

# Package ‘RPESE’

October 24, 2020

**Type** Package

**Title** Estimates of Standard Errors for Risk and Performance Measures

**Version** 1.1.0

**Date** 2020-10-23

**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 (2018) <<https://www.ssrn.com/abstract=3085672>>.

**Biarch** true

**License** GPL (>= 2)

**Imports** xts, zoo, boot, sandwich, RPEIF, RPEGLMEN,

**Suggests** testthat, R.rsp

**Depends** R (>= 3.5.0)

**RoxygenNote** 7.0.2

**VignetteBuilder** R.rsp

**LazyData** true

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-10-24 08:20:02 UTC

## R topics documented:

DSR.SE . . . . .	2
edhec . . . . .	3
ES.SE . . . . .	5
ESratio.SE . . . . .	6
EstimatorSE . . . . .	8
LPM.SE . . . . .	9

Mean.SE . . . . .	11
OmegaRatio.SE . . . . .	12
printSE . . . . .	14
RachevRatio.SE . . . . .	15
SD.SE . . . . .	16
SemiSD.SE . . . . .	18
SoR.SE . . . . .	19
SR.SE . . . . .	21
VaR.SE . . . . .	22
VaRratio.SE . . . . .	24

<b>Index</b>	<b>26</b>
--------------	-----------

---

DSR.SE	<i>Standard Error Estimate for Downside Sharpe Ratio (DSR) of Returns</i>
--------	---

---

## 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")[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.
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-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.

<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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)
# 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])
```

---

edhec

---

*EDHEC-Risk Hedge Fund Style Indices*


---

**Description**

EDHEC composite hedge fund style index returns.

**Usage**

```
edhec
```

## Format

CSV conformed into an xts object with monthly observations

## Details

EDHEC Data used in PerformanceAnalytics and related publications with the kind permission of the EDHEC Risk and Asset Management Research Center.

The 'edhec' data set included with PerformanceAnalytics will be periodically updated (typically annually) to include additional observations. If you intend to use this data set in automated tests, please be sure to subset your data like `edhec[1:120, ]` to use the first ten years of observations.

From the EDHEC website: "The EDHEC Risk and Asset Management Research Centre plays a noted role in furthering applied financial research and systematically highlighting its practical uses. As part of its philosophy, the centre maintains a dialogue with professionals which benefits the industry as a whole. At the same time, its proprietary R&D provides sponsors with an edge over competition and joint ventures allow selected partners to develop new business opportunities.

To further assist financial institutions and investors implement the latest research advances in order to meet the challenges of the changing asset management landscape, the centre has spawned two consultancies and an executive education arm. Clients of these derivative activities include many of the leading organisations throughout Europe."

see <https://risk.edhec.edu/>

## Source

<https://risk.edhec.edu/indices-investment-solutions/>

## References

About EDHEC Alternative Indexes. December 16, 2003. EDHEC-Risk.

<https://risk.edhec.edu/indices-investment-solutions/>

Vaissie Mathieu. A Detailed Analysis of the Construction Methods and Management Principles of Hedge Fund Indices. October 2003. EDHEC.

<https://risk.edhec.edu/>

## Examples

```
data(edhec)

#preview the data
head(edhec)

#summary period statistics
summary(edhec)

#cumulative index returns
tail(cumprod(1+edhec),1)
```

ES.SE

*Standard Error Estimate for Expected Shortfall (ES) of Returns***Description**

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

**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**

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>p</code>	Confidence level for calculation. Default value is $p = 0.95$ .
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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**

```
# Loading data
data(edhec)
# 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")[1])
```

---

ESratio.SE

*Standard Error Estimate for Expected Shortfall Ratio (ESratio) of Returns*

---

**Description**

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

**Usage**

```
ESratio.SE(
  data,
  alpha = 0.1,
  rf = 0,
  se.method = c("IFiid", "IFcor", "IFcorAdapt", "IFcorPW", "BOOTiid", "BOOTcor")[c(1,
    3)],
  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**

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>alpha</code>	Lower tail probability.
<code>rf</code>	Risk-free interest rate.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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)
# 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])
```

EstimatorSE

*Wrapper Function for Standard Errors Estimates Functions***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",
    "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**

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>estimator.fun</code>	Risk or performance measure to compute estimates of standard errors.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>a</code>	First adaptive method parameter.
<code>b</code>	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)
# 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

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

```

# Loading data
data(edhec)
# 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	<i>Standard Error Estimate for Mean of Returns</i>
---------	--

---

### 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,
    3)],
  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

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	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)
# 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,
  3)],
  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

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>const</code>	Constant threshold.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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)
# 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

```
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

```
# Loading data
data(edhec)
# 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,
    3)],
  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.

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 from PerformanceAnalytics
data(edhec)
# 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])
```

---

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**

```
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**

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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)
# 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
SD.SE(edhec, se.method = c("IFiid","IFcor","IFcorAdapt"),
      cleanOutliers = FALSE,
      fitting.method = c("Exponential", "Gamma")[1])
```

---

SemiSD.SE	<i>Standard Error Estimate for Semi-Standard Deviation (SemiSD) of Returns</i>
-----------	--

---

### Description

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 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)
# 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")[c(1,
    3)],
  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**

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>const</code>	Minimum acceptable return for threshold.
<code>threshold</code>	Parameter to determine whether we use a "mean" or "const" threshold.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFcor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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)
# 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
SoR.SE(edhec, se.method = c("IFiid", "IFcorAdapt"),
       cleanOutliers = FALSE,
       fitting.method = c("Exponential", "Gamma")[1])
```

---

 SR.SE

*Standard Error Estimate for Sharpe Ratio (SR) of Returns*


---

### 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,
    3)],
  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

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>rf</code>	Risk free rate.
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	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)
# 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**

```
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**

<code>data</code>	Data of returns for one or multiple assets or portfolios.
<code>alpha</code>	Confidence level for calculation. Default is $\alpha=0.95$ .
<code>se.method</code>	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".
<code>cleanOutliers</code>	Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. Default if FALSE.
<code>fitting.method</code>	Distribution used in the standard errors computation. Should be one of "Exponential" (default) or "Gamma".
<code>d.GLM.EN</code>	Order of the polynomial for the Exponential or Gamma fitting. Default polynomial order of 5.
<code>freq.include</code>	Frequency domain inclusion criteria. Must be one of "All" (default), "Decimate" or "Truncate."
<code>freq.par</code>	Percentage of the frequency used if "freq.include" is "Decimate" or "Truncate." Default is 0.5.
<code>corOut</code>	Return correlation of the returns or the influence function transformed returns. Must be one of "retCor", "retIFCor" or "none" (default).
<code>return.coef</code>	Boolean variable to indicate whether the coefficients of the penalized GLM fit are returned. Default if FALSE.
<code>...</code>	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)
# 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,
    3)],
  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.
...	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)
# 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

## \* datasets

edhec, 3

DSR.SE, 2

edhec, 3

ES.SE, 5

ESratio.SE, 6

EstimatorSE, 8

LPM.SE, 9

Mean.SE, 11

OmegaRatio.SE, 12

printSE, 14

RachevRatio.SE, 15

SD.SE, 16

SemiSD.SE, 18

SoR.SE, 19

SR.SE, 21

VaR.SE, 22

VaRatio.SE, 24