)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| sdlag | sdlag |
| lag | INTEGER. Number of lag periods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## rema *Regularised Exponential Moving Averages*

### Description

Compute multiple Regularised Exponential Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
rema(X, win.size = NROW(X), alpha = 0.5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| alpha | weight in the interval [0, 1]. (DEFAULT: 0.7). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters for future development. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
REMA is a second order IIR filter with the two coefficients are regulated by the smoothing factors lambda and alpha.
Smoothing factors: lambda = 2/(win.size+1) and alpha.

### Value

A object of class 'ma' with attributes type = "REMA", 'lambda' and 'alpha':
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

## rgev *Generalised Extreme Value (GEV) - Random Numbers Generator*

---

### Description

Generalised Extreme Value (GEV) - Random Numbers Generator

### Usage

```
rgev(N, mu = 0, xi = 0.1, sigma = 1)
```

### Arguments

| | |
|---|---|
| N | N |
| mu | mu |
| xi | xi |
| sigma | sigma |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## rgpd *Generalised Pareto Distribution (GPD) - Random Numbers Generator*

### Description

Generalised Pareto Distribution (GPD) - Random Numbers Generator

### Usage

```
rgpd(n, xi = 0.1, sigma = 1, trsh = 0)
```

### Arguments

| | |
|---|---|
| n | n |
| xi | xi |
| sigma | sigma |
| trsh | trsh |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## roc *Rate of Change index*

### Description

Compute Rate of Change index (Technical Analysis)

### Usage

```
roc(X, lag = 5, pc = TRUE, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| pc | pc |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

The Rate of Change function measures rate of change relative to previous periods. The function is used to determine how rapidly the data is changing. The factor is usually 100, and is used merely to make the numbers easier to interpret or graph. The function can be used to measure the Rate of Change of any data series, such as price or another indicator. When used with the price, it is referred to as the Price Rate Of Change, or PROC.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## rocplot *ROC analysis*

---

### Description

Plot ROC curve

### Usage

```
ROCplot(x, ...)
## S3 method for class 'scorecard'
ROCplot(x, ...)
```

### Arguments

| | |
|---|---|
| x | x |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## rowmax *Maximum / Minimum by row*

### Description

`rowMax`: Compute parallel max across the rows of X
`rowMin`: Compute parallel min across the rows of X

### Usage

```
rowMax(X)
rowMin(X)
```

### Arguments

| | |
|---|---|
| `X` | Input matrix/sequence |

### Value

A matrix NROW(X) by one, where each row is the max / min of the rows of X).

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## rschint *Interval for uniroot function*

### Description

Compute a proper search interval for uniroot function

### Usage

```
root.search.interval(from, func = NULL,
type = c("left", "both", "right"), max.iter = 500,
show.warnings = FALSE, debug = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `from` | from |
| `func` | func |
| `type` | type |
| `max.iter` | max.iter |
| `show.warnings` | show.warnings |
| `debug` | debug |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`rsi` *Relative strength indicator*

---

### Description

Compute Relative strength indicator (Technical Analysis)

### Usage

```
rsi(X, lag, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `lag` | INTEGER. Number of lag periods. |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Details

The Relative Strength Index (RSI) is an oscillator that measures current price strength in relation to previous prices.
The RSI is a versatile tool, it can be used to generate buy and sell signals, show overbought and oversold conditions, confirm price movement and warn of potential price reversals through divergences.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## runlog *Error Handling and Log with runner*

### Description

`write.log`: Simple function to write/append log to file (csv format).
`error.handling`: Error handling function

### Usage

```
write.log(log = matrix(NA, nrow = 0, ncol = 0), logfile = "runlog.log")
error.handling(err)
```

### Arguments

| | |
|---|---|
| `log` | Matrix containing logging information. |
| `logfile` | Filename of the log |
| `err` | List containing the status code of the error. |

### Details

Function `error.handling` is to be called ONLY inside a tryCatch statement.
It assigns three variables:
- log.status = "Failed": the status of the execution is set to "Failed"
- log.message: The error message generated inside the tryCatch
- res = NA: the result is set to NA

### Value

VOID

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

`run`, `multirun`

## runner *Runner and Multirunner*

### Description

Wrapper function to execute any function.
Run single or multiple functions and provide a list of results.

## Usage

