Package ‘PeerPerformance’

August 2, 2026

Version 2.4.0

Title Luck-Corrected Peer Performance Analysis in R

Depends R (>= 3.5.0)

Imports compiler, graphics, grDevices, lmtest, parallel, sandwich, stats

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Description Provides functions to perform the peer performance analysis of funds' returns as described in Ardia and Boudt (2018) <doi:10.1016/j.jbankfin.2017.10.014>. For each fund, the package estimates the proportion of peers it outperforms, is equalled by, and is outperformed by, correcting for luck with the false discovery approach of Storey (2002) <doi:10.1111/1467-9868.00346>. Screenings can be based on factor-model alphas, Sharpe ratios, or modified Sharpe ratios, the latter using the equality test of Ardia and Boudt (2015) <doi:10.1016/j.frl.2015.02.008>. Funds can be screened within a universe or against a separate peer group, over rolling windows, and results come with bootstrap confidence intervals, summary, plot, and tidy data frame methods.

BugReports `https://github.com/ArdiaD/PeerPerformance/issues`

URL `https://github.com/ArdiaD/PeerPerformance`

License GPL (>= 2)

LazyData true

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author David Ardia [aut, cre, cph, fnd] (ORCID: <`https://orcid.org/0000-0003-2823-782X`>),
Nabil Bouamara [ctb],
Kris Boudt [aut] (ORCID: <`https://orcid.org/0000-0002-1000-5142`>),
Sebastien Legros [ctb],
Murilo Andre Peres Pereira [ctb] (ORCID:

<https://orcid.org/0009-0007-0086-4801>),
Benjamin Seguin [ctb]

Maintainer David Ardia <david.ardia.ch@gmail.com>

Repository CRAN

Date/Publication 2026-08-02 20:50:02 UTC

Contents

`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(),
  screen_beta = NULL,
  Y = NULL
)
```

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*).
`screen_beta`	Boolean to screen all factors' coefficients (beta). Default: `screen_beta = NULL`, in which case the value is taken from `control$screen_beta` (itself defaulting to `FALSE`, i.e. only the alpha is screened). When supplied directly, the argument takes precedence over the control list. If `TRUE`, each element of the returned list will have a new first dimension representing each coefficient (the first one being alpha).
`Y`	Optional matrix $(T \times M)$ of returns for a second (peer) group of M funds. When supplied, the ratios are computed for each fund in `X` *against the funds in* `Y` (cross-group screening) instead of against the other funds in `X`. A single focal fund versus a peer group corresponds to `X` being a vector (or a one-column matrix). Columns of `Y` identical to the focal fund (e.g. when `X` is a subset of `Y`) are automatically excluded. Default: `Y = NULL`, i.e. within-group screening.

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 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 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.
- 'gammaPos' One-sided quantile level (of the standard Normal distribution) used as the critical value for counting outperformed peers: a peer counts as outperformed when the pairwise t-statistic exceeds qnorm(gammaPos) (a *negative* threshold for gammaPos < 0.5; e.g., qnorm(0.4) is about -0.25), and the expected fraction 1 - gammaPos of false positives among the equal-performing peers is then subtracted. Smaller values count more peers before the correction. Default: gammaPos = 0.4 (the value recommended in Ardia and Boudt, 2018).
- 'gammaNeg' Mirror image of gammaPos for the peers that outperform the focal fund: the count uses tstat <= qnorm(gammaNeg) and subtracts the expected fraction gammaNeg of false positives. Default: gammaNeg = 0.6.
- 'fastAdjust' Use a fast vectorised inversion in the truncated-normal bias correction of π^0 instead of one uniroot call per value. This is the dominant cost when lambda is data driven and gives a large speed-up on big universes. The bisection locates the root to about 1e-12; since uniroot stops at its own tolerance (about 1.2e-4), the two paths typically differ by a few 1e-5, the fast path being the more accurate. Default: fastAdjust = FALSE, i.e. the original code path, kept as default so that published results reproduce exactly.
- 'screen_beta' Screen the factor exposures (betas) in addition to the alpha; see the screen_beta argument. Default: screen_beta = FALSE.

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.

alpha: Vector (of length N) of unconditional alpha.

dalpha: Matrix (of size $N \times N$) of alpha differences.

tstat: Matrix (of size $N \times N$) of t-statistics.

pval: Matrix (of size $N \times N$) of p-values of test for alpha 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).

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

HAC standard errors are available via `control = list(hac = TRUE)` (computed with **sandwich/lmtest**). The studentized circular block bootstrap of Ledoit and Wolf (2008) applies to the Sharpe-ratio routines (`sharpeScreening`, `msharpeScreening`) and is not used by `alphaScreening`.

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, pp.97–104. doi:10.1016/j.frl.2015.02.008

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, pp.351–368. doi:10.1016/j.jbankfin.2017.10.014

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), pp.179–216.

Carhart, M. (1997). On persistence in mutual fund performance. *Journal of Finance* **52**(1), pp.57–82.

Fama, E., French, K. (2010). Luck versus skill in the cross-section of mutual fund returns. *Journal of Finance* **65**(5), pp.1915–1947.

Fung, W., Hsieh, D. (2004). Hedge fund benchmarks: A risk based approach. *Financial Analysts Journal* **60**(5), pp.65–80.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), pp.479–498.

Treynor, J. L., Black, F. (1973). How to use security analysis to improve portfolio selection. *Journal of Business* **46**(1), pp.66–86.

See Also

`sharpeScreening` and `msharpeScreening`.

Examples

