Package ‘PeerPerformance’

August 30, 2018

Version 2.2.1
Date 2018-08-29
Title Luck-Corrected Peer Performance Analysis in R
Author David Ardia [aut, cre],
    Kris Boudt [aut]
Maintainer David Ardia <david.ardia.ch@gmail.com>
Depends parallel, sandwich, lmtest, compiler
Suggests testthat
Description Provides functions to perform the peer performance
    analysis of funds’ returns as described in Ar-
BugReports https://github.com/ArdiaD/PeerPerformance/issues
URL https://github.com/ArdiaD/PeerPerformance
License GPL (>= 2)
RoxygenNote 6.1.0
NeedsCompilation no
Repository CRAN
Date/Publication 2018-08-30 16:50:03 UTC

R topics documented:

alphaScreening ........................................... 2
hfdata ..................................................... 4
msharpe .................................................. 5
msharpeScreening ........................................ 6
msharpeTesting .......................................... 8
PeerPerformance .......................................... 11
sharpe ..................................................... 12
sharpeScreening .......................................... 13
sharpeTesting ........................................... 16

Index 19
alphaScreening  

Screening using the alpha outperformance ratio

Description

Function which performs the screening of a universe of returns, and computes the alpha outperformance ratio.

Usage

alphaScreening(X, factors = NULL, control = list())

Arguments

- **X**: Matrix \((T \times N)\) of \(T\) returns for the \(N\) funds. NA values are allowed.
- **factors**: Matrix \((T \times K)\) of \(T\) returns for the \(K\) factors. NA values are allowed.
- **control**: Control parameters (see *Details*).

Details

The alpha measure (Treynor and Black 1973, Carhart 1997, Fung and Hsieh 2004) is one industry standard for measuring the absolute risk adjusted performance of hedge funds. We propose to complement the alpha measure with the fund’s alpha outperformance ratio, defined as the percentage number of funds that have a significantly lower alpha. In a pairwise testing framework, a fund can have a significantly higher alpha because of luck. We correct for this by applying the false discovery rate approach by Storey (2002).

The methodology proceeds as follows:

- (1) compute all pairwise tests of alpha differences. This means that for a universe of \(N\) funds, we perform \(N(N - 1)/2\) tests. The algorithm has been parallelized and the computational burden can be splitted across several cores. The number of cores can be defined in control, see below.
- (2) for each fund, the false discovery rate approach by Storey (2002) is used to determine the proportions of over, equal, and underperforming funds, in terms of alpha, in the database.

The argument control is a list that can supply any of the following components:

- 'hac' Heteroscedastic-autocorrelation consistent standard errors. Default: hac = FALSE.
- 'minObs' Minimum number of concordant observations to compute the ratios. Default: minObs = 10.
- 'minObsPi' Minimum number of observations for computing the p-values. Default: minObsPi = 1.
- 'nCore' Number of cores used to perform the screening. Default: nCore = 1.
- 'lambda' Threshold value to compute pi0. Default: lambda = NULL, i.e. data driven choice.
Value

A list with the following components:

- \text{n}: Vector (of length \(N\)) of number of non-NA observations.
- \text{npeer}: Vector (of length \(N\)) of number of available peers.
- \text{alpha}: Vector (of length \(N\)) of unconditional alpha.
- \text{dalpha}: Matrix (of size \(N \times N\)) of alpha differences.
- \text{tstat}: Matrix (of size \(N \times N\)) of t-statistics.
- \text{pval}: Matrix (of size \(N \times N\)) of p-values of test for alpha differences.
- \text{lambda}: Vector (of length \(N\)) of lambda values.
- \text{pizero}: Vector (of length \(N\)) of probability of equal performance.
- \text{pipos}: Vector (of length \(N\)) of probability of outperformance performance.
- \text{pineg}: Vector (of length \(N\)) of probability of underperformance performance.

Note

Further details on the methodology with an application to the hedge fund industry is given in Ardia and Boudt (2018).

