

Package ‘rportfolio’

October 14, 2022

Title Portfolio Theory

Version 0.0.3

Description

Collection of tools to calculate portfolio performance metrics. Portfolio performance is a key measure for investors. These metrics are important to analyse how effectively their money has been invested. This package uses portfolio theories to give investor tools to evaluate their portfolio performance. For more information see, Markowitz, H.M. (1952), <doi:10.2307/2975974>. Analysis of Investments & Management of Portfolios [2012, ISBN:978-8131518748].

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports xts, zoo, stats

Depends R (>= 2.10)

Suggests spelling

Language en-US

NeedsCompilation no

Author Anurag Agrawal [aut, cre] (<<https://orcid.org/0000-0003-2272-8273>>)

Maintainer Anurag Agrawal <agrawalanurag1999@gmail.com>

Repository CRAN

Date/Publication 2020-06-12 10:20:02 UTC

R topics documented:

alpha.capm	2
beta.capm	3
funds	3
jenson.alpha	4
markowitz.model	4
premium.active	5
ratio.information	6

ratio.sharpe	6
ratio.sortino	7
ratio.treynor	8
returns.cal	8
risk.premium	9
semi.deviation	10
tracking.error	10

Index	12
--------------	-----------

alpha.capm	<i>CAPM Alpha</i>
------------	-------------------

Description

Calculates the portfolio alpha

Usage

```
alpha.capm(R1, R2)
```

Arguments

R1	Portfolio return as xts
R2	Benchmark Returns

Details

Alpha is a term used in investing to describe a strategy's ability to beat the market, or it's "edge." Alpha is thus also often referred to as "excess return" or "abnormal rate of return," which refers to the idea that markets are efficient, and so there is no way to systematically earn returns that exceed the broad market as a whole.

Value

Returns the alpha of the portfolio

Examples

```
alpha.capm(funds$ret1, funds$rfr)
```

 beta.capm

CAPM Beta

Description

Returns the Beta of Security using the CAPM Model

Usage

beta.capm(R1, R2)

Arguments

R1	Returns data of the security
R2	Returns data of the benchmark security

Details

Beta is a measure of the volatility—or systematic risk—of a security or portfolio compared to the market as a whole.

Value

Value of the beta of the security

Examples

```
beta.capm(funds$ret1, funds$rfr)
```

 funds

Sample Portfolio Return

Description

An xts dataset to calculate the portfolio metrics in the package

Usage

funds

Format

A xts with 901 rows and 2 variables:

ret1 Portfolio Return

rfr Benchmark Return (Proxy for risk free rate)

Details

This is a sample returns data of the portfolio to understand and use the functions of the package.

jenson.alpha	<i>Jenson's Alpha</i>
--------------	-----------------------

Description

Calculates the Jenson's Alpha of the security

Usage

```
jenson.alpha(R1, R2, rf = 0)
```

Arguments

R1	Portfolio Return
R2	Benchmark Return
rf	Risk Free Rate of Return, Default: 0

Value

The Jensen's measure, or Jensen's alpha, is a risk-adjusted performance measure that represents the average return on a portfolio or investment, above or below that predicted by the capital asset pricing model (CAPM), given the portfolio's or investment's beta and the average market return.

Examples

```
jenson.alpha(funds$ret1, funds$rfr)
```

markowitz.model	<i>Markowitz Mean-Variance Model</i>
-----------------	--------------------------------------

Description

Calculates the optimum Portfolio weights

Usage

```
markowitz.model(R1, R2)
```

Arguments

R1	Portfolio Returns
R2	Benchmark Returns

Details

Modern portfolio theory (MPT), or mean-variance analysis, is a mathematical framework for assembling a portfolio of assets such that the expected return is maximized for a given level of risk.

Value

Returns the optimum portfolio weights and their risk and return profile.

Examples

```
markowitz.model(funds$ret1, funds$rfr)
```

premium.active	<i>Active Premium</i>
----------------	-----------------------

Description

Calculates the active premium

Usage

```
premium.active(R1, R2)
```

Arguments

R1	Returns of Portfolio as xts
R2	Risk Free Return as xts

Value

Calculates the active premium of the portfolio

Examples

```
premium.active(funds$ret1, funds$rfr)
```

ratio.information *Information Ratio*

Description

Calculates the information ratio of the portfolio

Usage

```
ratio.information(R1, R2)
```

Arguments

R1	Returns of the portfolio
R2	Returns of the benchmark portfolio

Details

The information ratio (IR) is a measurement of portfolio returns beyond the returns of a benchmark, usually an index, compared to the volatility of those returns.

