

# Package ‘RPEIF’

July 3, 2019

**Type** Package

**Title** Computation and Plots of Influence Functions for Risk and Performance Measures

**Version** 1.0

**Date** 2019-06-30

**Author** Anthony Christidis <anthony.christidis@stat.ubc.ca>,  
Shengyu Zhang <syzhang@uw.edu>,  
Douglas Martin <doug@amath.washington.edu>

**Maintainer** Anthony Christidis <anthony.christidis@stat.ubc.ca>

**Description** Computes the influence functions time series of the returns for the risk and performance measures as mentioned in Zhang and Martin (2017) <<https://ssrn.com/abstract=2747179>> as well as Chen and Martin (2018) <<https://ssrn.com/abstract=3085672>>. Also evaluates estimators influence functions at a set of parameter values and plots them to display the shapes of the influence functions.

**License** GPL (>= 2)

**Biarch** true

**Imports** Rcpp (>= 0.12.17), ggplot2, PerformanceAnalytics, xts, zoo,  
RobStatTM, stats

**Depends**

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 6.1.1

**Suggests** R.rsp, testthat

**VignetteBuilder** R.rsp

**NeedsCompilation** yes

**LazyData** true

**Repository** CRAN

**Date/Publication** 2019-07-03 16:20:07 UTC

## R topics documented:

|              |           |
|--------------|-----------|
| IF           | 2         |
| IF.ES        | 4         |
| IF.ESratio   | 5         |
| IF.LPM       | 7         |
| IF.mean      | 8         |
| IF.Omega     | 10        |
| IF.RachR     | 12        |
| IF.SD        | 13        |
| IF.SoR       | 15        |
| IF.SR        | 17        |
| IF.SSD       | 18        |
| IF.VaR       | 20        |
| IF.VaRratio  | 21        |
| nuisParsFn   | 23        |
| <b>Index</b> | <b>25</b> |

---

IF *Influence Function for Available Risk Measures*

---

### Description

IF returns the data and plots the shape of either the IF or the IF TS for a risk measure specified.

### Usage

```
IF(risk, returns = NULL, evalShape = FALSE, retVals = NULL,
   nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
   compile = TRUE, prewhiten = FALSE, ar.prewhiten.order = 1,
   cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
   eff = 0.99, alpha.robust = 0.05, ...)
```

### Arguments

|           |  |
|-----------|--|
| risk      | Risk measure.  |
| returns   | Vector of the returns of the asset or portfolio.   |
| evalShape | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed. |
| retVals   | Values used to evaluate the shape of the IF.   |
| nuisPars  | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).                       |
| k         | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot    | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint   | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |

|                                 |  |
|---------------------------------|--|
| <code>compile</code>            | Boolean variable to indicate if the IF TS should be computed using compiled code (C++) (TRUE) or not (FALSE).                              |
| <code>prewhiten</code>          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| <code>ar.prewhiten.order</code> | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| <code>cleanOutliers</code>      | Boolean variable to indicate whether the pre-whitening of the IF TS should be done through a robust filter.                                |
| <code>cleanMethod</code>        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| <code>eff</code>                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| <code>alpha.robust</code>       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| <code>...</code>                | Additional parameters passed on to influence function of risk measure.   |

### Author(s)

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

### Examples

```
# Plot of IF using the wrapper function
outIF <- IF(risk="mean",
           returns=NULL, evalShape=TRUE, retVals=NULL, nuisPars=list(mu=0.005),
           IFplot=TRUE, IFprint=TRUE)

#' # Loading data (hedge funds returns)
data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF using wrapper function and with a specified TS
outIF <- IF(risk="mean",
           returns=edhec[, "CA"], evalShape=TRUE,
           retVals=seq(-0.1, 0.1, by=0.001), nuisPars=NULL,
           IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF(risk="mean",
           returns=edhec[, "CA"], evalShape=FALSE, retVals=NULL, nuisPars =NULL,
           IFplot=TRUE, IFprint=TRUE,
           compile=TRUE, prewhiten=FALSE,
           cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

IF.ES

*Influence Function - Expected Shortfall (ES)***Description**

IF.ES returns the data and plots the shape of either the IF or the IF TS for the ES

**Usage**