Application of the false discovery rate approach applied to the mutual fund industry has been presented in Barras, Scaillet and Wermers (2010).

Currently, the HAC asymptotic and studentized circular block bootstrap presented in Ledoit and Wolf (2008) are not supported by the \text{alphaScreening} function.

Author(s)

David Ardia and Kris Boudt.

References


See Also

`sharpeScreening` and `msharpeScreening`.

Examples

```r
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
rets = hfdata[,1:10]

## Run alpha screening
ctr = list(nCore = 1)
alphascreening(rets, control = ctr)

## Run alpha screening with HAC standard deviation
ctr = list(nCore = 1, hac = TRUE)
alphascreening(rets, control = ctr)
```

---

**hfdata**

*Hedge fund data*

**Description**

The matrix `hfdata` contains randomized and modified monthly returns of hedge funds.

**Usage**

`data(hfdata)`

**Format**

A matrix of size 60 x 100, containing 60 monthly net returns for 100 hedge funds. Returns have been modified.

**Source**

HFR database
msharpe

Compute modified Sharpe ratio

Description
Function which computes the modified Sharpe ratio

Usage
msharpe(X, level = 0.9, na.rm = TRUE, na.neg = TRUE)

Arguments
- X: Vector (of length \(T\)) or matrix (of size \(T \times N\)) of returns. NA values are allowed.
- level: Modified Value-at-Risk level. Default: level = 0.90.
- na.rm: A logical value indicating whether NA values should be stripped before the computation. Default na.rm = TRUE.
- na.neg: A logical value indicating whether NA values should be returned if a negative modified Value-at-Risk is obtained. Default na.neg = TRUE.

Details
The modified Sharpe ratio (Favre and Galeano 2002) is one industry standard for measuring the absolute risk adjusted performance of hedge funds.

Value
Scalar or a vector (of size \(N\)) with the modified Sharpe ratios.

Author(s)
David Ardia and Kris Boudt.

References

See Also
msharpeTesting, msharpeScreening and sharpe.
Examples

```r
## Load the data (randomized data of monthly hedge fund returns)
data('hfdata')

out = msharpe(hfdata)
print(out)

out = msharpe(hfdata, na.rm = FALSE)
print(out)
```

---

**msharpeScreening**  
*Screening using the modified Sharpe outperformance ratio*

---

**Description**

Function which performs the screening of a universe of returns, and computes the modified Sharpe outperformance ratio.

**Usage**

```r
msharpeScreening(X, level = 0.9, na.neg = TRUE, control = list())
```

**Arguments**

- `X`  
  Matrix \( (T \times N) \) of \( T \) returns for the \( N \) funds. NA values are allowed.
- `level`  
  Modified Value-at-Risk level. Default: `level = 0.90`.
- `na.neg`  
  A logical value indicating whether NA values should be returned if a negative modified Value-at-Risk is obtained. Default `na.neg = TRUE`.
- `control`  
  Control parameters (see *Details*).

**Details**

The modified Sharpe ratio (Favre and Galeano 2002, Gregoriou and Gueyie 2003) is one industry standard for measuring the absolute risk adjusted performance of hedge funds. We propose to complement the modified Sharpe ratio with the fund’s outperformance ratio, defined as the percentage number of funds that have a significantly lower modified Sharpe ratio. In a pairwise testing framework, a fund can have a significantly higher modified Sharpe ratio because of luck. We correct for this by applying the false discovery rate approach by Storey (2002).

For the testing, only the intersection of non-NA observations for the two funds are used.