```
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
rets = hfdata[,1:4]
```

```
## 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)

## Cross-group screening: a single focal fund against a peer group
alphaScreening(hfdata[, 1], Y = hfdata[, 11:20], control = list(nCore = 1))

## Cross-group screening: peer group X against peer group Y
alphaScreening(hfdata[, 1:5], Y = hfdata[, 11:20], control = list(nCore = 1))
```

`alphaTesting` *Testing the difference of alpha outperformance ratios*

Description

Function which performs the testing of the difference of alpha outperformance ratios.

Usage

```
alphaTesting(x, y, factors = NULL, control = list(), screen_beta = NULL)
```

Arguments

`x`	Vector (of length T) of returns for the first fund. `NA` values are allowed.
`y`	Vector (of length T) returns for the second fund. `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*).
`screen_beta`	Boolean to screen all factors' coefficients (beta). Default: `screen_beta = NULL`, in which case the value is taken from `control$screen_beta` (itself defaulting to `FALSE`). When supplied directly, the argument takes precedence. `screen_beta = TRUE` requires `factors`.

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. This function performs the testing of alpha outperformance ratio difference for two funds.

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:

- `'hac'` Heteroscedastic-autocorrelation consistent standard errors. Default: `hac = FALSE`.

Value

A list with the following components:

n: Number of non-NA concordant observations.

alpha: Vector (of length 2) of the two funds' unconditional alphas.

dalpha: Difference of the two funds' alphas.

tstat: t-stat of the alpha difference.

pval: p-value of the test of equal alpha.

When screen_beta = TRUE, alpha is a matrix (one row per coefficient, the first being the alpha) and dalpha, tstat and pval are vectors with one entry per coefficient; the print method reports the alpha (first) coefficient only.

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

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, 97–104.

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, 351–368.

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), 179–216.

Sharpe, W.F. (1994). The Sharpe ratio. *Journal of Portfolio Management* **21**(1), 49–58.

Ledoit, O., Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance* **15**(5), 850–859.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), 479–498.

See Also

alphaScreening.

Examples

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

## Run alpha testing
alphaTesting(x, y)
```

---

as.data.frame.SCREENING

Coerce a 'SCREENING' object to a data frame

Description

Returns a tidy `data.frame` with one row per fund (or, when the screening was run with `screen_beta = TRUE`, one row per fund/coefficient), suitable for downstream filtering, ranking, and plotting.

Usage

```
## S3 method for class 'SCREENING'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A `SCREENING` object.
`row.names, optional`	Currently ignored.
`...`	Further arguments (ignored).

Value

A `data.frame` with columns `fund`, the performance measure (`alpha`, `sharpe`, or `msharpe`), `pipos`, `pizero`, `pineg`, `npeer`, and `lambda`; a `coefficient` column is added when `screen_beta = TRUE`.

Author(s)

David Ardia and Kris Boudt.

See Also

`alphaScreening`.

Examples

```
data("hfdata")
sc <- alphaScreening(hfdata[, 1:10], control = list(nCore = 1))
head(as.data.frame(sc))
```

`confint.SCREENING` *Bootstrap confidence intervals for the peer performance ratios*

Description

Computes confidence intervals for one of the peer performance ratios ($\hat{\pi}^+$, $\hat{\pi}^0$, or $\hat{\pi}^-$) of a `SCREENING` object, by a nonparametric peer (pairwise) bootstrap: for each focal fund its peers are resampled with replacement and the ratio is recomputed from the stored pairwise p-values and test statistics. The interval therefore reflects the peer-composition and aggregation uncertainty, holding the individual pairwise test statistics fixed.

Usage