```
run(func = NULL, args = list(), writelog = TRUE,
logfile = "runlog.log", check.input = TRUE,
output = c("console", "sing.file"))

multirun(func.array = character(0), args.list = list(),
writelog = TRUE, logfile = "runlog.log",
output = c("console", "sing.file", "multi.file"))
```

## Arguments

| | |
|---|---|
| `func` | Name of the function to run |
| `func.array` | Array of function names to execute |
| `args` | Named list of parameters of the function.<br>Each entry is of the form: args[["PARAM.NAME"]] = VALUE. |
| `args.list` | Array of named list of parameters of the function.<br>Each entry is a list of parameters, as required by the wrapper function "run". |
| `writelog` | LOGICAL. If TRUE, execution log is written to file. |
| `logfile` | Filename of the log |
| `check.input` | LOGICAL. If TRUE, basic checks are performed on input data, and stop code execution in case of wrong data. |
| `output` | Choose wether to return the results in the console or export the to text file. |

## Details

When called the function `multirun` the elements of the argument `args.list` can be specified with or without names. If the names are specified the arguments can be put in a different order from the array function.
If `writelog = TRUE` a log containing information about submitted computation is saved in the current working directory. If `output` = "sing.file", a text file containing all the results is saved in current working directory.
The file will be named "Run_time_date.txt" If `output` = "sing.file", a text for each called function is saved in a text file.
The files will be named "Function Name_time_date.txt"

## Value

The object returned depends on the function being called.
`multirun` returns a list of results, One entry for each function being executed.

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## See Also

`write.log`, `error.handling`

### Examples

```
# Run Exponential Moving Average and Simple Moving Average.
# For each function a list of parameters has been specified
multirun(c("ema","sma")
,list( list(rnorm(150), 5), list(rnorm(100), 10) )
, writelog = TRUE
)
# Specifies names in the list of arguments
multirun(func.array=c("ema","sma")
,args.list=list( sma=list(rnorm(150), 5), ema=list(rnorm(100), 30) )
, TRUE
)
# Output to text file
multirun(func.array=c("ema","sma")
,args.list=list( sma=list(rnorm(150), 5), ema=list(rnorm(100), 30) )
, output = "multi.file"
)
```

---

## rvi *Relative Vigor indicator*

---

### Description

Compute Relative Vigor indicator (Technical Analysis)

### Usage

```
rvi(Close, High = NULL, Low = NULL, Open = NULL, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Open | VECTOR. Open price. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Details

An indicator used in technical analysis that measures the conviction of a recent price action and the likelihood that it will continue. The RVI compares the positioning of a security closing price relative to its price range, and the result is smoothed by calculating an exponential moving average of the values.

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`sampmom` *Sample moments (Brownian motion)*

## Description

Calculate sample moments of a Brownian motion

## Usage

```
SampMom(P, X, moms = 1:2)
```

## Arguments

| | |
|---|---|
| `P` | P |
| `X` | X |
| `moms` | moms |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

`scaledf` *Apply functions on a scaled window*

## Description

`scalApply`: Applies a given function to the pairs (X[n, i], X[n-lag, i]).
`scalMax`: Scaled max on each column of the input matrix. `scalMin`: Scaled min on each column of the input matrix

## Usage

```
scalApply(X, lag = 0, padding = NA, na.rm = FALSE, func = NULL, ...)
scalMax(X, lag = 1, padding = -Inf, na.rm = FALSE, func = NULL)
scalMin(X, lag = 1, padding = Inf, na.rm = FALSE, func = NULL)
```

## Arguments

| | |
|---|---|
| `X` | Input matrix/sequence |
| `lag` | vector of integer lags. If lag >= 0 data are shifted to the right, else to the left. (DEFAULT = 0) |
| `padding` | value used to initialise the output matrix (DEFAULT = NA) |
| `na.rm` | LOGICAL. If TRUE, N-lag entries are removed from the output (DEFAULT = FALSE) |
| `func` | function applied to the data (DEFAULT = NULL) |
| `...` | Additional parameters accepted by the function 'func' |

### Details

Sequences are treated as one-column matrices.

