

Package ‘PortfolioAnalysis’

August 1, 2020

Title Portfolio Optimization Methods

Version 1.0.6

Description Collection of functions to optimize portfolio weights using quadratic programming. This package includes different functions to compute portfolio weights based on different constraints and methods. 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 PerformanceAnalytics, stringr, stringi, plotly, ggplot2,
purrr, rvest, quantmod, rMorningStar, quadprog, dplyr, xts,
lubridate, readr, tidyr, xml2

NeedsCompilation no

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

Maintainer Anurag Agrawal <agrawalanurag1999@gmail.com>

Repository CRAN

Date/Publication 2020-08-01 10:30:02 UTC

R topics documented:

EfficientFrontier	2
optim.portfolio	3
optim.TargetReturn	4
optim.Weight.Return	5
PortfolioManager	6
Index	7

EfficientFrontier *Plot Efficient frontier*

Description

Plots the efficient frontier for the security (mean-variance portfolios)

Usage

```
EfficientFrontier(  
  asset.names,  
  increment = 100,  
  rf = 0,  
  period = c("months", "weeks", "quarters", "years")  
)
```

Arguments

asset.names	Vector of ticker of securities
increment	Number of portfolio to be generated, Default: 100
rf	Risk-free rate of return, Default: 0
period	Period for which the returns are calculated, Default: c("months", "weeks", "quarters", "years")

Details

Give and interactive effient frontier for given securities

Value

Interactive plot of the effiecnt frontier for given security/funds

Examples

```
EfficientFrontier(c('FXAIX', 'TIBFX'), period = 'weeks')
```

optim.portfolio	<i>Optimize portfolio weights</i>
-----------------	-----------------------------------

Description

Optimizes portfolio weights by minimizing the variance

Usage

```
optim.portfolio(  
  asset.names,  
  increment = 100,  
  rf = 0,  
  period = c("months", "weeks", "quarters", "years")  
)
```

Arguments

asset.names	Vector of ticker of securities
increment	Number of portfolio to be generated, Default: 100
rf	Risk-free rate of return, Default: 0
period	Period for which the returns are calculated, Default: c("months", "weeks", "quarters", "years")

Details

Minimizes the variance using the method of lagrange multiplier to calculate the portfolio weights with minimized variance for given target return.

Value

Returns a dataframe of the portfolios with different portfolio weights and different target returns and target standard deviation. Using this an investor can choose between range of portfolio to allocate funds.

Examples

```
optim.portfolio(c('FXAIX', 'TIBFX'), period = 'weeks')
```

optim.TargetReturn *Optimize portfolio weights given target returns*

Description

Optimizes portfolio weights by minimizing the variance for a given target return

Usage

```
optim.TargetReturn(  
  asset.names,  
  increment = 100,  
  rf = 0,  
  tgt.ret,  
  period = c("months", "weeks", "quarters", "years")  
)
```

Arguments

asset.names	Vector of ticker of securities
increment	Number of portfolio to be generated, Default: 100
rf	Risk-free rate of return, Default: 0
tgt.ret	Target return for given funds
period	Period for which the returns are calculated, Default: c("months", "weeks", "quarters", "years")

Details

Minimizes the variance using the method of lagrange multiplier to calculate the portfolio weights with minimized variance for given target return.

Value

Returns a dataframe of the portfolios with different portfolio weights and different target returns and target standard deviation. Using this an investor can choose between range of portfolio to allocate funds.

Examples

```
optim.TargetReturn(c('FXAIX', 'TIBFX'), period = 'weeks', tgt.ret = 0.0021)
```

optim.Weight.Return *Optimize portfolio weights given target returns and weights*

Description

Optimizes portfolio weights by minimizing the variance for a given target return and weights

Usage

```
optim.Weight.Return(  
  asset.names,  
  tgt.ret,  
  rf = 0,  
  max.wgt = 1,  
  period = c("months", "weeks", "quarters", "years")  
)
```

Arguments

asset.names	Vector of ticker of securities
tgt.ret	Target return for given funds
rf	Risk-free rate of return, Default: 0
max.wgt	Maximum weight to be allocated for one fund/security, Default: 1
period	Period for which the returns are calculated, Default: c("months", "weeks", "quarters", "years")

Details

Minimizes the variance using the method of lagrange multiplier to calculate the portfolio weights with minimized variance for given target return and given maximum weights.

Value

Returns a dataframe of the portfolios with different portfolio weights and different target returns and target standard deviation. Using this an investor can choose between range of portfolio to allocate funds.

Examples

```
optim.Weight.Return(c('FXAIX', 'TIBFX'), period = 'weeks', max.wgt = 0.8, tgt.ret = 0.0015)
```

PortfolioManager *PortfolioManager*

Description

Gives the detailed Risk-Reward Metrics as Computed by MorningStar

Usage

```
PortfolioManager(ticker, view = c("overview", "detailed"))
```

Arguments

ticker	Enter the Fund TICKER
view	Choose a View of the Portfolio, Default: c("overview", "detailed")

Details

This function can be used to examine and assess your holdings in different funds

Value

The output is a tibble that gives the overview of the risk and reward metrics of different holdings in a portfolio.

Index

EfficientFrontier, [2](#)

optim.portfolio, [3](#)

optim.TargetReturn, [4](#)

optim.Weight.Return, [5](#)

PortfolioManager, [6](#)