Influence Functions for Risk and Performance Measures

Anthony Christidis, Shengyu Zhang, Doug Martin

December 11, 2018

Abstract

The `InfluenceFunctions` package implements the influence functions for risk and performance measures for assets or portfolios returns. Some of the influence functions included for the risk and performance measures are the mean, standard deviation, value at risk, expected shortfall, Sharpe ratio, Sortino ratio, and many more. The influence function of a risk or performance measure is useful in obtaining an estimate of the standard error for this same measure when the returns are correlated, as described in Chen and Martin [2018]. This `InfluenceFunctions` package is used in the `EstimatorStandardError` to compute the influence functions time series for correlated returns to obtain the standard errors.

1 Theoretical Background - Influence Functions

1.1 Risk and Performance Estimator Functional Representations

The large-sample value (as sample size n tends to infinity) of any risk and performance estimators may be represented as a functional $T = T(F)$ of the marginal distribution function F of the returns. And the finite sample estimate $T_n = T(F_n) = T(r_1, r_2, \ldots, r_n)$ may be obtained by evaluating the functional at the empirical distribution F_n that has a jump of height $1/n$ at each of the observed returns values $r_1, r_2, \ldots, r_n$.

For example, the sample mean and samle volatility have the large sample functional representations:

$$\mu(F) = \int r dF(r) \tag{1}$$

$$\sigma(F) = \left[\int (r - \mu(F))^2 dF(r) \right]^{\frac{1}{2}} \tag{2}$$

and the finite sample estimators are the sample mean

$$\hat{\mu}_n = \frac{1}{n} \sum_{t=1}^{n} r_t \tag{3}$$

and sample volatility

$$\sigma(F) = \left[\int (r - \mu(F_n))^2 dF_n(r) \right]^{\frac{1}{2}} = \left[\frac{1}{n} \sum_{t=1}^{n} (r_t - \hat{\mu}_n) \right]^{\frac{1}{2}} \tag{4}$$

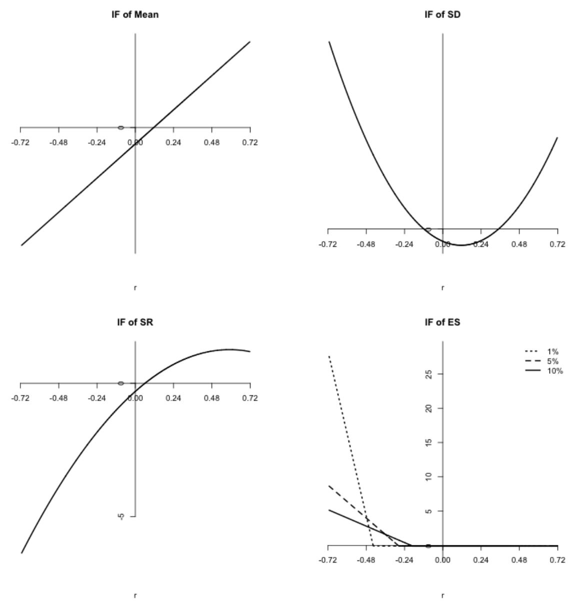


Figure 1: Influence Functions of Some Risk and Performance Measures Chen and Martin [2018].

1.2 Risk and Performance Measure Estimator Influence Functions

Influence Functions are based on the use of the following mixture distribution perturbation of a fixed target distribution $F(x)$:

$$F_{\gamma}(x) = (1-\gamma)F(x) + \gamma\delta_r(x) \tag{5}$$

where $\delta_r(x)$ is a point mass discrete distribution function with a jump of height one located at return value r . Then the influence function of a risk of performance measure estimator is defined through its functional representation $T(F)$ as:

$$IF(r;T,F) = \lim_{\gamma\to 0}\frac{T(F_{\gamma}) - T(F)}{\gamma} = \frac{d}{d\gamma}T(F_{\gamma})|_{\gamma=0} \tag{6}$$

The influence function is a special directional derivative (i.e., a Gateaux derivative) of the functional T on an infinite dimensional space of distribution function, in the direction of point mass distributions δ_r evaluated at F.

It is straightforward, though sometimes tedious, to derive formulas for influence functions for risk and performance measures, and make plots of them. The formulas for influence functions for sample mean (MEAN), standard deviation (SD), Sharpe ratio (SR) and expected short fall (ES) estimators in Figure 1 are derived in Chen and Martin [2018] for the case of montly returns with mean 0.12 and standard deviation (volatility) 0.24. Note that more derivations for the other risk and performance measures may be found in Zhang [2009].