The argument `control` is a list that can supply any of the following components:

- `'type'` Asymptotic approach (`type = 1`) or studentized circular bootstrap approach (`type = 2`). Default: `type = 1`.
- `'tttype'` Test based on ratio (`type = 1`) or product (`type = 2`). Default: `type = 2`.
- `'hac'` heteroscedastic-autocorrelation consistent standard errors. Default: `hac = FALSE`.  

msharpeScreening

- 'nBoot' Number of bootstrap replications for computing the p-value. Default: \( n \text{Boot} = 499 \).
- 'bboot' Block length in the circular bootstrap. Default: \( b \text{Boot} = 1 \), i.e. iid bootstrap. \( b \text{Boot} = 0 \) uses optimal block-length.
- 'pBoot' Symmetric p-value \( (p \text{Boot} = 1) \) or asymmetric p-value \( (p \text{Boot} = 2) \). Default: \( p \text{Boot} = 1 \).
- 'nCore' Number of cores to be used. Default: \( n \text{Core} = 1 \).
- 'minObs' Minimum number of concordant observations to compute the ratios. Default: \( \text{minObs} = 10 \).
- 'minObsPi' Minimum number of observations to compute \( \pi_0 \). Default: \( \text{minObsPi} = 1 \).
- '\lambda' Threshold value to compute \( \pi_0 \). Default: \( \lambda = \text{null} \), i.e. data driven choice.

Value

A list with the following components:

- \( n \): Vector (of length \( N \)) of number of non-NA observations.
- \( n\text{peer} \): Vector (of length \( N \)) of number of available peers.
- \( m\text{sharpe} \): Vector (of length \( N \)) of unconditional modified Sharpe ratios.
- \( d\text{msharpe} \): Matrix (of size \( N \times N \)) of modified Sharpe ratios differences.
- \( t\text{stat} \): Matrix (of size \( N \times N \)) of t-statistics.
- \( p\text{val} \): Matrix (of size \( N \times N \)) of p-values of test for modified Sharpe ratios differences.
- \( \lambda \text{d} \): Vector (of length \( N \)) of \( \lambda \) values.
- \( \pi_0 \text{r} \): Vector (of length \( N \)) of probability of equal performance.
- \( \pi_0 \text{p} \): Vector (of length \( N \)) of probability of outperformance performance.
- \( \pi_0 \text{n} \): Vector (of length \( N \)) of probability of underperformance performance.

Note

Further details on the methodology with an application to the hedge fund industry is given in: Ardia and Boudt (2018).

Some internal functions were adapted from Wolf’s R code.

Application of the false discovery rate approach applied to the mutual fund industry has been presented in Barraz, Scaillet and Wermers (2010).
Author(s)

David Ardia and Kris Boudt.

References


See Also

`msharpe, msharpeTesting, sharpeScreening` and `alphaScreening`.

Examples

```r
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
rets = hfdata[,1:10]

## Modified Sharpe screening
msharpeScreening(rets, control = list(nCore = 1))

## Modified Sharpe screening with bootstrap and HAC standard deviation
msharpeScreening(rets, control = list(nCore = 1, type = 2, hac = TRUE))
```

Description

Function which performs the testing of the difference of modified Sharpe ratios.
Usage

msharpeTesting(x, y, level = 0.9, na.neg = TRUE, control = list())

Arguments

x  Vector (of length \( T \)) of returns for the first fund. NA values are allowed.
y  Vector (of length \( T \)) of returns for the second fund. NA values are allowed.
level  Modified Value-at-Risk level. Default: \( \text{level} = 0.90 \).
na.neg  A logical value indicating whether NA values should be returned if a negative modified Value-at-Risk is obtained. Default \( \text{na.neg} = \text{TRUE} \).
control  Control parameters (see *Details*).

Details

The modified Sharpe ratio (Favre and Galeano 2002) is one industry standard for measuring the absolute risk adjusted performance of hedge funds. This function performs the testing of modified Sharpe ratio difference for two funds using a similar approach than Ledoit and Wolf (2002). See also Gregoriou and Gueyie (2003).

For the testing, only the intersection of non-NA observations for the two funds are used.