```
IF.ES(returns = NULL, evalShape = FALSE, retVals = NULL,
      nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
      alpha.ES = 0.05, prewhiten = FALSE, ar.prewhiten.order = 1,
      cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
      eff = 0.99, alpha.robust = 0.05, ...)
```

**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| alpha.ES           | Tail Probability.  |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of the ES.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.ES(returns=NULL, evalShape=TRUE,
              retVals=NULL, nuisPars =NULL,
              IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.ES(returns=edhec[, "CA"], evalShape=TRUE,
              retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
              IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.ES(returns=edhec[, "CA"], evalShape=FALSE,
              retVals=NULL, nuisPars =NULL,
              IFplot=TRUE, IFprint=TRUE,
              prewhiten=FALSE,
              cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

---

 IF.ESratio

*Influence Function - Expected Shortfall (ES) Ratio*


---

**Description**

IF.ESratio returns the data and plots the shape of either the IF or the IF TS for the Expected Shortfall Ratio.

**Usage**

```
IF.ESratio(returns = NULL, evalShape = FALSE, retVals = NULL,
           nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
           alpha = 0.1, rf = 0, prewhiten = FALSE, ar.prewhiten.order = 1,
           cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
           eff = 0.99, alpha.robust = 0.05, ...)
```

**Arguments**

**returns**            Vector of the returns of the asset or portfolio.

**evalShape**        Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.

|                    |  |
|--------------------|--|
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| alpha              | Tail Probability.  |
| rf                 | Risk-free interest rate.   |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of ESratio

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.ESratio(returns=NULL, evalShape=TRUE,
                   retVals=NULL, nuisPars =NULL,
                   IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.ESratio(returns=edhec[, "CA"], evalShape=TRUE,
                   retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
                   IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.ESratio(returns=edhec[, "CA"], evalShape=FALSE,
```

```

retVals=NULL, nuisPars =NULL,
IFplot=TRUE, IFprint=TRUE,
prewhiten=FALSE,
cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)

```

IF.LPM

*Influence Function - Lower Partial Moment (LPM)***Description**

IF.LPM returns the data and plots the shape of either the IF or the IF TS for the LPM

**Usage**

```

IF.LPM(returns = NULL, evalShape = FALSE, retVals = NULL,
       nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
       const = 0, order = 1, prewhiten = FALSE, ar.prewhiten.order = 1,
       cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
       eff = 0.99, alpha.robust = 0.05, ...)

```

**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| const              | Constant threshold.  |
| order              | Order of LPM. Can only take values 1 or 2.   |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of LPM.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.LPM(returns=NULL, evalShape=TRUE,
               retVals=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.LPM(returns=edhec[, "CA"], evalShape=TRUE,
               retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.LPM(returns=edhec[, "CA"], evalShape=FALSE,
               retVals=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE,
               prewhiten=FALSE,
               cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

---

IF.mean

*Influence Function - Mean*

---

**Description**

IF.mean returns the data and plots the shape of either the IF or the IF TS for the mean.

**Usage**

```
IF.mean(returns = NULL, evalShape = FALSE, retVals = NULL,
        nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
        compile = TRUE, prewhiten = FALSE, ar.prewhiten.order = 1,
        cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
        eff = 0.99, alpha.robust = 0.05, ...)
```



**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| compile            | Boolean variable to indicate if the IF TS should be computed using compiled code (C++) (TRUE) or not (FALSE).                              |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function for the specified risk measure.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.mean(returns=NULL, evalShape=TRUE, retVals=NULL, nuisPars =NULL,
                IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.mean(risk="mean",
```

```

returns=edhec[, "CA"], evalShape=TRUE,
retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.mean(returns=edhec[, "CA"], evalShape=FALSE,
retVals=NULL, nuisPars =NULL,
IFplot=TRUE, IFprint=TRUE,
compile=TRUE, prewhiten=FALSE,
cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)

```

---

IF.Omega

*Influence Function - Omega Ratio*


---

### Description

IF.OmegaRatio returns the data and plots the shape of either the IF or the IF TS for the Omega Ratio.

### Usage