Similarly, Figure 2 shows the influence function time series for Convertible Arbitrage (CA) hedge fund monthly returns from 1997/01 to 2009/08. Note that all of these time series have outliers around the late-

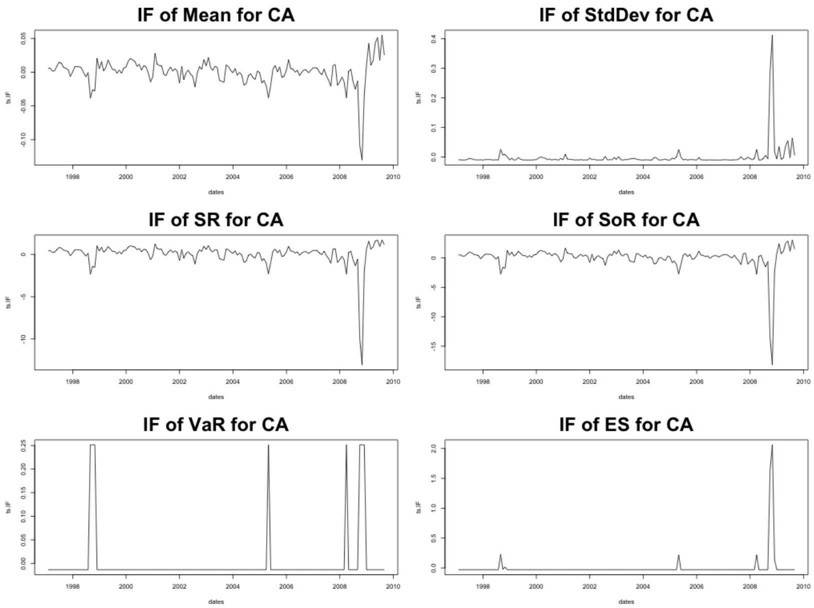


Figure 2: Influence Functions Time Series of CA Hedge Fund Returns Chen and Martin [2018]

2008 period following the market crash of that year. An extension to this work and the `InfluenceFunctions` package that will be discussed later is to implement some form of filtering mechanism for these outliers, to minimize their effects on the fitted model. References on these topics may be found in Peterson et al. [2007] and Martin et al. [1979] for outlier multivariate and univariate filtering, respectively.

1.3 The Key Influence Function Property

The key influence function property, which is used in the `EstimatorStandardError` package, is that for well behaved estimator functionals, the difference between the finite-sample estimator $T(F_n)$ and its asymptotic value $T(F)$ can be expressed as the following linear combination of influence functions of the returns at each point of time:

$$T(F_n) - T(F) = \frac{1}{n}\sum_{t=1}^{n} IF(r_t; T, F) + remainder \tag{7}$$

where the remainder goes to zero as $n \to \infty$ in a probablistic sense. Ignoring the relatively ignorable remainder term, the variance of the estimator $T(F_n)$ is the variance of the series on the right-hand side of the above equation, namely:

$$Var_n[T(F_n)] = \frac{1}{n}Var[IF(r_1; T, F)] = \frac{1}{n}E[IF^2(r_1; T, F)] \tag{8}$$

and the expression on the right-hand side can be evaluated empirically.

However, when the $r_t, t = 1, 2, \ldots, n$ are serially correlated, one makes use of the fact that the variance in equation (8) is given approximately by computing the spectral density function $S_{IF_t}(f)$ of the influence functions time series $IF_t = IF(r_t; T, F), t = 1, 2, \ldots, n$ evaluated at frequency zero, i.e., $S_{IF_t}(f)|_{f=0}$. Further details on this topic may be found in Chen and Martin [2018], as the `InfluenceFunctions` package encompasses solely the computation of the influence functions for various risk and performane measures.

2 Formulas for Influence Functions of Risk and Performance Measures

For the derivations of the influence functions of all the risk and performance measures in this section, refer to Zhang [2009], Martin and Zhang [2017] and Chen and Martin [2018].

2.1 Mean - Functional and Influence Function

The functional representation for the mean of the returns is:

$$\mu(F) = \int r dF(r) \tag{9}$$

The formula for the influence function of the mean (with $\mu = \mu(F)$ for notational convenience) is:

$$IF(r; \mu; F) = r - \mu \tag{10}$$

The computation of the influence function of the mean is available via the function `IF.mean` in the `InfluenceFunctions` package.

2.2 Standard Deviation - Functional and Influence Function

The functional representation for the SD of the returns is:

$$\sigma(F) = \left[\int (r - \mu(F))^2 dF(r) \right]^{\frac{1}{2}} \tag{11}$$

The formula for the influence function of the SD (with $\sigma = \sigma(F)$ for notational convenience) is:

$$IF(r; \sigma; F) = (2\sigma)^{-1}((r - \mu)^2 - \sigma^2) \tag{12}$$

The computation of the influence function of the mean is available via the function `IF.SD` in the `InfluenceFunctions` package.