```
## S3 method for class 'SCREENING'
confint(
  object,
  parm = c("pipos", "pizero", "pineg"),
  level = 0.95,
  ...,
  nBoot = 499L,
  gammaPos = 0.4,
  gammaNeg = 0.6
)
```

Arguments

`object`	A `SCREENING` object (within-group or cross-group; not `screen_beta`).
`parm`	Which ratio to bound: `"pipos"` (default), `"pizero"`, or `"pineg"`.
`level`	Confidence level. Default: `0.95`.
`...`	Unused.
`nBoot`	Number of bootstrap resamples. Default: `499`.
`gammaPos, gammaNeg`	One-sided thresholds used to recompute the ratios (should match those of the original screening). Defaults: `0.4`, `0.6`.

Value

A matrix with one row per fund and two columns (lower and upper bounds); the point estimate is attached as attribute `"estimate"`.

Author(s)

David Ardia and Kris Boudt.

See Also

`alphaScreening`.

Examples

```
data("hfdata")
set.seed(1234)
sc <- alphaScreening(hfdata[, 1:15], control = list(nCore = 1))
confint(sc, parm = "pipos")
```

`exposureHeterogeneity` *Factor exposure heterogeneity from a beta screening*

Description

Aggregates the per-fund equal-performance ratios of a `screen_beta = TRUE` `alphaScreening` into the factor-by-factor heterogeneity measure of Ardia et al. (2023). For each coefficient k (the alpha and each factor beta) it reports the average equal-exposure ratio $\pi_k^0 = \frac{1}{N} \sum_i \pi_{i,k}^0$ and the heterogeneity $1 - \pi_k^0$: the share of peers that are significantly differentiated on coefficient k. A value close to one indicates large heterogeneity (much room to differentiate); close to zero, homogeneity.

Usage

```
exposureHeterogeneity(object)
```

Arguments

`object`	A `SCREENING` object produced with `screen_beta = TRUE`.

Value

A `data.frame` of class `exposureHeterogeneity` with columns `coefficient`, `equalExposure` (π_k^0), and `heterogeneity` ($1 - \pi_k^0$).

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Bluteau, K., Lortie-Cloutier, G., Tran, D. (2023). Factor exposure heterogeneity in green and brown stocks. *Finance Research Letters* **55**, Part A, 103900.

See Also

`alphaScreening`.

Examples

```
data("hfdata")
set.seed(1234)
fac <- matrix(rnorm(nrow(hfdata) * 2), ncol = 2,
              dimnames = list(NULL, c("MKT", "SMB")))
sc <- alphaScreening(hfdata[, 1:20], factors = fac, screen_beta = TRUE,
                     control = list(nCore = 1))
exposureHeterogeneity(sc)
```

`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

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, 97–104.

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge Funds. *Journal of Banking and Finance* **87**, 351–368.

Favre, L., Galeano, J.A. (2002). Mean-modified Value-at-Risk Optimization with Hedge Funds. *Journal of Alternative Investments* **5**(2), 21–25.

Gregoriou, G. N., Gueyie, J.-P. (2003). Risk-adjusted performance of funds of hedge funds using a modified Sharpe ratio. *Journal of Wealth Management* **6**(3), 77–83.

See Also

`msharpeTesting`, `msharpeScreening` and `sharpe`.

Examples

```
## 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

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

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*).
`Y`	Optional matrix $(T \times M)$ of returns for a second (peer) group of M funds. When supplied, the ratios are computed for each fund in `X` against the funds in `Y` (cross-group screening) instead of against the other funds in `X`; a single focal fund versus a peer group corresponds to `X` being a vector. Columns of `Y` identical to the focal fund are automatically excluded. Default: `Y = NULL`, i.e. within-group screening.

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`.
- `'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. (The data-driven choice `bBoot = 0` is only available in `msharpeTesting`, not in screening.)