```

IF.Omega(returns = NULL, evalShape = FALSE, retVals = NULL,
nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
const = 0, compile = TRUE, prewhiten = FALSE,
ar.prewhiten.order = 1, cleanOutliers = FALSE,
cleanMethod = c("locScaleRob", "Boudt")[1], eff = 0.99,
alpha.robust = 0.05, ...)

```

### Arguments

|           |  |
|-----------|--|
| returns   | Vector of the returns of the asset or portfolio.   |
| evalShape | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed. |
| retVals   | Values used to evaluate the shape of the IF.   |
| nuisPars  | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).                       |
| k         | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot    | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint   | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| const     | Constant threshold.  |
| compile   | Boolean variable to indicate if the IF TS should be computed using compiled code (C++) (TRUE) or not (FALSE).          |
| prewhiten | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).                                       |

|                                 |  |
|---------------------------------|--|
| <code>ar.prewhiten.order</code> | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| <code>cleanOutliers</code>      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| <code>cleanMethod</code>        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| <code>eff</code>                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| <code>alpha.robust</code>       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| <code>...</code>                | Additional parameters.   |

**Value**

Influence function of Omega Ratio.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.Omega(returns=NULL, evalShape=TRUE,
                 retVals=NULL, nuisPars =NULL,
                 IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.Omega(returns=edhec[, "CA"], evalShape=TRUE,
                 retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
                 IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.Omega(returns=edhec[, "CA"], evalShape=FALSE,
                 retVals=NULL, nuisPars =NULL,
                 IFplot=TRUE, IFprint=TRUE,
                 compile=TRUE, prewhiten=FALSE,
                 cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

IF.RachR

*Influence Function - Rachev Ratio***Description**

IF.Rachev returns the data and plots the shape of either the IF or the IF TS for the Rachev Ratio.

**Usage**

```
IF.RachR(returns = NULL, evalShape = FALSE, retVals = NULL,
        nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
        alpha = 0.1, beta = 0.1, rf = 0, prewhiten = FALSE,
        ar.prewhiten.order = 1, cleanOutliers = FALSE,
        cleanMethod = c("locScaleRob", "Boudt")[1], eff = 0.99,
        alpha.robust = 0.05, ...)
```

**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| alpha              | Lower tail probability.  |
| beta               | Upper tail probability.  |
| rf                 | Risk-free interest rate.   |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of Rachev Ratio.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.RachR(returns=NULL, evalShape=TRUE,
                 retVals=NULL, nuisPars =NULL,
                 IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.RachR(returns=edhec[, "CA"], evalShape=TRUE,
                 retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
                 IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.RachR(returns=edhec[, "CA"], evalShape=FALSE,
                 retVals=NULL, nuisPars =NULL,
                 IFplot=TRUE, IFprint=TRUE,
                 prewhiten=FALSE,
                 cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

---

IF.SD

*Influence Function - Standard Deviation*

---

**Description**

IF.SD returns the data and plots the shape of either the IF or the IF TS for the standard deviation

**Usage**

```
IF.SD(returns = NULL, evalShape = FALSE, retVals = NULL,
      nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
      compile = TRUE, prewhiten = FALSE, ar.prewhiten.order = 1,
      cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
      eff = 0.99, alpha.robust = 0.05, ...)
```

**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| compile            | Boolean variable to indicate if the IF TS should be computed using compiled code (C++) (TRUE) or not (FALSE).                              |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of the standard deviation.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.SD(returns=NULL, evalShape=TRUE, retVals=NULL, nuisPars=NULL,
              IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.SD(returns=edhec[, "CA"], evalShape=TRUE,
```

```

retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.SD(returns=edhec[,"CA"], evalShape=FALSE,
retVals=NULL, nuisPars =NULL,
IFplot=TRUE, IFprint=TRUE,
compile=TRUE, prewhiten=FALSE,
cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)

```

---

IF . SoR

*Influence Function - Sortino Ratio*


---

### Description

IF . SoR returns the data and plots the shape of either the IF or the IF TS for the Sortino Ratio

### Usage

```

IF.SoR(returns = NULL, evalShape = FALSE, retVals = NULL,
nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
compile = TRUE, threshold = c("mean", "const")[1], const = 0,
rf = 0, prewhiten = FALSE, ar.prewhiten.order = 1,
cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
eff = 0.99, alpha.robust = 0.05, ...)