2.3 Sharpe Ratio - Functional and Influence Function

The functional representation for the SR of the returns is:

$$SR(F) = \frac{\mu(F) - r_f}{\sigma(F)} \tag{13}$$

The formula for the influence function of the SR (with $SR = SR(F)$ for notational convenience) is:

$$IF(r; SR; F) = -\frac{SR}{2\sigma^2}(r - \mu)^2 + \frac{1}{\sigma}(r - \mu) + \frac{SR}{2} \tag{14}$$

The computation of the influence function of the mean is available via the function `IF.SR` in the `InfluenceFunctions` package.

2.4 Value at Risk Ratio - Functional and Influence Funfction

Let the Value-at-Risk (defined as $VaR_\alpha(F)$ for a confidence level α) of the returns be defined as:

$$VaR_\alpha(F) = -infr|F(r) \geq \alpha| \tag{15}$$

The functional representation for the VaR Ratio of the returns is:

$$VR(F) = -\frac{\mu(F) - r_f}{VaR_\alpha(F)} \tag{16}$$

The formula for the influence function of the VaR (with $VR_\alpha = VR_\alpha(F)$ and $VaR_\alpha(F) = VaR_\alpha$ for notational convenience) is:

$$IF(r; SR; F) = -\frac{r-\mu}{VaR_\alpha} - \frac{VR_\alpha}{VaR_\alpha} \times \frac{I(r \leq -VaR_\alpha) - \alpha}{f(-VaR_\alpha)} \tag{17}$$

The computation of the influence function of the mean is available via the function `IF.VaR` in the `InfluenceFunctions` package.

2.5 Expected Shortfall - Functional and Influence Function

The functional representation for the ES of the returns is:

$$ES_\alpha(F) = -\frac{1}{\alpha}\int_{-\infty}^{q_\alpha(F)} rdF(r) \tag{18}$$

The formula for the influence function of the ES (with $ES_\alpha = ES_\alpha(F)$ for notational convenience) is:

$$IF(r; SR; F) = -\frac{r - q_\alpha}{\alpha} I(r \leq q_{alpha}) + q_\alpha + ES_\alpha \tag{19}$$

The computation of the influence function of the mean is available via the function `IF.ES` in the `InfluenceFunctions` package.

2.6 Semi-Standard Deviation (Mean Threshold) - Functional and Influence Function

The functional representation for the SSD (with mean threshold) of the returns is:

$$\sigma_{SSD}(F) = \sqrt{\int_{-\infty}^{\mu(F)} (r - \mu(F))^2 dF(r)} \tag{20}$$

The formula for the influence function of the SSD (with $\sigma_{SSD}(F) = \sigma_{SSD}$ for notational convencience) is:

$$IF(r; \sigma_{SSD}; F) = \frac{(r-\mu)^2 \times I(r \leq \mu) - 2 \times \int_{-\infty}^{\mu(F)} (r - \mu(F))dF(r) \times (r - \mu) - \sigma_{SSD}}{2 \times \sigma_{SSD}} \tag{21}$$

The computation of the influence function of the mean is available via the function `IF.SSD` in the `InfluenceFunctions` package.

2.7 Sortino Ratio (Mean Threshold) - Functional and Influence Function

The functional representation for the SortR (with mean threshold) of the returns is:

$$SortR(F) = \frac{\mu(F) - r_f}{\sigma_{SSD}(F)} \tag{22}$$

The formula for the influence function of the SortR (with mean threshold and with $SortR(F) = SortR$ for notational convenience) of the returns is:

$$IF(r; SortR; F) = \frac{-SortR}{2\sigma^c_{SSD}}(r-\mu)^2 I(r \leq \mu) + \frac{1}{\sigma^c_{SSD}}(r-\mu) + \frac{SortR\int_{-\infty}^{\mu}(r-\mu)dF(r)}{\sigma_{SSD}(F)}(r-\mu) + \frac{SortR}{2} \tag{23}$$

The computation of the influence function of the mean is available via the function `IF.SoR.mean` in the `InfluenceFunctions` package.

2.8 Sortino Ratio (Constant Threshold) - Functional and Influence Function

Let the functional representation of the SSD (with constant threshold) of the returns be:

$$\sigma^c_{SSD}(F) = \sqrt{\int_{-\infty}^{c}(r-c)^2 dF(r)} \tag{24}$$

The functional representation for the SortR (with constant threshold) of the returns is:

$$SortR(F) = \frac{\mu(F) - r_f}{\sigma^c_{SSD}(F)} \tag{25}$$

The formula for the influence function of the SortR (with constant threshold and with $SortR(F) = SortR$ for notational convenience) of the returns is:

$$IF(r; SortR; F) = \frac{-SortR}{2\sigma^c_{SSD}}(r-c)^2 I(r \leq c) + \frac{1}{\sigma^c_{SSD}}(r-\mu) + \frac{SortR}{2} \tag{26}$$

The computation of the influence function of the mean is available via the function `IF.SoR.const` in the `InfluenceFunctions` package.