- '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.
- 'gammaPos' One-sided quantile level (of the standard Normal distribution) used as the critical value for counting outperformed peers: a peer counts as outperformed when the pairwise t-statistic exceeds qnorm(gammaPos) (a *negative* threshold for gammaPos < 0.5), and the expected fraction 1 - gammaPos of false positives among the equal-performing peers is then subtracted. Default: gammaPos = 0.4 (the value recommended in Ardia and Boudt, 2018).
- 'gammaNeg' Mirror image of gammaPos for the peers that outperform the focal fund: the count uses tstat <= qnorm(gammaNeg) and subtracts the expected fraction gammaNeg of false positives. Default: gammaNeg = 0.6.
- 'fastAdjust' Use a fast vectorised inversion in the truncated-normal bias correction of π^0 instead of one uniroot call per value. This is the dominant cost when lambda is data driven and gives a large speed-up on big universes. The bisection locates the root to about 1e-12; since uniroot stops at its own tolerance (about 1.2e-4), the two paths typically differ by a few 1e-5, the fast path being the more accurate. Default: fastAdjust = FALSE, i.e. the original code path, kept as default so that published results reproduce exactly.

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.

msharpe: Vector (of length N) of unconditional modified Sharpe ratios.

dmsharpe: Matrix (of size $N \times N$) of modified Sharpe ratios differences.

tstat: Matrix (of size $N \times N$) of t-statistics.

pval: Matrix (of size $N \times N$) of p-values of test for modified 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

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, 97–104.

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, 351–368.

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), 179–216.

Favre, L., Galeano, J.A. (2002). Mean-modified Value-at-Risk Optimization with Hedge Funds. *Journal of Alternative Investments* **5**(2), 21–25.

Gregoriou, G. N., Gueyie, J.-P. (2003). Risk-adjusted performance of funds of hedge funds using a modified Sharpe ratio. *Journal of Wealth Management* **6**(3), 77–83.

Ledoit, O., Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance* **15**(5), 850–859.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), 479–498.

See Also

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

Examples

```
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
rets = hfdata[,1:4]

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

## Modified Sharpe screening with the studentized circular bootstrap
## (the 'hac' flag is ignored when type = 2)
msharpeScreening(rets, control = list(nCore = 1, type = 2))
```

`msharpeTesting` *Testing the difference of modified Sharpe ratios*

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: `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) 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: `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`.
- `'minObs'` Minimum number of concordant observations to compute the ratios. Default: `minObs = 10`.
- `'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`.

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

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, pp.97–104. doi:10.1016/j.frl.2015.02.008

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, pp.351–368. doi:10.1016/j.jbankfin.2017.10.014

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), pp.179–216.

Favre, L., Galeano, J.A. (2002). Mean-modified Value-at-Risk Optimization with Hedge Funds. *Journal of Alternative Investments* **5**(2), pp.21–25.

Gregoriou, G. N., Gueyie, J.-P. (2003). Risk-adjusted performance of funds of hedge funds using a modified Sharpe ratio. *Journal of Wealth Management* **6**(3), pp.77–83.

Ledoit, O., Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance* **15**(5), pp.850–859.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), pp.479–498.

See Also

`msharpe`, `msharpeScreening` and `sharpeTesting`.

Examples

```
## 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)

## Run modified Sharpe testing (iid bootstrap)
set.seed(1234)
ctr = list(type = 2, nBoot = 250)
out = msharpeTesting(x, y, level = 0.95, control = ctr)
print(out)

## Run modified Sharpe testing (circular bootstrap)
set.seed(1234)
ctr = list(type = 2, nBoot = 250, bBoot = 5)
out = msharpeTesting(x, y, level = 0.95, control = ctr)
print(out)
```

`PeerPerformance` *PeerPerformance: Luck-corrected peer performance analysis in R*

Description

`PeerPerformance` 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). See also Ardia et al. (2022,2023) for applications in sustainable finance.

Functions

- Alpha: `alphaTesting` and `alphaScreening`;
- Sharpe ratio: `sharpe`, `sharpeTesting` and `sharpeScreening`;
- Modified Sharpe ratio: `msharpe`, `msharpeTesting` and `msharpeScreening`;
- Screening: `alphaScreening`, `sharpeScreening`, and `msharpeScreening`.

Update

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

Note

By using `PeerPerformance` you agree to the following rules: (1) You must cite Ardia and Boudt (2018) in working papers and published papers that use `PeerPerformance` (use `citation("PeerPerformance")`), (2) you must place the URL https://CRAN.R-project.org/package=PeerPerformance in a footnote to help others find `PeerPerformance`, and (3) you assume all risk for the use of `PeerPerformance`.

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

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, 97–104.

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge Funds. *Journal of Banking and Finance* **87**, 351–368.