The argument control is a list that can supply any of the following components:

- 'type' Asymptotic approach (type = 1) or studentized circular bootstrap approach (type = 2). Default: \( \text{type} = 1 \).
- 'ttype' Test based on ratio (type = 1) or product (type = 2). Default: \( \text{type} = 2 \).
- 'hac' Heteroscedastic-autocorrelation consistent standard errors. Default: \( \text{hac} = \text{FALSE} \).
- 'minObs' Minimum number of concordant observations to compute the ratios. Default: \( \text{minObs} = 10 \).
- 'nBoot' Number of bootstrap replications for computing the p-value. Default: \( \text{nBoot} = 499 \).
- 'bBoot' Block length in the circular bootstrap. Default: \( \text{bBoot} = 1 \), i.e. iid bootstrap. \( \text{bBoot} = 0 \) uses optimal block-length.
- 'pBoot' Symmetric p-value (\( \text{pBoot} = 1 \)) or asymmetric p-value (\( \text{pBoot} = 2 \)). Default: \( \text{pBoot} = 1 \).

Value

A list with the following components:

- \( n \): Number of non-NA concordant observations.
- msharpe: Vector (of length 2) of unconditional modified Sharpe ratios.
- dmsharpe: Modified Sharpe ratios difference.
- tstat: t-stat of modified Sharpe ratios differences.
- pval: pvalues of test of modified Sharpe ratios differences.
Note

Further details on the methodology with an application to the hedge fund industry is given in Ardia and Boudt (2018).

Some internal functions where adapted from Michael Wolf MATLAB code.

Author(s)

David Ardia and Kris Boudt.

References


See Also

msharpe, msharpeScreening and sharpeTesting.

Examples

```r
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
x = hfdata[,1]
y = hfdata[,2]

## Run modified Sharpe testing (asymptotic)
ctr = list(type = 1)
out = msharpeTesting(x, y, level = 0.95, control = ctr)
print(out)

## Run modified Sharpe testing (asymptotic hac)
ctr = list(type = 1, hac = TRUE)
out = msharpeTesting(x, y, level = 0.95, control = ctr)
print(out)
```
PeerPerformance

## PeerPerformance

PeerPerformance (Ardia and Boudt, 20xx) is an R package for the peer-performance evaluation of financial investments with luck-correction, useful in the financial industry. In particular, it implements the peer performance ratios of Ardia and Boudt (2018) which measure the percentage of peers a focal (hedge) fund outperforms and underperforms, after correction for luck. It is useful for fund or portfolio managers to benchmark their investments or screen a universe of new funds. In addition, the package implements the testing framework for the Sharpe and modified Sharpe ratios, described in Ledoit and Wolf (2008) and Ardia and Boudt (2015).

### Functions

- Sharpe ratio: `sharpe`, `sharpeTesting` and `sharpeScreening`;
- Modified Share ratio: `msharpe`, `msharpeTesting` and `msharpeScreening`;
- Screening function: `alphascreening`, `sharpeScreening` and `msharpeScreening`.

### Update

The latest version of the package is available at https://github.com/Arda/D/PeerPerformance

### Note

Full description of the methodologies implemented in the various functions is available in Ledoit and Wolf (2008) and Ardia and Boudt (2015, 2018).

Please cite the package in publications. Use citation("PeerPerformance").

### Author(s)

David Ardia and Kris Boudt.
References


---

**sharpe**

*Compute Sharpe ratio*

**Description**

Function which computes the Sharpe ratio.

**Usage**

`sharpe(X, na.rm = TRUE)`

**Arguments**

- `X` Vector (of length `T`) or matrix (of size `T × N`) of returns for `N` funds. NA values are allowed.
- `na.rm` A logical value indicating whether NA values should be stripped before the computation. Default `na.rm = TRUE`

**Details**

The Sharpe ratio (Sharpe 1992) is one industry standard for measuring the absolute risk adjusted performance of hedge funds.
sharpeScreening

Value
A scalar or a vector (of size $N$) with the Sharpe ratios.

Author(s)
David Ardia and Kris Boudt.

References

See Also
sharpeTesting, sharpeScreening and msharpe.