### Value

A matrix where func / max / min has been applied on each pair (X[n, i], X[n-lag, i]) for each column i of X. Number of rows depends on the na.rm parameter. Number of columns is NCOL(X)

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## scorecd *Score Card*

---

### Description

Create Credit Score Card based on Logistic Regression

### Usage

```
Score.card(Y, X, nseg = 2)
```

### Arguments

| | |
|---|---|
| Y | VECTOR. Target variable in 0-1 format. |
| X | DATA.FRAME / MATRIS of regressors. |
| nseg | INTEGER / VECTOR. Number of segments to factorise numerical variables. |

### Details

For the moment only numerical variables! Work in progress...

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sde *Error standard deviation*

### Description

Calculate Error standard deviation for model evaluation

### Usage

```
sde(actual, predicted)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sensan *Sensitivity analysis default method*

### Description

Sensitivity analysis default method

### Usage

```
sensAnalysis(X, ...)
## Default S3 method:
sensAnalysis(X, win.size = length(coef(X)), plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| win.size | win.size |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sensanlm *Sensitivity analysis method for lm*

### Description

Sensitivity analysis method for lm

### Usage

```
## S3 method for class 'lm'
sensAnalysis(X, ...)
```

### Arguments

| | |
|---|---|
| X | OBJECT of class "lm". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sensanrg *Sensitivity analysis method for reg*

### Description

Sensitivity analysis method for reg

### Usage

```
## S3 method for class 'reg'
sensAnalysis(X, ...)
```

### Arguments

| | |
|---|---|
| X | OBJECT of class "reg". |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sharpe *Sharpe index*

### Description

`Sharpe`: Calculate Sharpe index for a portfolio.
`Sharpe.Capm`: Get Sharpe index from an object of class. "Capm"

### Usage

```
Sharpe(PTF, ...)
## Default S3 method:
Sharpe(PTF, rfr = 0, ...)
## S3 method for class 'Capm'
Sharpe(PTF, rfr = 0, ...)
```

### Arguments

| | |
|---|---|
| `PTF` | Input portfolio or an object of class "Capm" |
| `rfr` | risk free rate |
| `...` | Further arguments to or from other methods |

### Author(s)

RAdamant Development Team `<team@r-adamant.org>`

### See Also

`Treynor`, `Jensen`, `Appraisal`

## sinma *(Normalised) Sine Weighted Moving Averages*

### Description

Compute multiple (Normalised) Sine Weighted Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
sinma(X, win.size = 10, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable). |
| `win.size` | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = 10). |
| `plot` | LOGICAL. Return plot. |
| `...` | Further arguments to or from other methods |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
Weights: sin(pi * (1:win.size)/(win.size+1))

### Value

A object of class 'ma' with attributes type = "SINMA" and 'win.size' as from the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

Movav

---

## sma *Simple Moving Average*

---

### Description

Compute multiple Simple Moving Averages on the input data, one for each column of X[, i] and window size win.size[j]

### Usage

```
sma(X, win.size = 10, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = 10). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by the function Mmovav. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A object of class 'ma' with attributes type = "SMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

ema

---

## sme *Sample Mean Excess function*

---

### Description

Sample Mean Excess function

### Usage

```
sme(X, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| plot | plot |
| ... | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## somerd *Somers D*

---

### Description

Calculate Somers d statistic

### Usage

```
SomerD(target, pred, ...)
```

### Arguments

| | |
|---|---|
| target | target |
| pred | pred |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## sort *Sort matrix*

