# Package ‘RPESE’

*September 22, 2021*

**Type** Package  
**Title** Estimates of Standard Errors for Risk and Performance Measures  
**Version** 1.2.4  
**Date** 2021-09-21  
**Author** Anthony Christidis <anthony.christidis@stat.ubc.ca>, Xin Chen <chenx26@uw.edu>  
**Maintainer** Anthony Christidis <anthony.christidis@stat.ubc.ca>  
**Description** Estimates of standard errors of popular risk and performance measures for asset or portfolio returns using methods as described in Chen and Martin (2021) [doi:10.21314/JOR.2020.446].  
**Biarch** true  
**License** GPL (>= 2)  
**Imports** xts, zoo, boot, RPEIF, RPEGLMEN, RobStatTM  
**Suggests** testthat, R.rsp, PerformanceAnalytics  
**Depends**  
**RoxygenNote** 7.1.1  
**VignetteBuilder** R.rsp  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2021-09-22 17:40:02 UTC

**R topics documented:**

- DSR.SE .......................................................... 2  
- ES.SE ............................................................ 3  
- ESRatio.SE ...................................................... 5  
- EstimatorSE .................................................... 7  
- LPM.SE .......................................................... 8  
- Mean.SE ......................................................... 10  
- OmegaRatio.SE ................................................ 11
printSE ................................................................. 13
RachevRatio.SE ...................................................... 14
robMean.SE ................................................................. 15
SD.SE ........................................................................ 17
SemiSD.SE ................................................................. 18
SoR.SE ...................................................................... 20
SR.SE ...................................................................... 21
VaR.SE ...................................................................... 23
VaRratio.SE ................................................................. 24

Index ........................................................................ 27

DSR.SE  Standard Error Estimate for Downside Sharpe Ratio (DSR) of Returns

Description

ES.SE computes the standard error of the downside Sharpe ratio of the returns.

Usage

DSR.SE(
  data,
  rf = 0,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[, 4],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[, 1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[, 1],
  freq.par = 0.5,
  corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[, 1],
  return.coef = FALSE,
  ...
)

Arguments

data         Data of returns for one or multiple assets or portfolios.
rf           Risk free rate.
se.method    A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor" (default), "IFcorPW", "IFcorAdapt", "BOOTiid" or "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitenning of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
ES.SE

ES.SE computes the standard error of the expected shortfall of the returns.

Value

A vector or a list depending on se.method.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
DSR.SE(edhec, se.method = c("IFiid","IFcor"),
cleanOutliers = FALSE,
fitting.method = c("Exponential", "Gamma")[[1]])
Usage

ES.SE(
  data,
  p = 0.95,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[1:2],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  freq.par = 0.5,
  corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
  return.coef = FALSE,
  ...
)

Arguments

data                  Data of returns for one or multiple assets or portfolios.
p                     Confidence level for calculation. Default value is p = 0.95.
se.method             A character string indicating which method should be used to compute the
                      standard error of the estimated standard deviation. One or a combination of:
                      "IFiid" (default), "IFcor" (default), "IFcorPW", "IFcorAdapt", "BOOTiid"
                      or "BOOTcor".
cleanOutliers         Boolean variable to indicate whether the pre-whitening of the influence
                      functions TS should be done through a robust filter. Default if FALSE.
fitting.method        Distribution used in the standard errors computation. Should be one of "Exponen-
                      tial" (default) or "Gamma".
d.GLM.EN              Order of the polynomial for the Exponential or Gamma fitting. Default poly-
                      nomial order of 5.
freq.include          Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate"
                      or "Truncate."
freq.par              Percentage of the frequency used if "freq.include" is "Decimate" or "Trun-
                      cate." Default is 0.5.
corOut                Return correlation of the returns or the influence function transformed returns.
                      Must be one of "retCor", "retIFCor" or "none" (default).
return.coef            Boolean variable to indicate whether the coefficients of the penalized GLM fit
                      are returned. Default if FALSE.
...                    Additional parameters.

Value

A vector or a list depending on se.method.