2.9 STARR Ratio - Functional and Influence Function

The functional representation for the STARR Ratio of the returns is:

$$STARR(F) = \frac{\mu(F) - r_f}{ES_\alpha(F)} \tag{27}$$

The formula for the influence function of the STARR (with $STARR(F) = STARR$ for notational convenience) of the returns is:

$$IF(r; STARR; F) = \frac{r-\mu}{ES_\alpha} - \frac{STARR}{ES_\alpha}\left(\frac{-r - VaR_\alpha}{\alpha} I(r \leq -VaR_\alpha) + VaR_\alpha - ES_\alpha\right) \tag{28}$$

The computation of the influence function of the mean is available via the function `IF.STARR` in the `InfluenceFunctions` package.

2.10 Rachev Ratio - Functional and Influence Function

Define the expected tail gain to be the following:

$$CVaR_{\beta}^{+}(F) = \frac{1}{\beta}\int_{CVaR_{\beta}^{+}(F)}^{+\infty} rdF(r) \tag{29}$$

where $VaR_{\beta}^{+}(F)$ is the upper β -quantile of gain defined by the following equation:

$$VaR_{\beta}^{+}(F) = supr|F(r) \leq 1-\beta| \tag{30}$$

The functional representation for the Rachev Ratio of the returns is:

$$RachR(F) = \frac{ES_{\alpha}(F)}{CVaR_{\beta}^{+}(F)} \tag{31}$$

The formula for the influence function of the Rachev Ratio (with $RachR(F) = RachR$ for notational convenience) of the returns is:

$$\begin{aligned} IF(r; RachR; F) = \frac{1}{ES_{\alpha}}&\left(\frac{1}{\beta}(r - VaR_{\beta}^{+})I(r \leq -VaR_{\beta}^{+}) + VaR_{\beta}^{+} - CVaR_{\beta}^{+}\right) \\ &- \frac{CVaR_{\beta}^{+}}{(ES_{\alpha})^2}\left(\frac{-r - VaR_{\alpha}}{\alpha}I(r \leq -VaR_{\alpha}) + VaR_{\alpha} - ES_{\alpha}\right) \end{aligned} \tag{32}$$

The computation of the influence function of the mean is available via the function `IF.Rachev` in the `InfluenceFunctions` package.

2.11 Lower Partial Moment Ratio - Functional and Influence Function

The functional representation for the LPM of the returns is:

$$\mu_{k}^{-}(F) = \int_{-\infty}^{c} (r-c)dF(r) \tag{33}$$

The formula for the influence function of the LPM (with $\mu_{k}^{-}(F) = \mu_{k}^{-}$ for notational convenience) of the returns is:

$$IF(r; \mu_{k}^{-}; F) = (c-r)I(r \leq c) - \mu_{k}^{-} \tag{34}$$

The computation of the influence function of the mean is available via the function `IF.LPM` in the `InfluenceFunctions` package.

2.12 Omega Ratio - Functional and Influence Function

The functional representation for the Omega function of the returns is:

$$\Omega_{c}(F) = \frac{\Omega_{+}(F)}{\Omega_{-}(F)} = \frac{\int_{c}^{\infty}(1-F(r))dr}{\int_{-\infty}^{c}F(r)dr} = \frac{\int_{c}^{\infty}(r-c)dF(r)}{\int_{-\infty}^{c}(c-r)dF(r)} \tag{35}$$

The formula for the influence function of the Omega function (with $\Omega(F) = \Omega$ for notational convenience) of the returns is:

$$IF(r;\Omega;F) = \frac{1}{\Omega_-}\Big((r-c)I(r \geq c) - \Omega_+\Big) - \frac{\Omega_+}{\Omega_-^2}\Big((c-r)I(r \leq c) - \Omega_-\Big) \tag{36}$$

The computation of the influence function of the mean is available via the function `IF.OmegaRatio` in the `InfluenceFunctions` package.

3 Prewhitening and Robust Filtering

3.1 Prewhitening

Prewhitening is a widely used technique spectral density function estimation in the field of signal processing and other applications areas in engineering and science. Prewhitening can often improve the performance of estimators. Since the core of the method described in Chen and Martin [2018] is estimation of a spectral density at frequency zero of an influence function time series as accurately as possible, prewhitening is used to improve the accuracy of the `seCorIF` method in the `EstimatorStandardError` package.

The following model for prewhitening the IF_t time series is used:

$$IF_t^{pw} = IF_t - \hat{\rho} IF_{t-1} \tag{37}$$

where $\hat{\rho}$ is a lag-one serial correlation coefficient estimate. In general IFpw t is not an uncorrelated (white noise) series, but it has considerably less serial correlation than IF_t, and a periodogram estimator based on IF_t^{pw} will suffer from relatively little bias compared with one based on r_t.