Ardia, D., Bluteau, K., Tran, D. (2022). How easy is it for investment managers to deploy their talent in green and brown stocks? *Finance Research Letters* **48**, 102992.

Ardia, D., Bluteau, K., Lortie-Cloutier, G., Tran, D. (2023). Factor exposure heterogeneity in green and brown stocks. *Finance Research Letters* **55**, Part A, 103900.

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), 179–216.

Favre, L., Galeano, J.A. (2002). Mean-modified Value-at-Risk Optimization with Hedge Funds. *Journal of Alternative Investments* **5**(2), 21–25.

Gregoriou, G. N., Gueyie, J.-P. (2003). Risk-adjusted performance of funds of hedge funds using a modified Sharpe ratio. *Journal of Wealth Management* **6**(3), 77–83.

Ledoit, O., Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance* **15**(5), 850–859.

Sharpe, W.F. (1994). The Sharpe ratio. *Journal of Portfolio Management* **21**(1), 49–58.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), 479–498.

plot.exposureHeterogeneity

Plot factor exposure heterogeneity

Description

Bar plot of the heterogeneity measure $1 - \pi_k^0$ returned by `exposureHeterogeneity`.

Usage

```
## S3 method for class 'exposureHeterogeneity'
plot(x, ...)
```

Arguments

`x`	An `exposureHeterogeneity` object.
`...`	Further arguments passed to `barplot`.

Value

Invisibly returns `x`.

Author(s)

David Ardia and Kris Boudt.

See Also

`exposureHeterogeneity`.

Examples

```
data("hfdata")
sc <- alphaScreening(hfdata[, 1:10], factors = hfdata[, 50:51],
                     control = list(nCore = 1, screen_beta = TRUE))
plot(exposureHeterogeneity(sc))
```

`plot.rollScreening` *Plot a rolling peer performance screening*

Description

Time-series plot of the rolling peer performance ratios produced by `rollScreening`.

Usage

```
## S3 method for class 'rollScreening'
plot(x, what = NULL, legend = TRUE, ...)
```

Arguments

`x`	A `rollScreening` object.
`what`	Either `"heterogeneity"` ($1-\bar{\pi}^0$) or `"ratios"` (the averaged $\bar{\pi}^+, \bar{\pi}^0, \bar{\pi}^-$). Default: `"heterogeneity"` for a beta screening, `"ratios"` otherwise.
`legend`	A logical value indicating whether to draw a legend. Default: `TRUE`.
`...`	Further graphical arguments.

Value

Invisibly returns `x`.

Author(s)

David Ardia and Kris Boudt.

See Also

`rollScreening`.

Examples

```
data("hfdata")
roll <- rollScreening(hfdata[, 1:15], width = 36, by = 12,
                      control = list(nCore = 1))
plot(roll)
```

plot.SCREENING *Peer performance screening plot*

Description

Plot method for the SCREENING object returned by alphaScreening, sharpeScreening and msharpeScreening. It reproduces the peer performance screening plot of Ardia and Boudt (2018): funds are sorted by their performance measure and, for each fund, the estimated outperformance ($\hat{\pi}^+$), equal-performance ($\hat{\pi}^0$) and underperformance ($\hat{\pi}^-$) ratios are displayed as a horizontal stacked bar. The dashed diagonal lines depict the naive percentile-rank benchmark ($\hat{\pi}^0 = 0$); the gap between the black $\hat{\pi}^+$ area and the diagonal visualizes the luck correction.

Usage

```
## S3 method for class 'SCREENING'
plot(
  x,
  nblock = NULL,
  reference = NULL,
  band = 27.5,
  colorset = c(grDevices::gray(0), grDevices::gray(0.8), grDevices::gray(0.5)),
  ...
)
```

Arguments

x	A SCREENING object.
nblock	Optional number of equally-sized blocks into which the funds (sorted by performance) are aggregated by averaging, as in Ardia and Boudt (2018). Default: nblock = NULL, i.e. one bar per fund.
reference	A logical value indicating whether the dashed percentile-rank reference lines should be drawn. Default: reference = NULL, i.e. drawn for within-group screening and omitted for cross-group (Y-based) screening, where the within-group percentile-rank benchmark does not apply. A single focal fund (cross-group with one fund in X) is shown as a single stacked bar.
band	Half-width (in percentage points) of the reference band drawn around the central diagonal. Default: band = 27.5.
colorset	Vector of three colors for the $\hat{\pi}^+$, $\hat{\pi}^0$ and $\hat{\pi}^-$ areas. Default: c(gray(0), gray(0.8), gray(0.5)).
...	Further graphical arguments passed to barplot.