Author(s)

Xin Chen, <chenx26@uw.edu>
Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>
### Examples

```r
# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN",
                 "ED", "FIA", "GM", "LS", "MA",
                 "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
ES.SE(edhec, se.method = c("IFiid","IFcor"),
      cleanOutliers = FALSE,
      fitting.method = c("Exponential", "Gamma"))
```

### Description

**ESratio.SE** computes the standard error of the expected shortfall ratio of the returns.

### Usage

```r
ESratio.SE(data,
           alpha = 0.1,
           rf = 0,
           se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[[1]],
           cleanOutliers = FALSE,
           fitting.method = c("Exponential", "Gamma")[[1]],
           d.GLM.EN = 5,
           freq.include = c("All", "Decimate", "Truncate")[[1]],
           freq.par = 0.5,
           corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[[1]],
           return.coef = FALSE,
           ...
)
```

### Arguments

- **data** Data of returns for one or multiple assets or portfolios.
- **alpha** Lower tail probability.
- **rf** Risk-free interest rate.
se.method A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or "BOOTcor".

cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.

fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".

d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.

freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."

freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.

corOut Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).

return.coef Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.

... Additional parameters.

Value

A vector or a list depending on se.method.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for the two influence functions based approaches
ESratio.SE(edhec, se.method=c("IFiid", "IFcorAdapt"),
            cleanOutliers=FALSE,
            fitting.method=c("Exponential", "Gamma")[1])
Description

EstimatorSE computes the standard error for specified risk and performance measures.

Usage

EstimatorSE(
  data,
  estimator.fun = c("DSR", "ES", "ESratio", "LPM", "Mean", "OmegaRatio", "RachevRatio", "robMean", "SD", "SemiSD", "SR", "SoR", "VaR", "VaRratio")[1],
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[1],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  freq.par = 0.5,
  a = 0.3,
  b = 0.7,
  return.coef = FALSE,
  ...
)

Arguments

data Data of returns for one or multiple assets or portfolios.
estimator.fun Risk or performance measure to compute estimates of standard errors.
se.method A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One of: "IFiid"; "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor", or "none".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
a First adaptive method parameter.
b Second adaptive method parameter.
return.coef  

Boolean variable to indicate whether the coefficients of the Exponential or Gamma fit are returned. Default is FALSE.

...  

Additional parameters.

Value

A vector standard error estimates.

Author(s)

Xin Chen, <chenx26@uw.edu>

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN",
    "ED", "FIA", "GM", "LS", "MA",
    "RV", "SS", "FOF")
# Computing the standard errors for
# the three influence functions based approaches
EstimatorSE(edhec[,"CA"], se.method = c("IFcor"),
cleanOutliers = FALSE,
fitting.method = c("Exponential", "Gamma")[1])

LPM.SE  

Standard Error Estimate for Lower Partial Moment (LPM) of Returns

Description

LPM.SE computes the standard error of the LPM of the returns.

Usage

LPM.SE(
  data,
  const = 0,
  order = 1,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[1:2],
cleanOutliers = FALSE,
fitting.method = c("Exponential", "Gamma")[1],
d.GLM.EN = 5,
freq.include = c("All", "Decimate", "Truncate")[1],
freq.par = 0.5,
corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
return.coef = FALSE,
...  
)

Arguments

- **data**: Data of returns for one or multiple assets or portfolios.
- **const**: Constant threshold.
- **order**: Order for the lower partial moment (should be 1 or 2).
- **se.method**: A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor" (default), "IFcorPW", "IFcorAdapt", "BOOTiid" or "BOOTcor".
- **cleanOutliers**: Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
- **fitting.method**: Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
- **d.GLM.EN**: Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
- **freq.include**: Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
- **freq.par**: Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
- **corOut**: Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
- **return.coef**: Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.

Value

A vector or a list depending on se.method.

Author(s)

Xin Chen, <chenx26@uw.edu>
Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
LPM.SE(edhec, se.method = c("IFiid","IFcor"),
       cleanOutliers = FALSE,
       fitting.method = c("Exponential", "Gamma")[1])

Mean.SE STANDARD ERROR ESTIMATE FOR MEAN OF RETURNS

Description

Mean.SE computes the standard error of the mean of the returns.