3.2 Robust Filtering

Recall that influence functions time series for the risk and performance measure of Section 2 may have outliers, as in Figure 2 (late-2008 period following the market crash of that year). Ideally, these should be filtered our from the time series, as a generalized linear model computational method will be used to fit to the spectral density of the time series Chen and Martin [2018]. There are two methods for robust filtering implemented in the `InfluenceFunctions` package (see Section 4.2 for computation details), and references on these topics may be found in Peterson et al. [2007] and Martin and Zamar [1993].

4 Sample Code

4.1 Basic Functionality

First, to install the `InfluenceFunctions` package, one way run the following code:

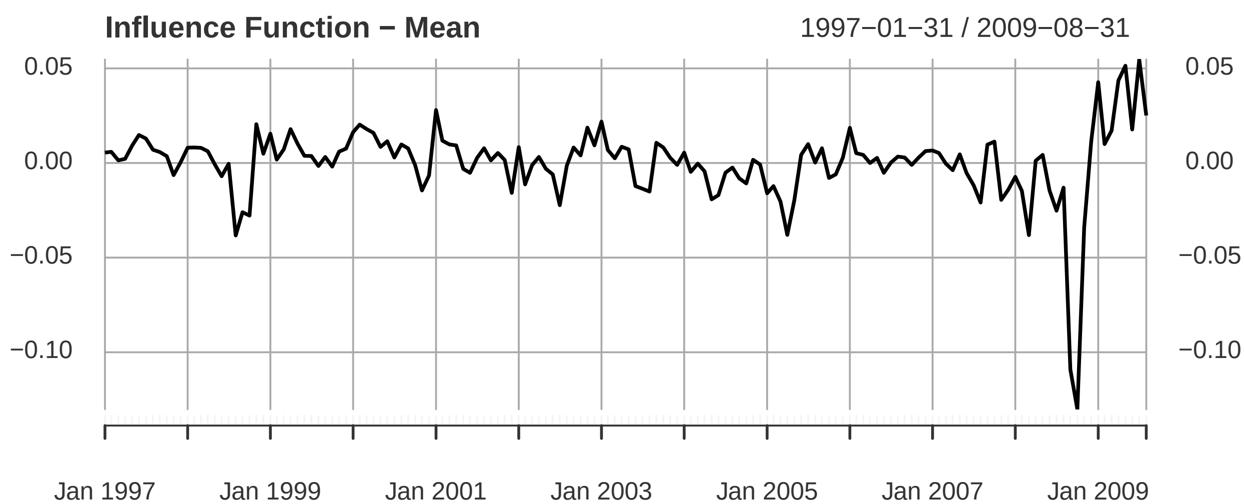
```
library(devtools)
install_github("AnthonyChristidis/InfluenceFunctions")
```

There is a data set available with the `InfluenceFunctions` package which contains the monthly returns of various hedge funds over a time period. Also, the Convertible Arbitrage fund has been extracted from this data set, as it is used as the default data for various functions of the package if the user does not provide any data.

```r
# Load IF library
library(InfluenceFunctions)

# edhec is an xts object with the returns of multiple hedge funds
data(edhec)
head(edhec)

## [1] 0.0119 0.0123 0.0078 0.0086 0.0156 0.0212

# CA.fund is an xts object with the returns for the CA fund
data(CA.fund)
head(CA.fund)

## [1] 0.0119 0.0123 0.0078 0.0086 0.0156 0.0212
```

To compute the influence function time series, one may use the `IF.mean`, `IF.SD`, `IF.SR`, `IF.ES`, `IF.VaR`, etc., functions on a vector (xts object or not) to compute the desired time series. For simplicity, we will use the influence function of the mean as an example throughout the remaining of this section. However, all the code is valid for any risk and performance measure included in the `InfluenceFunctions` package.

```r
# Computing the IF time series for the mean of CA hedge fund
if.vector <- IF.mean(CA.fund)
plot(if.vector, main = "Influence Function - Mean")
```




```r
# Note that we may use compiled code to compute the IF time series for
# computational efficiency and accuracy
if.vector <- IF.mean(CA.fund, compile = TRUE)

# We can also compute the desired influence functions time series via the
```

```r
# wrapper function IF
if.vector <- IF(CA.fund, risk = "mean")
```

Looking at the documentation for any of the influence functions via the following code:

```r
# Documentation of the IF function
`?`(IF.mean)
```

The influence functions currently available are listed in Section 2.