Details

If the screening was run with screen_beta = TRUE, the alpha (first) coefficient is used.

Value

Invisibly returns the (sorted, possibly aggregated) matrix of ratios that is plotted.

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, pp.351–368. doi:10.1016/j.jbankfin.2017.10.014

See Also

`alphaScreening`, `sharpeScreening` and `msharpeScreening`.

Examples

```
data("hfdata")
set.seed(1234)
sc <- alphaScreening(hfdata[, 1:30], control = list(nCore = 1))
plot(sc)
```

`print.SCREENING` *Print method for the 'SCREENING' object*

Description

Print method for the `SCREENING` object returned by `alphaScreening`, `sharpeScreening` and `msharpeScreening`.

Usage

```
## S3 method for class 'SCREENING'
print(x, ...)
```

Arguments

`x`	A `SCREENING` object.
`...`	Further arguments passed to or from other methods.

Value

Invisibly returns `x`.

Author(s)

David Ardia and Kris Boudt.

See Also

alphaScreening, sharpeScreening, msharpeScreening and plot.SCREENING.

---

print.summary.SCREENING *Print method for the 'summary.SCREENING' object*

Description

Print method for the object returned by summary.SCREENING.

Usage

```
## S3 method for class 'summary.SCREENING'
print(x, ...)
```

Arguments

x	A summary.SCREENING object.
...	Further arguments (ignored).

Value

Invisibly returns x.

Author(s)

Murilo Andre Peres Pereira, David Ardia and Kris Boudt.

---

print.TESTING *Print method for the 'TESTING' object*

Description

Print method for the TESTING object returned by sharpeTesting, msharpeTesting and alphaTesting.

Usage

```
## S3 method for class 'TESTING'
print(x, ...)
```

Arguments

x	A TESTING object.
...	Further arguments passed to or from other methods.

Value

Invisibly returns `x`.

Author(s)

David Ardia and Kris Boudt.

See Also

`sharpeTesting`, `msharpeTesting` and `alphaTesting`.

`rollScreening` *Rolling-window peer performance screening*

Description

Runs a peer performance screening on a rolling time window and returns the time series of cross-sectionally averaged peer performance ratios. This is the design underlying the dynamic analyses of Ardia et al. (2022, 2023): on each window the screening is computed and the ratios are averaged across funds (per coefficient when `screen_beta = TRUE`), yielding a series of (factor exposure) heterogeneity measures.

Usage

```
rollScreening(
  X,
  factors = NULL,
  Y = NULL,
  screen = c("alpha", "sharpe", "msharpe"),
  width = 36L,
  by = 1L,
  level = 0.9,
  na.neg = TRUE,
  control = list(),
  dates = NULL
)
```

Arguments

`X`	Matrix $(T \times N)$ of T returns for the N funds.
`factors`	Matrix $(T \times K)$ of factor returns (for `screen = "alpha"`). Default: `NULL`.
`Y`	Optional matrix $(T \times M)$ of returns for a peer group (cross-group screening, see `alphaScreening`). Default: `NULL`.
`screen`	Performance measure to screen on: `"alpha"` (default), `"sharpe"`, or `"msharpe"`.
`width`	Window length (number of observations). Default: `36`.

by	Step (in observations) between successive windows. Default: 1.
level	Modified Value-at-Risk level (for screen = "msharpe"). Default: 0.9.
na.neg	A logical value passed to msharpeScreening (for screen = "msharpe") indicating whether a negative modified Value-at-Risk yields NA. Default: TRUE.
control	Control parameters passed to the screening function (see alphaScreening); set control = list(screen_beta = TRUE) to obtain per-factor heterogeneity series. With nCore > 1 each window spins up its own PSOCK cluster, which is wasteful for many small windows; the default nCore = 1 runs serially without any cluster.
dates	Optional vector of length T (e.g. Date) used to label the windows by their end date. Default: NULL.

Value

A data.frame of class rollScreening with one row per window (or per window/coefficient when screen_beta = TRUE), with columns window, index (end-of-window row), optionally date, coefficient, the averaged pizero/pipos/ pineg, and heterogeneity ($1 - \bar{\pi}^0$).

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Bluteau, K., Tran, D. (2022). How easy is it for investment managers to deploy their talent in green and brown stocks? *Finance Research Letters* **48**, 102992.