---

### Description

Sort each column of the input matrix X independently

### Usage

```
SORT(x, decreasing = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | Input matrix. |
| decreasing | LOGICAL. Decreasing order. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## specgram *Spectrogram*

---

### Description

Create spectrogram for FFT

### Usage

```
specgram(X, win.size = max(1, NROW(X)/20), plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| win.size | win.size |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`splitwdw` *Sliding windows*

---

### Description

Sliding windows

### Usage

```
splitWindow(N, direction = c("forward", "backward"),
mode = c("EW", "SW"), from = NULL,
win.size = 1, by = 1, labels = 1:N, ...)
```

### Arguments

| | |
|---|---|
| `N` | N |
| `direction` | direction |
| `mode` | mode |
| `from` | from |
| `win.size` | win.size |
| `by` | by |
| `labels` | labels |
| `...` | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

`sssym` *State Space system simulation*

## Description

Generic function for State Space system simulation. The system can be either linear or non linear.

## Usage

```
ss.sym(X, F = NULL, G = NULL, H = NULL, D = NULL,
init = 0, SLen = ifelse(is.function(F), NA,
NROW(F)), YLen = ifelse(is.function(H), NA, NROW(H)), ...)
```

## Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable). |
| `F` | [State -> State] transition matrix or [(State, Input) -> State] function (F = function(S, X, n, ...)<br>returning the new state vector S_new based on the current State S and the data X at time period n) (DEFAULT = NULL) |
| `G` | [Input -> State] transition matrix. Only for linear models (DEFAULT = NULL) |
| `H` | [State -> Output] transition matrix or [(State, Input) -> Output] function (H = function(S, X, n, ...)<br>returning the new output vector Y[, n] based on the new state S[, n] and the data X at time period n) (DEFAULT = NULL -> converted in diag(SLen)) |
| `D` | [Input -> Output] transition matrix. Only for linear models (DEFAULT = NULL -> converted to a zero matrix SLen by NCOL(X) ) |
| `init` | Initial values for the state vactor S (DEFAULT = 0, recycled to length SLen if necessary) |
| `SLen` | Length of the state vector S. (DEFAULT = ifelse(is.function(F), NA, NROW(F)) ) |
| `YLen` | Number of columns of the output vector Y. (DEFAULT = ifelse(is.function(H), NA, NROW(H)) ) |
| `...` | Additional parameters accepted by the functions F and H |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.

## Value

A object of class 'ss' with attributes 'F', 'G', 'H', 'D' as given by the corresponding input parameters:
- matrix of size NROW(X) by YLen, result of the symulation of the given dynamic system subject to input 'X' and initial condition 'init'.

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## starc *Stoller Starc bands*

### Description

Compute Stoller Starc bands (Technical Analysis)

### Usage

```
starc(Close, High = NULL, Low = NULL, atr.mult = 2, lag = 5, atr.lag =
14, mov = c("sma", "ema", "wma"), plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| atr.mult | atr.mult |
| lag | INTEGER. Number of lag periods. |
| atr.lag | atr.lag |
| mov | mov |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## statbar *Status bar*

### Description

Interactive status bar for console logging

### Usage

```
statusbar(message = "Computing..", status = 0, n = 1, N = 1, step = 0.01)
```

### Arguments

| | |
|---|---|
| message | message |
| status | status |
| n | n |
| N | N |
| step | step |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## stepmat *Step matrix for binomial tree*

---

### Description

Simulate binomial path of a binomial tree

### Usage

```
StepMat(init, n_step, up, down)
```

### Arguments

| | |
|---|---|
| init | init |
| n_step | n_step |
| up | up |
| down | down |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## strvar *Structural Vector Autoregressive model*

### Description

Estimate Structural Vector Autoregressive model

### Usage

```
Strvar.VecAr(X, A = NULL, B = NULL, inter = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| A | A |
| B | B |
| inter | inter |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## styles *Styles analysis (portfolio)*