Usage

Mean.SE(
  data,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[[c(1, 4)],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  freq.par = 0.5,
  corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[[1],
  return.coef = FALSE,
  ...
)

Arguments

data Data of returns for one or multiple assets or portfolios.
se.method A character string indicating which method should be used to compute the
            standard error of the estimated standard deviation. One or a combination of:
            "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or
            "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence func-
           tions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Expo-
           nential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default poly-
           nomial order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or
           "Truncate."
freq.par: Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.

corOut: Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).

return.coef: Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.

Value
A vector or a list depending on se.method

Author(s)
Anthony-Alexander Christidis, anthony.christidis@stat.ubc.ca

Examples
# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
Mean.SE(edhec, se.method = c("IFiid","IFcorAdapt"),
        cleanOutliers = FALSE,
        fitting.method = c("Exponential", "Gamma")[1])

OmegaRatio.SE

Standard Error Estimate for Omega Ratio of Returns

Description
OmegaRatio.SE computes the standard error of the Omega ratio of the returns.

Usage
OmegaRatio.SE(
    data,
    const = 0,
    se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[c(1, 4)],
    cleanOutliers = FALSE,
    fitting.method = c("Exponential", "Gamma")[1],
    d.GLM.EN = 5,
freq.include = c("All", "Decimate", "Truncate")[1],
freq.par = 0.5,
corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
return.coef = FALSE,
...
)

Arguments

data Data of returns for one or multiple assets or portfolios.
const Constant threshold.
se.method A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid", "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
corOut Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
return.coef Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
...
Value
A vector or a list depending on se.method.

Author(s)
Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN",
"ED", "FIA", "GM", "LS", "MA",
"RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
OmegaRatio.SE(edhec, se.method = c("IFiid","IFcorAdapt")[1],
    cleanOutliers = FALSE,
    fitting.method = c("Exponential", "Gamma")[1])

printSE

## Formatted Output for Standard Errors Functions in RPESE

### Description

`printSE` returns a formatted output from standard error functions from RPESE.

### Usage

```r
printSE(SE.data, round.digit = 3, round.out = TRUE)
```

### Arguments

- **SE.data**: Standard error estimates output from RPESE functions.
- **round.digit**: Number of digits for rounding.
- **round.out**: Round data (TRUE) with round.digit number of digits. Default is TRUE.

### Value

A data frame with formatted output from standard error functions from RPESE.

### Author(s)

Xin Chen, <chenx26@uw.edu>

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

### Examples

```r
# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN",
    "ED", "FIA", "GM", "LS", "MA",
    "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
ES.out <- ES.SE(edhec, se.method = c("IFiid","IFcor"),
    cleanOutliers = FALSE,
    fitting.method = c("Exponential", "Gamma")[1])
# Print the output
printSE(ES.out)
```
RachevRatio.SE  
Standard Error Estimate for Rachev Ratio of Returns

Description

RachevRatio.SE computes the standard error of the Rachev ratio of the returns.

Usage

RachevRatio.SE(
  data,
  alpha = 0.1,
  beta = 0.1,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[c(1, 4)],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  freq.par = 0.5,
  corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
  return.coef = FALSE,
  ...
)

Arguments

data  Data of returns for one or multiple assets or portfolios.
alpha  Lower tail probability.
beta  Upper tail probability.
se.method  A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or "BOOTcor".
cleanOutliers  Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method  Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN  Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
freq.include  Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
freq.par  Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
robMean.SE

- **corOut**: Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).

- **return.coef**: Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.

- **...**: Additional parameters.

**Value**

A vector or a list depending on se.method.

**Author(s)**

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

**Examples**

```r
# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for the two influence functions based approaches
RachevRatio.SE(edhec, se.method = c("IFiid","IFcorAdapt"),
                cleanOutliers = FALSE,
                fitting.method = c("Exponential", "Gamma")[1])
```

**Description**

robMean.SE computes the standard error of the robust location (mean) M-estimator of the returns.