```

### Arguments

|           |  |
|-----------|--|
| returns   | Vector of the returns of the asset or portfolio.   |
| evalShape | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed. |
| retVals   | Values used to evaluate the shape of the IF.   |
| nuisPars  | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).                       |
| k         | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot    | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint   | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| compile   | Boolean variable to indicate if the IF TS should be computed using compiled code (C++) (TRUE) or not (FALSE).          |
| threshold | Parameter of threshold is either "mean" or "const". Default is "mean".   |
| const     | The threshold if threshold is "const".   |
| rf        | Risk-free interest rate.   |
| prewhiten | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).                                       |

|                                 |  |
|---------------------------------|--|
| <code>ar.prewhiten.order</code> | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| <code>cleanOutliers</code>      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. |
| <code>cleanMethod</code>        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                |
| <code>eff</code>                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.                          |
| <code>alpha.robust</code>       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm.   |
| <code>...</code>                | Additional parameters.   |

**Value**

Influence function of SoR\_C.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.SoR(returns=NULL, evalShape=TRUE,
               retVals=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.SoR(returns=edhec[, "CA"], evalShape=TRUE,
               retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.SoR(returns=edhec[, "CA"], evalShape=FALSE,
               retVals=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE,
               compile=TRUE, prewhiten=FALSE,
               cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```



---

IF.SR *Influence Function - Sharpe Ratio (SR)*

---

**Description**

IF.SR returns the data and plots the shape of either the IF or the IF TS for the SR

**Usage**

```
IF.SR(returns = NULL, evalShape = FALSE, retVals = NULL,
      nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE, rf = 0,
      compile = TRUE, prewhiten = FALSE, ar.prewhiten.order = 1,
      cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
      eff = 0.99, alpha.robust = 0.05, ...)
```

**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| rf                 | Risk-free interest rate.   |
| compile            | Boolean variable to indicate if the IF TS should be computed using compiled code (C++) (TRUE) or not (FALSE).                              |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of the SR.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.SR(returns=NULL, evalShape=TRUE,
              retVals=NULL, nuisPars =NULL,
              IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.SR(returns=edhec[, "CA"], evalShape=TRUE,
              retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
              IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.SR(returns=edhec[, "CA"], evalShape=FALSE,
              retVals=NULL, nuisPars =NULL,
              IFplot=TRUE, IFprint=TRUE,
              compile=TRUE, prewhiten=FALSE,
              cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

---

IF.SSD

*Influence Function - Semi-Standard Deviation (SSD)*

---

**Description**

IF.SSD returns the data and plots the shape of either the IF or the IF TS for the SSD

**Usage**

```
IF.SSD(returns = NULL, evalShape = FALSE, retVals = NULL,
       nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE, rf = 0,
       compile = TRUE, prewhiten = FALSE, ar.prewhiten.order = 1,
       cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
       eff = 0.99, alpha.robust = 0.05, ...)
```



```

# Plot of IF a specified TS
outIF <- IF.SSD(returns=edhec[,"CA"], evalShape=TRUE,
               retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.SSD(returns=edhec[,"CA"], evalShape=FALSE,
               retVals=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE,
               compile=TRUE, prewhiten=FALSE,
               cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)

```

---

IF.VaR

*Influence Function - Value at Risk (VaR)*


---

### Description

IF.VaR returns the data and plots the shape of either the IF or the IF TS for the Value at Risk

### Usage

```

IF.VaR(returns = NULL, evalShape = FALSE, retVals = NULL,
       nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
       alpha = 0.05, prewhiten = FALSE, ar.prewhiten.order = 1,
       cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
       eff = 0.99, alpha.robust = 0.05, ...)

```

### Arguments

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.       |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).                             |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| alpha              | The tail probability of interest.  |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter. |

|                           |  |
|---------------------------|--|
| <code>cleanMethod</code>  | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| <code>eff</code>          | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| <code>alpha.robust</code> | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| <code>...</code>          | Additional parameters.   |

**Value**

Influence function of the VaR.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.VaR(returns=NULL, evalShape=TRUE,
               retValS=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.VaR(returns=edhec[, "CA"], evalShape=TRUE,
               retValS=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.VaR(returns=edhec[, "CA"], evalShape=FALSE,
               retValS=NULL, nuisPars =NULL,
               IFplot=TRUE, IFprint=TRUE,
               prewhiten=FALSE,
               cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

---

IF.VaRratio

*Influence Function - Value at Risk (VaR) Ratio*

---

**Description**

IF.VaRratio returns the data and plots the shape of either the IF or the IF TS for the VaR Ratio.