### Description

Perform Style analysis

### Usage

```
Styles(FUND, IND, W, lower = NULL, upper = NULL, ...)
```

### Arguments

| | |
|---|---|
| FUND | FUND |
| IND | IND |
| W | W |
| lower | lower |
| upper | upper |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## sumdd *Summary drawdown*

---

### Description

Summary function for drawdown

### Usage

```
SummaryDD(DD)
```

### Arguments

| | |
|---|---|
| DD | OBJECT of class "drawdown" |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## sumdens *Plot summary information*

---

### Description

Plot summary information of a vector with its density

### Usage

```
Sum.dens(x, ...)
```

### Arguments

| | |
|---|---|
| x | VECTOR. Input series. |
| ... | further arguments for "plot" function |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sumreg *Summary method for mreg*

### Description

Summary method for mreg

### Usage

```
## S3 method for class 'mreg'
summary(object, ...)
```

### Arguments

| | |
|---|---|
| object | OBJECT of class "mreg" |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## sumscrd *Summary Scorecard*

### Description

Summary method for "Scorecard" object

### Usage

```
## S3 method for class 'scorecard'
summary(object, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| object | OBJECT of class "scorecard". |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## swing *Swing Index*

### Description

Calculate Swing index (Technical Analysis)

### Usage

```
Swing(Close, High, Low, Open, ret_cum = FALSE, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Open | VECTOR. Open price. |
| ret_cum | ret_cum |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## symlkup *Lookup Stock Symbol from Yahoo!*

### Description

Lookup Stock Symbol from Yahoo!

### Usage

```
symbol.lookup(what = "")
```

### Arguments

| | |
|---|---|
| what | what |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

tema *Triple EMA*

## Description

Compute multiple Triple EMA on the input data, one for each column of X[, i] and window size win.size[j].

## Usage

```
tema(X, win.size = NROW(X), plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function ema. |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.
TEMA is a weighted combination of EMA: 3*EMA(X) - 3* EMA(EMA(X)) + EMA(EMA(EMA(X))).
Smoothing factor: lambda = 2/(win.size+1).

## Value

A object of class 'ma' with attributes type = "TEMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## See Also

ema

## thigh *True High oscillator*

### Description

Compute True High oscillator (Technical Analysis)

### Usage

```
thigh(Close, High = NULL, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## tirlev *Trione levels*

### Description

Compute Trione levels (Technical Analysis)

### Usage

```
tirLev(High, Low, Close, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| Close | VECTOR. Close price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## tlow *True Low oscillator*

---

### Description

Compute True Low oscillator (Technical Analysis)

### Usage

```
tlow(Close, Low = NULL, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| Low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

## tma *Triangular Moving Averages*

---

### Description

Compute multiple Triangular Moving Averages on the input data, one for each column of X[, i] and window size
win.size[j]

### Usage

```
tma(X, win.size = 10, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = 10). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by the function Mmovav. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.

### Value

A object of class 'ma' with attributes type = "TMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

Movav

---

## treynor *Treynor index*

---

### Description

Treynor: Calculate Treynor index for a portfolio
Treynor.Capm: Get Treynor index from an object of class "Capm"

### Usage

```
Treynor(PTF, ...)
## Default S3 method:
Treynor(PTF, PTF_M, rfr = 0, rf = NULL, ...)
## S3 method for class 'Capm'
Treynor(PTF, rfr = 0, ...)
```

### Arguments

| | |
|---|---|
| PTF | Input portfolio or an object of class "Capm" |
| PTF_M | Market/benchmark portfolio |
| rfr | risk free rate |
| rf | risk free asset |
| ... | Further arguments to or from other methods |

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

Jensen, Sharpe, Appraisal

---

## trf *(Average) True range*

---

### Description

Compute (Average) True range (Technical Analysis)

### Usage

```
trf(Close, High = NULL, Low = NULL, lag = 1,
average = TRUE, avg.lag = 14, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |
| average | average |
| avg.lag | avg.lag |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## trsigexp *Exponential track signal*