Examples
```r
## Load the data
data('hfdata')

## Compute the Sharpe ratio
e = sharpe(hfdata)
print(e)
e = sharpe(hfdata, na.rm = FALSE)
print(e)
```

Description
Function which performs the screening of a universe of returns, and computes the Sharpe outperformance ratio.

Usage
```
sharpeScreening(X, control = list())
```

Arguments
```
X          Matrix ($T \times N$) of $T$ returns for the $N$ funds. NA values are allowed.
control    Control parameters (see *Details*).
```
Details

The Sharpe ratio (Sharpe 1992) is one industry standard for measuring the absolute risk adjusted performance of hedge funds. We propose to complement the Sharpe ratio with the fund’s outperformance ratio, defined as the percentage number of funds that have a significantly lower Sharpe ratio. In a pairwise testing framework, a fund can have a significantly higher Sharpe ratio because of luck. We correct for this by applying the false discovery rate approach by Storey (2002).

For the testing, only the intersection of non-NA observations for the two funds are used.

The methodology proceeds as follows:

- (1) compute all pairwise tests of Sharpe differences using the bootstrap approach of Ledoit and Wolf (2002). This means that for a universe of \( N \) funds, we perform \( N(N-1)/2 \) tests. The algorithm has been parallelized and the computational burden can be split across several cores. The number of cores can be defined in control, see below.
- (2) for each fund, the false discovery rate approach by Storey (2002) is used to determine the proportions over, equal, and underperforming funds, in terms of Sharpe ratio, in the database.

The argument control is a list that can supply any of the following components:

- 'type' Asymptotic approach (type = 1) or studentized circular bootstrap approach (type = 2). Default: type = 1.
- 'ttype' Test based on ratio (type = 1) or product (type = 2). Default: type = 2.
- 'hac' Heteroscedastic-autocorrelation consistent standard errors. Default: hac = FALSE.
- 'nBoot' Number of bootstrap replications for computing the p-value. Default: nBoot = 499.
- 'bBoot' Block length in the circular bootstrap. Default: bBoot = 1, i.e. iid bootstrap. bBoot = 0 uses optimal block-length.
- 'pBoot' Symmetric p-value (pBoot = 1) or asymmetric p-value (pBoot = 2). Default: pBoot = 1.
- 'nCore' Number of cores to be used. Default: nCore = 1.
- 'minObs' Minimum number of concordant observations to compute the ratios. Default: minObs = 10.
- 'minObsPi' Minimum number of observations to compute pi0. Default: minObsPi = 1.
- 'lambda' Threshold value to compute pi0. Default: lambda = NULL, i.e. data driven choice.

Value

A list with the following components:

\( n \): Vector (of length \( N \)) of number of non-NA observations.

\( npeer \): Vector (of length \( N \)) of number of available peers.

\( sharpe \): Vector (of length \( N \)) of unconditional Sharpe ratios.

\( dsharpe \): Matrix (of size \( N \times N \)) of Sharpe ratios differences.
tstat: Matrix (of size $N \times N$) of t-statistics.

pval: Matrix (of size $N \times N$) of pvalues of test for Sharpe ratios differences.

lambda: vector (of length $N$) of lambda values.

pizero: vector (of length $N$) of probability of equal performance.

pipos: vector (of length $N$) of probability of outperformance performance.

pineg: Vector (of length $N$) of probability of underperformance performance.

Note

Further details on the methodology with an application to the hedge fund industry is given in Ardia and Boudt (2018).

Some internal functions where adapted from Michael Wolf MATLAB code.

Application of the false discovery rate approach applied to the mutual fund industry has been presented in Barras, Scaillet and Wermers (2010).

Author(s)

David Ardia and Kris Boudt.

References


See Also

sharpe, sharpeTesting, msharpeScreening and alphaScreening.
Examples

```r
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
rets = hfdata[,1:10]

## Sharpe screening
sharpeScreening(rets, control = list(nCore = 1))

## Sharpe screening with bootstrap and HAC standard deviation
sharpeScreening(rets, control = list(nCore = 1, type = 2, hac = TRUE))
```

---

**sharpeTesting**  
*Testing the difference of Sharpe ratios*

Description

Function which performs the testing of the difference of Sharpe ratios.