**Usage**

```r
robMean.SE(
  data,
  family = c("mopt", "opt", "bisquare")[1],
  eff = 0.95,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[c(1, 4)],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
)```
freq.include = c("All", "Decimate", "Truncate")[1],
freq.par = 0.5,
corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
return.coef = FALSE,
...)

Arguments

data Data of returns for one or multiple assets or portfolios.
family Family for robust m-estimator of location. Must be one of "mopt" (default),
"opt" or "bisquare".
eff Tuning parameter for the normal distribution efficiency. Default is 0.99.
se.method A character string indicating which method should be used to compute the
standard error of the estimated standard deviation. One or a combination of:
"IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid"
or "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitenning of the influence func-
tions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential"
(default) or "Gamma".
d.GLMMEN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial
order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate"
or "Truncate."
freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
corOut Return correlation of the returns or the influence function transformed returns.
Must be one of "retCor", "retIFCor" or "none" (default).
return.coef Boolean variable to indicate whether the coefficients of the penalized GLM fit
are returned. Default if FALSE.
...

Value

A vector or a list depending on se.method.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")

# Computing the standard errors for
# the two influence functions based approaches
robMean.SE(edhec, se.method = c("IFiid","IFcorAdapt"),
fitting.method = c("Exponential", "Gamma")[1],
family = "mopt", eff = 0.95)

---

**SD.SE**

*Standard Error Estimate for Standard Deviation (SD) of Returns*

**Description**

SD.SE computes the standard error of the standard deviation of the returns.

**Usage**

```r
SD.SE(
  data, 
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[1:2],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  freq.par = 0.5,
  corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
  return.coef = FALSE,
  ...
)
```

**Arguments**

- **data**
  Data of returns for one or multiple assets or portfolios.

- **se.method**
  A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor" (default), "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or "BOOTcor".

- **cleanOutliers**
  Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.

- **fitting.method**
  Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".

- **d.GLM.EN**
  Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.

- **freq.include**
  Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
SemiSD.SE

SemiSD.SE computes the standard error of the SSD of the returns.

Usage

SemiSD.SE(
  data,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[1:2],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  ...)
freq.par = 0.5,
corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
return.coef = FALSE,
...
)

Arguments

data Data of returns for one or multiple assets or portfolios.
se.method A character string indicating which method should be used to compute the
standard error of the estimated standard deviation. One or a combination of:
"IFiid" (default), "IFcor" (default), "IFcorPW", "IFcorAdapt", "BOOTiid", "BOOTcor", or "none".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence func-
tions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
corOut Return correlation of the returns or the influence function transformed returns.
Must be one of "retCor", "retIFCor" or "none" (default).
return.coef Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
...

Value

A vector or a list depending on se.method.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
SemiSD.SE(edhec, se.method = c("IFiid", "IFcor"),
    cleanOutliers = FALSE,
    fitting.method = c("Exponential", "Gamma")[1])

SoR.SE Standard Error Estimate for Sortino Ratio (SoR) of Returns

Description

SoR.SE computes the standard error of the Sortino ratio of the returns.

Usage

SoR.SE(
    data,
    const = 0,
    threshold = c("mean", "const")[1],
    se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[[1]],
    cleanOutliers = FALSE,
    fitting.method = c("Exponential", "Gamma")[1],
    d.GLM.EN = 5,
    freq.include = c("All", "Decimate", "Truncate")[[1]],
    freq.par = 0.5,
    corOut = c("none", "retCor", "retIFCor")[[1]],
    return.coef = FALSE,
    ...
)

Arguments

data Data of returns for one or multiple assets or portfolios.
const Minimum acceptable return for threshold.
threshold Parameter to determine whether we use a "mean" or "const" threshold.
se.method A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
SR.SE

Description

SR.SE computes the standard error of the Sharpe ratio of the returns.

Usage

SR.SE(
  data,
  rf = 0,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor") [c(1, 4)],
...)
cleanOutliers = FALSE, 
fitting.method = c("Exponential", "Gamma")[1], 
d.GLM.EN = 5, 
freq.include = c("All", "Decimate", "Truncate")[1], 
freq.par = 0.5, 
corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1], 
return.coef = FALSE, 
... 
)

Arguments

data Data of returns for one or multiple assets or portfolios.
rf Risk free rate.
se.method A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
corOut Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
return.coef Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
...
Additional parameters.

Value

A vector or a list depending on se.method.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN",  
"ED", "FIA", "GM", "LS", "MA",  
"RV", "SS", "FOF")

# Computing the standard errors for  
# the two influence functions based approaches  
SR.SE(edhec, se.method = c("IFiid","IFcorAdapt"),  
cleanOutliers = FALSE,  
fitting.method = c("Exponential", "Gamma")[1])

---

## VaR.SE

### Standard Error Estimate for Value-at-Risk (VaR) of Returns

#### Description

VaR.SE computes the standard error of the value-at-risk of the returns.

#### Usage