### Description

Calculate Exponential track signal for model evaluation

### Usage

```
track_sign_exp(actual, predicted, beta = 0.1)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |
| beta | beta |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## trsign *Track signal*

### Description

Calculate Track signal for model evaluation

### Usage

```
track_sign(actual, predicted)
```

### Arguments

| | |
|---|---|
| actual | actual |
| predicted | predicted |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

ttma *T3 EMA*

## Description

Compute multiple T3 EMA on the input data, one for each column of X[, i] and window size win.size[j].

## Usage

```
ttma(X, win.size = NROW(X), alpha = 0.7, plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| alpha | weight in the interval [0, 1]. (DEFAULT: 0.7). |
| plot | LOGICAL. Return plot. |
| ... | Additional parameters accepted by function ema. |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.
T3 EMA is a three times application of GDEMA: GDEMA(GDEMA(GDEMA(X, alpha), alpha), alpha).
Smoothing factor: lambda = 2/(win.size+1).

## Value

A object of class 'ma' with attributes type = "TTMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## See Also

ema, gdema

## typ *Typical price*

### Description

Compute Typical price (Technical Analysis)

### Usage

```
tyP(Close, High, Low, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## ulcer *Ulcer index*

### Description

Compute Ulcer index (Technical Analysis)

### Usage

```
ulcer(X, lag, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`ultima` *Ultima oscillator*

---

### Description

Compute Ultima oscillator (Technical Analysis)

### Usage

```
ultima(Close, High = NULL, Low = NULL, lag = 1, win1 = 7, win2 = 14, win3 = 28,
...)
```

### Arguments

| | |
|---|---|
| `Close` | VECTOR. Close price. |
| `High` | VECTOR. High price. |
| `Low` | VECTOR. Low price. |
| `lag` | INTEGER. Number of lag periods. |
| `win1` | win1 |
| `win2` | win2 |
| `win3` | win3 |
| `plot` | LOGICAL. If TRUE plot is returned. |
| `...` | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

univar *Univariate analysis*

## Description

Perform univariate analisys of the dependent variable Y versus each independent variable X, plotting the results

## Usage

```
univar(Y, X, stress.period.idx = c(),
Y.logit = FALSE, Y.logit.adj = 5e-05,
theme.params = getCurrentTheme(),
plot = TRUE, overrides = list(...), ...)
```

## Arguments

| | |
|---|---|
| Y | serie of the dependent variable |
| X | Matrix containing all independent variables (one column per variable) |
| stress.period.idx | vector of positions specifing the stress regime. If provided, the system will run a modified LS to capture the two regimes |
| Y.logit | LOGICAL. If TRUE, the dependent variable is transformed using the Logit transform. The results are retransformed using the inverse Logit. (DEFAULT: FALSE) |
| Y.logit.adj | Cut-off value. The range of the Y variable is restricted within the interval [Y.logit.adj, 1-Y.logit.adj] (DEFAULT: 0.00005) |
| theme.params | Theme parameters (DEFAULT: getCurrentTheme()) |
| plot | list of parameters to override the theme. Must match by name the parameters defined by the theme (DEFAULT: NULL) |
| overrides | LOGICAL. If TRUE, results are plotted. |
| ... | Further arguments to or from other methods |

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## See Also

plot.univar, print.univar

`var` *Value at Risk*

## Description

General VaR, computed on each column of the input matrix

## Usage

```
VaR(X, ...)
## Default S3 method:
VaR(X, p = 0.05, probf = c("norm","t","cofi"),
df = max(4, (kurt(X)+3)), params = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `X` | Input matrix/sequence. Sequences are treated as one column matrices. |
| `p` | vector of probabilities (DEFAULT = 0.05) |
| `probf` | probability dristribution, see Details |
| `df` | Degrees of freedom for the Student T distribution (DEFAULT = max(4, (kurt(X)+3))) |
| `params` | additional parameter for future development |
| `...` | Additional parameters accepted by the function cofit |

## Details

Accepted probability distributions:
- "norm" = Normal distribution
- "t" = Students T distribution
- "cofi" = Cornish-Fischer distribution

## Value

General VaR, computed on each column of the input matrix

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

`varptf` *Portfolio Value at Risk*

## Description

General VaR, computed for an input portfolio

## Usage

