Package: rolloptim (via r-universe)

September 18, 2024

Type Package

Title Rolling Optimizations

Version 1.0

Date YYYY-MM-DD

Author Jason Foster

Maintainer Jason Foster <jason.j.foster@gmail.com>

Description Analytical computation of rolling optimizations for
time-series data.

License GPL (>= 2)

URL https://github.com/jasonjfoster/rollport

BugReports https://github.com/jasonjfoster/rollport/issues

Imports Rcpp, RcppParallel

LinkingTo Rcpp, RcppArmadillo, RcppParallel

SystemRequirements GNU make

Roxygen list(old_usage = TRUE)

RoxygenNote 7.2.3

Encoding UTF-8

Suggests covr, testthat, zoo, roll (>= 1.1.7), ROI,
ROI.plugin.quadprog, ROI.plugin.glpk, ROI.plugin.qpoases, CVXR

Repository https://jasonjfoster.r-universe.dev

RemoteUrl https://github.com/jasonjfoster/rolloptim

RemoteRef HEAD

RemoteSha 1c4c08bcde350c4d6c8ad51a35d551e40dd10ff7

Contents

`rolloptim-package` *Rolling Optimizations*

Description

Analytical computation of rolling optimizations for time-series data.

Details

`rolloptim` is a package that provides analytical computation of rolling optimization for time-series data.

Author(s)

Jason Foster

References

Markowitz, H.M. (1952). "Portfolio Selection." *The Journal of Finance*, 7(1), 77–91.

Tam, A. (2021). "Lagrangians and Portfolio Optimization." *https://www.adrian.idv.hk/2021-06-22-kkt/*.

`roll_max_mean` *Rolling Optimizations to Maximize Mean*

Description

A function for computing rolling optimizations to maximize mean.

Usage

```
roll_max_mean(mu, total = 1, lower = 0, upper = 1)
```

Arguments

`mu`	matrix. Rows are means and columns are variables.
`total`	numeric. Sum of the weights.
`lower`	numeric. Lower bound of the weights.
`upper`	numeric. Upper bound of the weights.

Value

An object of the same class and dimension as `mu` with the rolling optimizations to maximize mean.

Examples

```
if (requireNamespace("roll", quietly = TRUE)) {

n_vars <- 3
n_obs <- 15
x <- matrix(rnorm(n_obs * n_vars), nrow = n_obs, ncol = n_vars)

mu <- roll::roll_mean(x, 5)

# rolling optimizations to maximize mean
roll_max_mean(mu)

}
```

`roll_max_utility` *Rolling Optimizations to Maximize Utility*

Description

A function for computing rolling optimizations to maximize utility.

Usage

```
roll_max_utility(mu, sigma, lambda = 1, total = 1, lower = 0,
  upper = 1)
```

Arguments

`mu`	matrix. Rows are means and columns are variables.
`sigma`	cube. Slices are covariance matrices.
`lambda`	numeric. Risk aversion parameter.
`total`	numeric. Sum of the weights.
`lower`	numeric. Lower bound of the weights.
`upper`	numeric. Upper bound of the weights.

Value

An object of the same class and dimension as `mu` with the rolling optimizations to maximize utility.

Examples

```
if (requireNamespace("roll", quietly = TRUE)) {

n_vars <- 3
n_obs <- 15
x <- matrix(rnorm(n_obs * n_vars), nrow = n_obs, ncol = n_vars)
mu <- roll::roll_mean(x, 5)
sigma <- roll::roll_cov(x, width = 5)

# rolling optimizations to maximize utility
roll_max_utility(mu, sigma, lambda = 1)

}
```

roll_min_rss *Rolling Optimizations to Minimize Residual Sum of Squares*

Description

A function for computing rolling optimizations to minimize residual sum of squares.

Usage

```
roll_min_rss(xx, xy, total = 1, lower = 0, upper = 1)
```

Arguments

`xx`	cube. Slices are crossproducts of `x` and `x`.
`xy`	cube. Slices are crossproducts of `x` and `y`.
`total`	numeric. Sum of the weights.
`lower`	numeric. Lower bound of the weights.
`upper`	numeric. Upper bound of the weights.

Value

An object of the same class and dimension as `x` with the rolling optimizations to minimize residual sum of squares.

Examples

```
if (requireNamespace("roll", quietly = TRUE)) {

n_vars <- 3
n_obs <- 15
x <- matrix(rnorm(n_obs * n_vars), nrow = n_obs, ncol = n_vars)
y <- rnorm(n_obs)

xx <- roll::roll_crossprod(x, x, 5)
xy <- roll::roll_crossprod(x, y, 5)

# rolling optimizations to minimize residual sum of squares
roll_min_rss(xx, xy)

}
```

`roll_min_var` *Rolling Optimizations to Minimize Variance*

Description

A function for computing rolling optimizations to minimize variance.

Usage

```
roll_min_var(sigma, total = 1, lower = 0, upper = 1)
```

Arguments

`sigma`	cube. Slices are covariance matrices.
`total`	numeric. Sum of the weights.
`lower`	numeric. Lower bound of the weights.
`upper`	numeric. Upper bound of the weights.

Value

An object of the same class and dimension as `mu` with the rolling optimizations to minimize variance.

Examples

```
if (requireNamespace("roll", quietly = TRUE)) {

n_vars <- 3
n_obs <- 15
x <- matrix(rnorm(n_obs * n_vars), nrow = n_obs, ncol = n_vars)

sigma <- roll::roll_cov(x, width = 5)

# rolling optimizations to minimize variance
roll_min_var(sigma)

}
```

Index