Ardia, D., Bluteau, K., Lortie-Cloutier, G., Tran, D. (2023). Factor exposure heterogeneity in green and brown stocks. *Finance Research Letters* **55**, Part A, 103900.

See Also

alphaScreening, exposureHeterogeneity.

Examples

```
data("hfdata")
set.seed(1234)
## a subset keeps the example quick; drop the column index to screen the
## whole universe
roll <- rollScreening(hfdata[, 1:20], screen = "alpha", width = 36, by = 6,
                      control = list(nCore = 1))
plot(roll)
```

`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 $T \times N$) 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 1994) is one industry standard for measuring the absolute risk adjusted performance of hedge funds.

Value

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

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, pp.97–104. doi:10.1016/j.frl.2015.02.008

Ardia, D., Boudt, K. (2016). The Peer Ratios Performance of Hedge Funds. *Working paper*. doi:10.2139/ssrn.2000901

Sharpe, W.F. (1994). The Sharpe ratio. *Journal of Portfolio Management* **21**(1), pp.49–58. doi:10.3905/jpm.1994.409501

See Also

`sharpeTesting`, `sharpeScreening` and `msharpe`.

Examples

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

## Compute the Sharpe ratio
out = sharpe(hfdata)
print(out)

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

`sharpeScreening` *Screening using the Sharpe outperformance ratio*

Description

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

Usage

```
sharpeScreening(X, control = list(), Y = NULL)
```

Arguments

`X`	Matrix $(T \times N)$ of T returns for the N funds. `NA` values are allowed.
`control`	Control parameters (see *Details*).
`Y`	Optional matrix $(T \times M)$ of returns for a second (peer) group of M funds. When supplied, the ratios are computed for each fund in `X` against the funds in `Y` (cross-group screening) instead of against the other funds in `X`; a single focal fund versus a peer group corresponds to `X` being a vector. Columns of `Y` identical to the focal fund are automatically excluded. Default: `Y = NULL`, i.e. within-group screening.

Details

The Sharpe ratio (Sharpe 1994) 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. (The data-driven choice `bBoot = 0` is only available in `sharpeTesting`, not in screening.)
- `'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.
- `'gammaPos'` One-sided quantile level (of the standard Normal distribution) used as the critical value for counting outperformed peers: a peer counts as outperformed when the pairwise t-statistic exceeds `qnorm(gammaPos)` (a *negative* threshold for `gammaPos < 0.5`), and the expected fraction `1 - gammaPos` of false positives among the equal-performing peers is then subtracted. Default: `gammaPos = 0.4` (the value recommended in Ardia and Boudt, 2018).
- `'gammaNeg'` Mirror image of `gammaPos` for the peers that outperform the focal fund: the count uses `tstat <= qnorm(gammaNeg)` and subtracts the expected fraction `gammaNeg` of false positives. Default: `gammaNeg = 0.6`.
- `'fastAdjust'` Use a fast vectorised inversion in the truncated-normal bias correction of π^0 instead of one `uniroot` call per value. This is the dominant cost when `lambda` is data driven and gives a large speed-up on big universes. The bisection locates the root to about 1e-12; since `uniroot` stops at its own tolerance (about 1.2e-4), the two paths typically differ by a few 1e-5, the fast path being the more accurate. Default: `fastAdjust = FALSE`, i.e. the original code path, kept as default so that published results reproduce exactly.

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

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, 97–104.

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, 351–368.

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), 179–216.

Sharpe, W.F. (1994). The Sharpe ratio. *Journal of Portfolio Management* **21**(1), 49–58.

Ledoit, O., Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance* **15**(5), 850–859.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), 479–498.

See Also

sharpe, sharpeTesting, msharpeScreening and alphaScreening.

Examples

```
## Load the data (randomized data of monthly hedge fund returns)
data("hfdata")
rets = hfdata[,1:4]

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

## Sharpe screening with HAC standard errors (asymptotic test)
sharpeScreening(rets, control = list(nCore = 1, hac = TRUE))

## Sharpe screening with the studentized circular bootstrap
## (the 'hac' flag is ignored when type = 2)
sharpeScreening(rets, control = list(nCore = 1, type = 2))
```