**Usage**

```
IF.VaRratio(returns = NULL, evalShape = FALSE, retVals = NULL,
  nuisPars = NULL, k = 4, IFplot = FALSE, IFprint = TRUE,
  alpha = 0.05, rf = 0, prewhiten = FALSE, ar.prewhiten.order = 1,
  cleanOutliers = FALSE, cleanMethod = c("locScaleRob", "Boudt")[1],
  eff = 0.99, alpha.robust = 0.05, ...)
```

**Arguments**

|                    |  |
|--------------------|--|
| returns            | Vector of the returns of the asset or portfolio.   |
| evalShape          | Evaluation of the shape of the IF risk measure if TRUE. Otherwise, a TS of the IF of the provided returns is computed.                     |
| retVals            | Values used to evaluate the shape of the IF.   |
| nuisPars           | Nuisance parameters used for the evaluation of the shape of the IF (if no returns are provided).   |
| k                  | Range parameter for the shape of the IF (the SD gets multiplied k times).  |
| IFplot             | If TRUE, the plot of the IF shape or IF TS of the returns is produced.   |
| IFprint            | If TRUE, the data for the IF shape or the IF TS of the returns is returned.  |
| alpha              | The tail probability of interest.  |
| rf                 | Risk-free interest rate.   |
| prewhiten          | Boolean variable to indicate if the IF TS is pre-whitened (TRUE) or not (FALSE).   |
| ar.prewhiten.order | Order of AR parameter for the pre-whitening. Default is AR(1).   |
| cleanOutliers      | Boolean variable to indicate whether the pre-whitening of the influence functions TS should be done through a robust filter.               |
| cleanMethod        | Robust method used to clean outliers from the TS. The choices are "Boudt" and "locScaleRob" for the function.                              |
| eff                | Tuning parameter for the normal distribution efficiency for the "locScaleRob" robust data cleaning.  |
| alpha.robust       | Tuning parameter for the quantile of the "Boudt" robust data cleaning algorithm, using the minimum covariance determinant estimator (MCD). |
| ...                | Additional parameters.   |

**Value**

Influence function of the VaRratio.

**Author(s)**

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

**Examples**

```
# Plot of IF with nuisance parameter with return value
outIF <- IF.VaRratio(returns=NULL, evalShape=TRUE,
                    retVals=NULL, nuisPars =NULL,
                    IFplot=TRUE, IFprint=TRUE)

data(edhec, package="PerformanceAnalytics")
colnames(edhec) = c("CA", "CTAG", "DIS", "EM", "EMN", "ED", "FIA",
                  "GM", "LS", "MA", "RV", "SS", "FoF")

# Plot of IF a specified TS
outIF <- IF.VaRratio(returns=edhec[, "CA"], evalShape=TRUE,
                    retVals=seq(-0.1, 0.1, by=0.001), nuisPars =NULL,
                    IFplot=TRUE, IFprint=TRUE)

# Computing the IF of the returns (with outlier cleaning and prewhitening) with a plot of IF TS
outIF <- IF.VaRratio(returns=edhec[, "CA"], evalShape=FALSE,
                    retVals=NULL, nuisPars =NULL,
                    IFplot=TRUE, IFprint=TRUE,
                    prewhiten=FALSE,
                    cleanOutliers=TRUE, cleanMethod=c("locScaleRob", "Boudt")[1], eff=0.99)
```

---

nuisParsFn

*Nuisance Parameters Computation*


---

**Description**

nuis.pars returns the value of the nuisance parameters used in the evaluation of the shape of influence functions for risk and performance measures.

**Usage**

```
nuisParsFn(mu = 0.01, sd = 0.05, c = 0, alpha = 0.1, beta = 0.1)
```

**Arguments**

|       |   |
|-------|---|
| mu    | Mean parameter.                         |
| sd    | Standard deviation parameter.           |
| c     | Constant value for threshold.           |
| alpha | Parameters for the lower tail quantile. |
| beta  | Parameter for the upper tail quantile.  |

**Value**

List of nuisance parameters.

**Author(s)**

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

**Examples**

```
# Nuisance parameters using default values
defaultNuisance <- nuisParsFn ()
```

```
# Nuisance parameters using specified values
specifiedNuisance <- nuisParsFn (mu=0.02, sd=0.1, c=0.01, alpha=0.05, beta=0.1)
```



# Index

IF, [2](#)  
IF.ES, [4](#)  
IF.ESratio, [5](#)  
IF.LPM, [7](#)  
IF.mean, [8](#)  
IF.Omega, [10](#)  
IF.RachR, [12](#)  
IF.SD, [13](#)  
IF.SoR, [15](#)  
IF.SR, [17](#)  
IF.SSD, [18](#)  
IF.VaR, [20](#)  
IF.VaRatio, [21](#)  
  
nuisParsFn, [23](#)