```r
VaR.SE(  
data = NULL,  
alpha = 0.95,  
se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[1:2],  
cleanOutliers = FALSE,  
fitting.method = c("Exponential", "Gamma")[1],  
d.GLM.EN = 5,  
freq.include = c("All", "Decimate", "Truncate")[1],  
freq.par = 0.5,  
corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],  
return.coef = FALSE,  
...  
)
```

#### Arguments

- **data**  
  Data of returns for one or multiple assets or portfolios.
- **alpha**  
  Confidence level for calculation. Default is alpha=0.95.
- **se.method**  
  A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor" (default), "IFcorPW", "IFcorAdapt", "BOOTiid" or "BOOTcor".
- **cleanOutliers**  
  Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
- **fitting.method**  
  Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN  Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.

freq.include  Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."

freq.par  Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.

corOut  Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).

return.coef  Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.

...  Additional parameters.

Value

A vector or a list depending on se.method.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
VaR.SE(edhec, se.method = c("IFiid","IFcor"),
       cleanOutliers = FALSE,
       fitting.method = c("Exponential", "Gamma")[1])

---

**VaRratio.SE**  **Standard Error Estimate for Value-at-Risk Ratio (VaRratio) of Returns**

**Description**

VaRratio.SE computes the standard error of the value-at-risk ratio of the returns.
Usage

VaRratio.SE(
  data,
  alpha = 0.1,
  rf = 0,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[c(1, 4)],
  cleanOutliers = FALSE,
  fitting.method = c("Exponential", "Gamma")[1],
  d.GLM.EN = 5,
  freq.include = c("All", "Decimate", "Truncate")[1],
  freq.par = 0.5,
  corOut = c("none", "retCor", "retIFCor", "retIFCorPW")[1],
  return.coef = FALSE,
  ...)

Arguments

data Data of returns for one or multiple assets or portfolios.
alpha The tail probability of interest.
rf Risk-free interest rate.
se.method A character string indicating which method should be used to compute the standard error of the estimated standard deviation. One or a combination of: "IFiid" (default), "IFcor", "IFcorPW", "IFcorAdapt" (default), "BOOTiid" or "BOOTcor".
cleanOutliers Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
fitting.method Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
d.GLM.EN Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
freq.include Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
freq.par Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
corOut Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
return.coef Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
...

Value

A vector or a list depending on se.method.
Author(s)
Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

Examples

# Loading data
data(edhec, package = "PerformanceAnalytics")
# Changing the data colnames
names(edhec) = c("CA", "CTA", "DIS", "EM", "EMN", "ED", "FIA", "GM", "LS", "MA", "RV", "SS", "FOF")
# Computing the standard errors for
# the two influence functions based approaches
VaRratio.SE(edhec, se.method = c("IFiid", "IFcorAdapt"), cleanOutliers = FALSE, fitting.method = c("Exponential", "Gamma") [1])
Index

DSR.SE, 2
ES.SE, 3
ESratio.SE, 5
EstimatorSE, 7
LPM.SE, 8
Mean.SE, 10
OmegaRatio.SE, 11
printSE, 13
RachevRatio.SE, 14
robMean.SE, 15
SD.SE, 17
SemiSD.SE, 18
SoR.SE, 20
SR.SE, 21
VaR.SE, 23
VaRratio.SE, 24