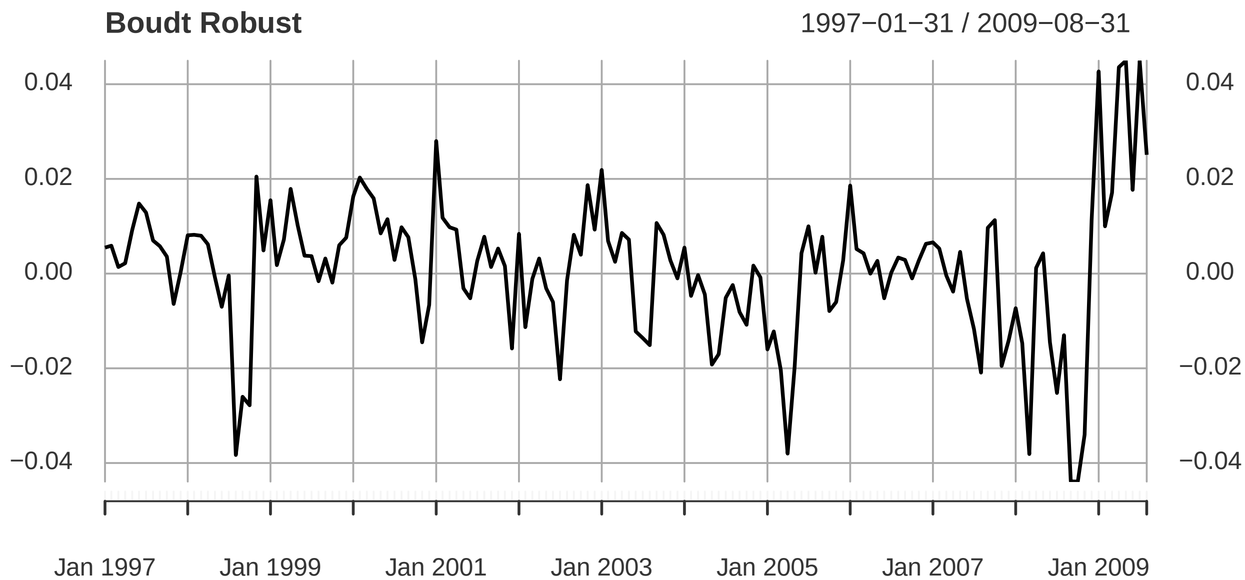
4.2 Prewhitening and Robust Filtering

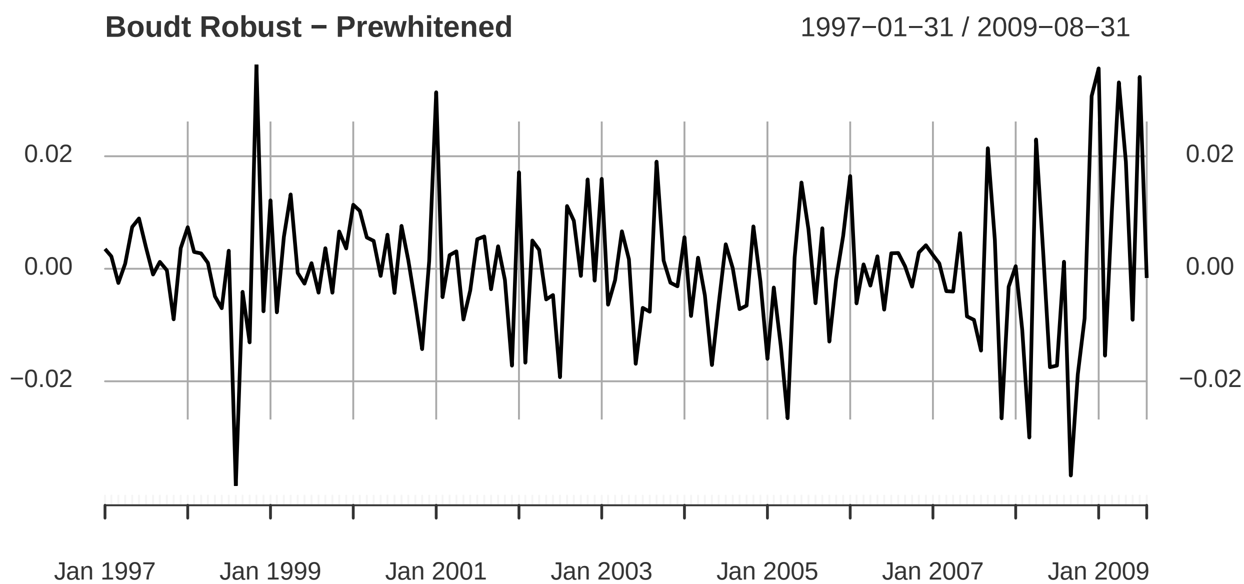
As described in Section 3.1, a prewhitened version of the influence function time series may be obtained via the option `pre.whiten`:

```r
# Computing the (prewhitened) IF time series for the mean of CA hedge fund
if.vector.prewhitened <- IF.mean(CA.fund, pre.whiten = TRUE)
```