```
VaRPtf(X, p = 0.05, weights = rep(1/NCOL(X), NCOL(X)), probf = c("norm","t"), df
```

## Arguments

| | |
|---|---|
| `X` | Input matrix/sequence. Sequences are treated as one column matrices. |
| `p` | vector of probabilities (DEFAULT = 0.05) |
| `probf` | probability dristribution, see Details |
| `weights` | portfolio weigths (DEFAULT = rep(1/NCOL(X), NCOL(X))) |
| `df` | D egrees of freedom for the Student T distribution (DEFAULT = 4) |
| `...` | Additional parameters for future development |

## Details

Accepted probability distributions:
- "norm" = Normal distribution
- "t" = Students T distribution
- "cofi" = Cornish-Fischer distribution

## Value

A matrix length(p) by 1 of computed portfolio VaRs

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team `<team@r-adamant.org>`

## vcmof *Variable Chande Momentum Oscillator*

### Description

Compute Variable Chande Momentum Oscillator (Technical Analysis)

### Usage

```
vcmof(X, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## vecar *Vector Autoregressive model*

### Description

Estimate Vector Autoregressive model

### Usage

```
VecAr(X, ...)

## Default S3 method:
VecAr(X, ar.lags = 1:2,
type = c("const", "trend", "constrend", "none"),
exog = NULL, ...)
```

### Arguments

| | |
|---|---|
| X | Input matrix of time series.<br>N.B. The first column is taken as dependent variable |
| ar.lags | Number (or vector) of lags for the AR components |
| type | Type of deterministic regressor(s) to be included in the model |
| exog | matrix of exogenous variables (Default = NULL) |
| ... | Further arguments to or from other methods |

### Value

An object list of class "VecAr". The list contains the following elements:
- Results of the estimation ("lm" object)
- Nunmber of Observations
- Number of Variables
- Number of Parameters
- LogLikelihood value
- AIC information criteria
- BIC information criteria

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

Strvar.VecAr, fitted.VecAr

---

## vhff *Vertical Horizontal Filter*

---

### Description

Compute Vertical Horizontal Filter (Technical Analysis)

### Usage

```
vhff(X, lag = 9, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## vidyaf *Variable Index Dynamic Average*

### Description

Compute Variable Index Dynamic Average (Technical Analysis)

### Usage

```
vidyaf(X, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | X |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## vwma *Volume Weighted Moving Averages*

### Description

Compute multiple Volume Weighted Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
vwma(X, Vol = NULL, win.size = 10, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| X | Matrix of data series (one column per variable). |
| Vol | Matrix of volumes (one column per variable). |
| win.size | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = 10). |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
If X is a financial time series (class = 'fs'), and Vol = NULL then Vol = X[, 'Volume'] (DEFAULT = NULL).

### Value

A object of class 'ma' with attributes type = "VWMA" and 'win.size' as from the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

sma

---

## wad *Williams Advance Decline*

---

### Description

Compute Williams Advance Decline (Technical Analysis)

### Usage

```
wad(Close, High = NULL, Low = NULL, lag = 5, na.rm = FALSE, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |
| na.rm | na.rm |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## weigevid *Weight of Evidence*

### Description

Calculate weight of evidence for a matrix with target variable

### Usage

```
WeightEvid(data, target, nseg, missing = FALSE, na.replace = NULL)
```

### Arguments

| | |
|---|---|
| data | data |
| target | target |
| nseg | nseg |
| missing | missing |
| na.replace | na.replace |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## whvar *Weighted Historical Value at Risk*

### Description

Compute Weighted historical VaR on each column of the input matrix

### Usage