Usage

```r
sharpeTesting(x, y, control = list())
```

Arguments

- **x**: Vector (of length \( T \)) of returns for the first fund. NA values are allowed.
- **y**: Vector (of length \( T \)) of returns for the second fund. NA values are allowed.
- **control**: Control parameters (see *Details*).

Details

The Sharpe ratio (Sharpe 1992) is one industry standard for measuring the absolute risk adjusted performance of hedge funds. This function performs the testing of Sharpe ratio difference for two funds using the approach by Ledoit and Wolf (2002).

For the testing, only the intersection of non-NA observations for the two funds are used.

The argument control is a list that can supply any of the following components:

- `'type'`: Asymptotic approach (\( \text{type} = 1 \)) or studentized circular bootstrap approach (\( \text{type} = 2 \)). Default: \( \text{type} = 1 \).
- `'tttype'`: Test based on ratio (\( \text{type} = 1 \)) or product (\( \text{type} = 2 \)). Default: \( \text{type} = 2 \).
- `'hac'`: Heteroscedastic-autocorrelation consistent standard errors. Default: \( \text{hac} = \text{FALSE} \).
- `'nBoot'`: Number of bootstrap replications for computing the p-value. Default: \( \text{nBoot} = 499 \).
- `'bBoot'`: Block length in the circular bootstrap. Default: \( \text{bBoot} = 1 \), i.e. iid bootstrap. \( \text{bBoot} = 0 \) uses optimal block-length.
- `'pBoot'`: Symmetric p-value (\( \text{pBoot} = 1 \)) or asymmetric p-value (\( \text{pBoot} = 2 \)). Default: \( \text{pBoot} = 1 \).
Value

A list with the following components:

- n: Number of non-NA concordant observations.
- sharpe: Vector (of length 2) of unconditional Sharpe ratios.
- dsharpe: Sharpe ratios difference.
- tstat: t-stat of Sharpe ratios differences.
- pval: pvalues of test of Sharpe ratios differences.

Note

Further details on the methodology with an application to the hedge fund industry is given in Ardia and Boudt (2018).

Some internal functions were adapted from Michael Wolf MATLAB code.

Author(s)

David Ardia and Kris Boudt.

References


See Also

- `sharpe`
- `sharpeScreening`
- `msharpeTesting`
Examples

```r
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
x = hfdata[,1]
y = hfdata[,2]

## Run Sharpe testing (asymptotic)
ctr = list(type = 1)
out = sharpeTesting(x, y, control = ctr)
print(out)

## Run Sharpe testing (asymptotic hac)
ctr = list(type = 1, hac = TRUE)
out = sharpeTesting(x, y, control = ctr)
print(out)

## Run Sharpe testing (iid bootstrap)
set.seed(1234)
ctr = list(type = 2, nBoot = 250)
out = sharpeTesting(x, y, control = ctr)
print(out)

## Run Sharpe testing (circular bootstrap)
set.seed(1234)
ctr = list(type = 2, nBoot = 250, bBoot = 5)
out = sharpeTesting(x, y, control = ctr)
print(out)
```
Index

*Topic **datasets**
  hfdata, 4

*Topic **htest**
  alphaScreening, 2
  msharpe, 5
  msharpeScreening, 6
  msharpeTesting, 8
  sharpe, 12
  sharpeScreening, 13
  sharpeTesting, 16

alphaScreening, 2, 8, 11, 15

hfdata, 4

msharpe, 5, 8, 10, 11, 13
msharpeScreening, 4, 5, 6, 10, 11, 15
msharpeTesting, 5, 8, 8, 11, 17

PeerPerformance, 11
PeerPerformance-package
  (PeerPerformance), 11

sharpe, 5, 11, 12, 15, 17
sharpeScreening, 4, 8, 11, 13, 13, 17
sharpeTesting, 10, 11, 13, 15, 16