A robust version of the influence function time series, using the *Boudt* method:

```r
# Computing the (robust) IF time series for the mean of CA hedge fund, Boudt
# method
if.vector.boudt <- IF(CA.fund, risk = "mean", pre.whiten = FALSE, robust.filtering = TRUE,
    robust.method = "Boudt", alpha.robust = 0.05)

# Computing the (robust and prewhitened) IF time series for the mean of CA
# hedge fund, Boudt method
if.vector.boudt.prewhitened <- IF(CA.fund, risk = "mean", pre.whiten = TRUE,
    robust.filtering = TRUE, robust.method = "Boudt", alpha.robust = 0.05)

# Plot the two time series
par(mfrow = c(2, 1))
plot(if.vector.boudt, main = "Boudt Robust")
plot(if.vector.boudt.prewhitened, main = "Boudt Robust - Prewhitened")
```

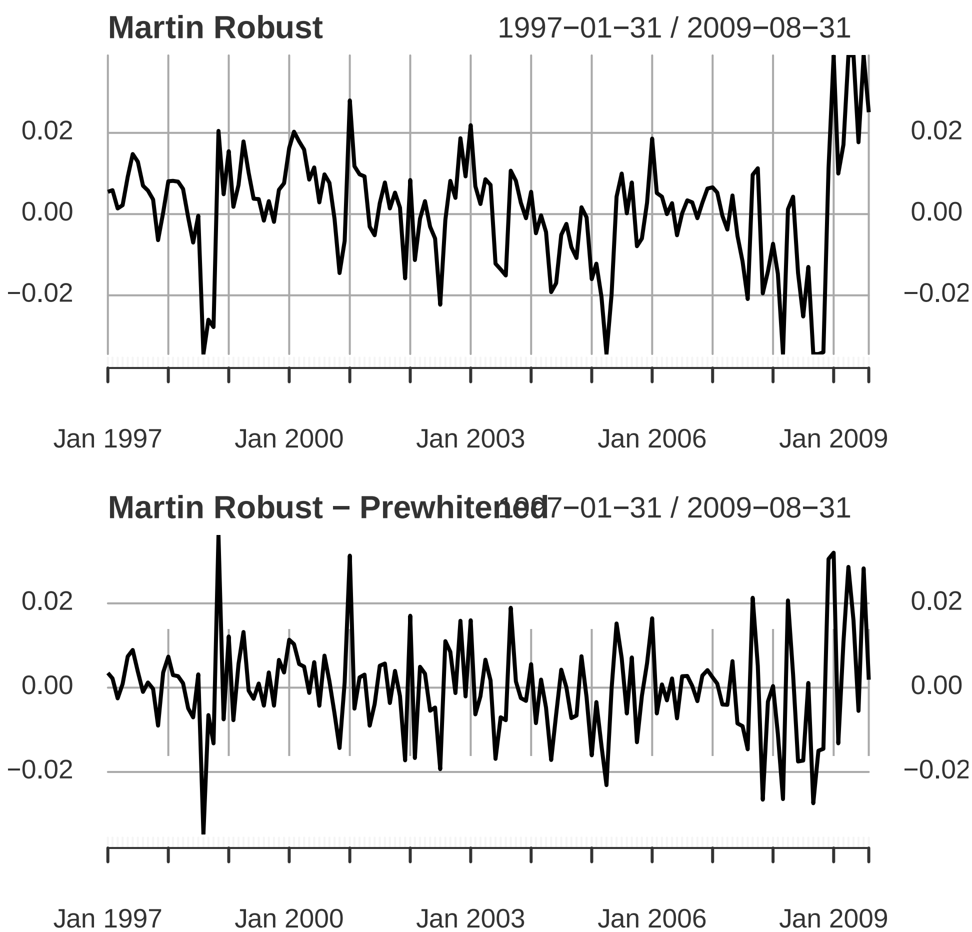




A robust version of the influence function time series, using the *Martin* method:

```
# Computing the (robust) IF time series for the mean of CA hedge fund,
# Martin method
if.vector.martin <- IF(CA.fund, risk = "mean", pre.whiten = FALSE, robust.filtering = TRUE,
    robust.method = "Martin", normal.efficiency = 0.99)
```

```r
# Computing the (robust and prewhitened) IF time series for the mean of CA
# hedge fund, Martin method
if.vector.martin.prewhitened <- IF(CA.fund, risk = "mean", pre.whiten = TRUE,
    robust.filtering = TRUE, robust.method = "Martin", normal.efficiency = 0.99)

# Plot the two time series
par(mfrow = c(2, 1))
plot(if.vector.martin, main = "Martin Robust")
plot(if.vector.martin.prewhitened, main = "Martin Robust - Prewhitened")
```



One could plot the influence functions for the various methods and compare them:

```r
# Plot of the IF TS for all methods
plot(if.vector, if.vector.boudt.prewhitened, col = "black", lwd = 2)
lines(if.vector.boudt.prewhitened, col = "red", lwd = 2)
lines(if.vector.martin.prewhitened, col = "blue", lwd = 2)
```