Value

Calculates the information ratio of the portfolio

Examples

```
ratio.information(funds$ret1, funds$rfr)
```

ratio.sharpe *Sharpe Ratio*

Description

Calculates the Sharpe Ratio of the Portfolio

Usage

```
ratio.sharpe(R1, Rf = 0)
```

Arguments

R1	Portfolio Returns
Rf	Risk Free Rate of Return, Default: 0

Details

The Sharpe ratio was developed by Nobel laureate William F. Sharpe and is used to help investors understand the return of an investment compared to its risk.

Value

Calculates the Sharpe Ratio of the portfolio

ratio.sortino	<i>Sortino Ratio</i>
---------------	----------------------

Description

Calculates the Sortino Ratio

Usage

```
ratio.sortino(R1, Rf = 0)
```

Arguments

R1	Returns of the portfolio
Rf	Risk Free rate of return, Default: 0

Details

The Sortino ratio is a variation of the Sharpe ratio that differentiates harmful volatility from total overall volatility by using the asset's standard deviation of negative portfolio returns, called downside deviation, instead of the total standard deviation of portfolio returns.

Value

Gives the Sortino ratio of the portfolio

Examples

```
ratio.sortino(funds$ret)
```

ratio.treynor	<i>Treynor Ratio</i>
---------------	----------------------

Description

Calculates the Treynor ratio of a particular portfolio

Usage

```
ratio.treynor(R1, Rf = 0)
```

Arguments

R1	Returns of the portfolio
Rf	Returns of the benchmark portfolio

Details

The Treynor ratio, also known as the reward-to-volatility ratio, is a performance metric for determining how much excess return was generated for each unit of risk taken on by a portfolio.

Value

This function can be used to calculate the Treynor ratio of a portfolio.

Examples

```
ratio.treynor(funds$ret1)
```

returns.cal	<i>Annualized Returns</i>
-------------	---------------------------

Description

Returns the annualized returns of a data returns data

Usage

```
returns.cal(R1, freq = 252, geometric = TRUE)
```

Arguments

R1	Returns dataset as xts
freq	The periodicity of the dataset, Default: 252
geometric	Boolean to control the geometric returns and mean annualized returns, Default: TRUE

Details

An annualized total return is the geometric average amount of money earned by an investment each year over a given time period.

Value

Gives annualized returns of data

Examples

```
returns.cal(funds$ret1)
```

risk.premium	<i>Risk Premium of a Security</i>
--------------	-----------------------------------

Description

This function is used to calculate the risk premium of excess return over the risk free rate

Usage

```
risk.premium(R1, Rf)
```

Arguments

R1	The returns of the security as xts
Rf	The risk free rate of return as xts

Details

A risk premium is the return in excess of the risk-free rate of return an investment is expected to yield

Value

Returns the risk premium of the security

Examples

```
risk.premium(funds$ret1, funds$rfr)
```

semi.deviation	<i>Semi Deviation/ Down side Deviation</i>
----------------	--

Description

Calculates the semi deviation of the xts object

Usage

```
semi.deviation(R1)
```

Arguments

R1	Returns dataset of the portfolio
----	----------------------------------

Details

Semi-deviation is a method of measuring the below-mean fluctuations in the returns on investment.

Value

Calculates the semi deviation of the xts object

Examples

```
semi.deviation(funds$ret1)
```

tracking.error	<i>Tracking Error</i>
----------------	-----------------------

Description

Calculates the Tracking Error

Usage

```
tracking.error(R1, R2)
```

Arguments

R1	Returns of the portfolio
R2	Returns of the benchmark

Details

Tracking error is the divergence between the price behavior of a position or a portfolio and the price behavior of a benchmark.

tracking.error

11

Value

Calculates the Tracking error of the security

Examples

```
tracking.error(funds$ret1, funds$rfr)
```

Index

* datasets

funds, 3

alpha.capm, 2

beta.capm, 3

funds, 3

jenson.alpha, 4

markowitz.model, 4

premium.active, 5

ratio.information, 6

ratio.sharpe, 6

ratio.sortino, 7

ratio.treynor, 8

returns.cal, 8

risk.premium, 9

semi.deviation, 10

tracking.error, 10