```
whVaR(X, p = 0.05, lambda = 0.9, centered = FALSE)
```

### Arguments

| | |
|---|---|
| X | Input matrix/sequence. Sequences are treated as one column matrices. |
| p | vector of probabilities (DEFAULT = 0.05) |
| lambda | controls the exponential window lambda^((NROW(X)-1):0) (DEFAULT = 0.9) |
| centered | LOGICAL. If TRUE, input data are standardised |

### Value

A matrix length(p) by NCOL(X) of computed quantiles

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`wildavg` *Wilder Moving Average*

---

### Description

Compute Wilder Moving Average (Technical Analysis)

### Usage

```
wildAvg(X, lag = 5, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | X |
| `lag` | lag |
| `plot` | plot |
| `...` | ... |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`wildsum` *Wilder Summation*

---

### Description

Compute Wilder Summation (Technical Analysis)

### Usage

```
wildSum(x, lag = 5)
```

### Arguments

| | |
|---|---|
| `x` | x |
| `lag` | lag |

## Note

TO BE COMPLETED

## Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`wma` *Weighted Moving Averages*

---

## Description

Compute multiple Weighted Moving Averages on the input data, one for each column of X[, i] and window size
win.size[j]

## Usage

```
wma(X, win.size = 10, plot = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable). |
| `win.size` | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = 10). |
| `plot` | LOGICAL. Return plot. |
| `...` | Additional parameters accepted by the function Mmovav. |

## Details

For financial time series (class = 'fs'), only 'Close' column is processed.

## Value

A object of class 'ma' with attributes type = "WMA" and 'win.size' as given by the corresponding input parameter:
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

## Author(s)

RAdamant Development Team <team@r-adamant.org>

## wro *Williams R*

### Description

Compute Williams R (Technical Analysis)

### Usage

```
wro(Close, High = NULL, Low = NULL, lag = 5, plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| Close | VECTOR. Close price. |
| High | VECTOR. High price. |
| Low | VECTOR. Low price. |
| lag | INTEGER. Number of lag periods. |
| plot | LOGICAL. If TRUE plot is returned. |
| ... | Further arguments to or from other methods. |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

## zind *Z index*

### Description

Compute the Z-score of X (Standardize each column of X)

### Usage

```
Zind(x, sigma = 1, mi = 2)
```

### Arguments

| | |
|---|---|
| x | x |
| sigma | sigma |
| mi | mi |

### Note

TO BE COMPLETED

### Author(s)

RAdamant Development Team <team@r-adamant.org>

---

`zlma` *Zero lag Moving Average*

---

### Description

Compute multiple Zero-Lag Exponential Moving Averages on the input data, one for each column of X[, i] and window size win.size[j].

### Usage

```
zlma(X, win.size = NROW(X), plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable). |
| `win.size` | vector of moving average window sizes (lags) to be applied on the data X. (DEFAULT = NROW(X)). |
| `plot` | LOGICAL. Return plot. |
| `...` | Additional parameters accepted by function ema. |

### Details

For financial time series (class = 'fs'), only 'Close' column is processed.
ZLMA is a combination of EMA: EMA(X) + EMA(X - EMA(X)).

### Value

A object of class 'ma' with attributes type = "EMAT" and lambda = 2/(win.size+1):
- matrix of size NROW(X) by NCOL(X)*length(win.size) where each column is the moving average of length win.size[i] of the corresponding column of X.

### Author(s)

RAdamant Development Team <team@r-adamant.org>

### See Also

`ema`

`zscore` *Z Score*

## Description

Compute the Z-score of X (Standardize each column of X)

## Usage

```
Zscore(X, means = NULL, sigma = NULL)
```

## Arguments

| | |
|---|---|
| `X` | Matrix of data series (one column per variable) |
| `means` | Mean value |
| `sigma` | Standard deviation |

## Value

Matrix of standardised variables

## Author(s)

RAdamant Development Team <team@r-adamant.org>