4.3 Visualization of Influence Functions Time Series via `ggplot2`

To obtain a formatted plot in `ggplot2` of an influence function time series, the following function is available:

```
# Plot method of the IF TS
plot.IF_TS(if.vector)
```

4.4 Functions to Compute the Influence Function of Risk and Performance Measures

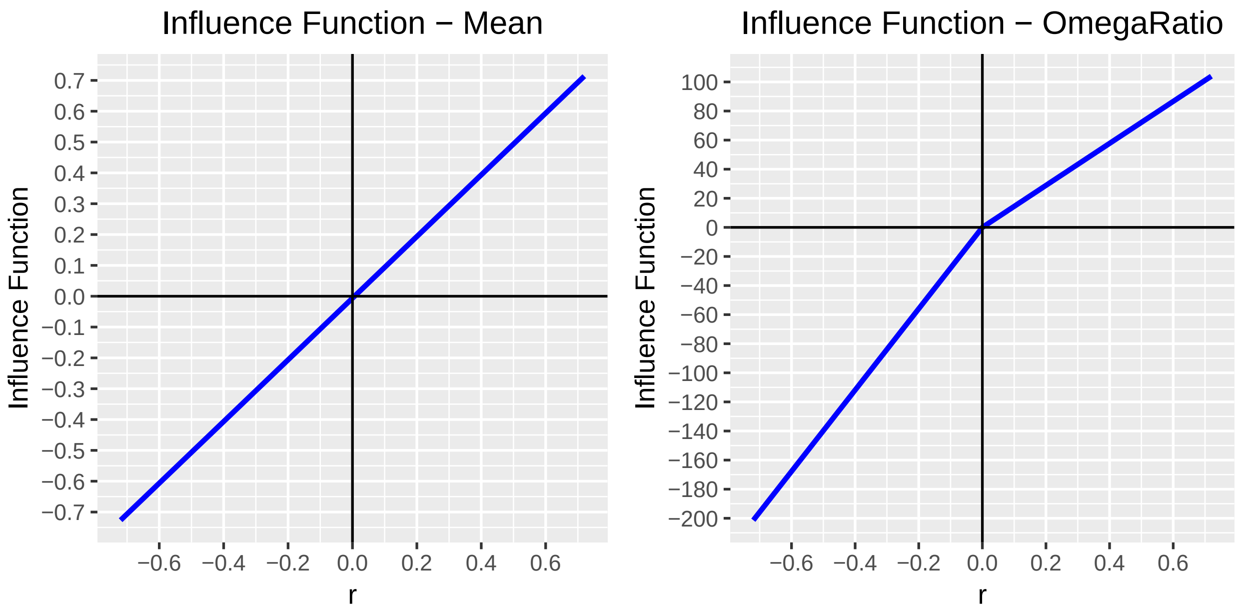
The functions `IF.mean.fn`, `IF.SD.fn`, `IF.SR.fn`, `IF.ES.fn`, `IF.VaR.fn`, etc. are available as well. Note that a vector of returns must be provided to compute some statistics for the influence function. By default, the CA fund is used.

```
# Plot of the IF for all risk measures or available
IF.mean.fn(0.02)

## [1] 0.01359145

IF.mean.fn(0.02, edhec$CTAG)

## [1] 0.01351053

# We can also use the wrapper function to compute the IF for the specified
# risk measure
IF.fn(0.02, edhec$CTAG, risk = "mean")

## [1] 0.01351053
```

4.5 Plots of Influence Functions for Risk and Performance Measures

The functions `IF.mean.plot`, `IF.SD.plot`, `IF.SR.plot`, `IF.ES.plot`, `IF.VaR.plot`, etc. are available to plot the influence function of the risk and performance measures included in the package. Note that a vector of returns must be provided to compute some statistics for the influence function. By default, the CA fund is used.

```
# Plot of the IF TS for various risk and performance measures We can also
# plot the IF TS for various risk and performance measures using the wrapper
# plot function
library(gridExtra)
grid.arrange(IF.mean.plot(), IF.plot(returns = edhec$CTAG, risk = "OmegaRatio"),
    ncol = 2)
```



References

X. Chen and R. D. Martin. Standard errors of risk and performance measure estimators for serially correlated returns. 2018. URL https://ssrn.com/abstract=3085672.

R. D. Martin and S. Zhang. Nonparametric versus parametric expected shortfall. 2017. URL https://ssrn.com/abstract=2747179.

R. D. Martin, B. Kleiner, and D. J. Thomson. Robust estimation of power spectra. *Journal of the Royal Statistical Society, Series B*, 1979.

R Douglas Martin and Ruben H Zamar. Bias robust estimation of scale. *The Annals of Statistics*, pages 991–1017, 1993.

Kris Peterson, Brian Boudt, and Christophe Croux. Estimation and decomposition of downside risk for portfolios with non-normal returns. 2007.

S. Zhang. Statistical analysis of portfolio risk and performance measures: the influence function approach. 2009.