`sharpeTesting` *Testing the difference of Sharpe ratios*

Description

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

Usage

```
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) returns for the second fund. `NA` values are allowed.
`control`	Control parameters (see *Details*).

Details

The Sharpe ratio (Sharpe 1994) 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 (`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`.

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 where adapted from Michael Wolf MATLAB code.

Author(s)

David Ardia and Kris Boudt.

References

Ardia, D., Boudt, K. (2015). Testing equality of modified Sharpe ratios. *Finance Research Letters* **13**, 97–104.

Ardia, D., Boudt, K. (2018). The peer performance ratios of hedge funds. *Journal of Banking and Finance* **87**, 351–368.

Barras, L., Scaillet, O., Wermers, R. (2010). False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* **65**(1), 179–216.

Sharpe, W.F. (1994). The Sharpe ratio. *Journal of Portfolio Management* **21**(1), 49–58.

Ledoit, O., Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *Journal of Empirical Finance* **15**(5), 850–859.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* **64**(3), 479–498.

See Also

`sharpe`, `sharpeScreening` and `msharpeTesting`.

Examples

```
## 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 = 100)
out = sharpeTesting(x, y, control = ctr)
print(out)

## Run Sharpe testing (circular bootstrap)
set.seed(1234)
ctr = list(type = 2, nBoot = 100, bBoot = 5)
out = sharpeTesting(x, y, control = ctr)
print(out)
```

---

summary.SCREENING *Summary method for the 'SCREENING' object*

Description

Produces a ranked summary of a screening: distribution of the performance measure, the peer performance ratios, optional win/loss counts from the pairwise tests, and the top/bottom funds (overall and by out-/underperformance ratio). The win/loss counts require the square (within-group) pairwise matrices and are therefore omitted for cross-group screenings (a Y peer group or targetPeerPerformance). Originally contributed by Murilo Andre Peres Pereira.

Usage

```
## S3 method for class 'SCREENING'
summary(object, coef = 1L, top = 5L, p_level = 0.05, ...)
```

Arguments

object	A SCREENING object.
coef	For a screen_beta screening, which coefficient row to use (default 1, the alpha).

top	Number of best/worst funds to display. Default: 5.
p_level	Significance level for the win/loss counts (used only when the object carries square pval/tstat matrices). Default: 0.05.
...	Further arguments (ignored).

Value

An object of class summary.SCREENING.

Author(s)

Murilo Andre Peres Pereira, David Ardia and Kris Boudt.

See Also

alphaScreening.

Examples

```
data("hfdata")
sc <- alphaScreening(hfdata[, 1:10], control = list(nCore = 1))
summary(sc)
```

---

targetPeerPerformance *Targeted peer-performance screening for selected funds*

Description

Performs peer-performance screening for a user-selected subset of funds (the "focal" funds) against the whole universe, returning detailed outputs for those focal funds only. It is a convenience wrapper that calls the corresponding screening function with the cross-group argument Y set to the full matrix X; equivalently, targetPeerPerformance(X, funds, "alpha") is alphaScreening(X[, funds], Y = X).

Usage

```
targetPeerPerformance(
  X,
  funds,
  method = c("alpha", "sharpe", "msharpe"),
  factors = NULL,
  level = 0.9,
  na.neg = TRUE,
  control = list()
)
```

Arguments

- `X` Matrix $(T \times N)$ of T returns for N funds. `NA` values are allowed.
- `funds` Integer indices or character column names identifying the focal funds. The output preserves the order provided in `funds`.
- `method` Screening method: `"alpha"` (default), `"sharpe"`, or `"msharpe"`.
- `factors` Optional matrix of factor returns (only when `method = "alpha"`).
- `level` Modified Value-at-Risk level (only when `method = "msharpe"`).
- `na.neg` Logical; return `NA` for a negative modified Value-at-Risk (only when `method = "msharpe"`).
- `control` Control parameters (see `alphaScreening`).

Value

A `SCREENING` object (see `alphaScreening`) whose fund-level components (`pizero`, `pipos`, `pineg`, the performance measure, `npeer`, ...) cover the focal funds only, and whose `pval`/`tstat`/difference matrices have one row per focal fund and one column per universe fund. Works with the `print`, `plot`, `summary`, and `as.data.frame` methods.

Author(s)

Murilo Andre Peres Pereira, David Ardia and Kris Boudt.

See Also

`alphaScreening`, `sharpeScreening`, `msharpeScreening`.

Examples

```
data("hfdata")
rets <- hfdata[, 1:10]

## focal Sharpe screening (funds by index) against the whole universe
out <- targetPeerPerformance(rets, funds = c(2, 5, 7), method = "sharpe",
                             control = list(nCore = 1))
out$sharpe
```

Index