GSOC 2018: PerformanceAnalytics Standard Errors

Submitted by Doug Martin

This Version: March 8, 2018.

1 Summary

The PerformanceAnalytics R package currently has approximately 50 functions for computing different risk and performance measure estimates based on assets and portfolios returns. Yet none of these functions compute standard errors (SE's) for the estimates. This project is focused on integrating the relatively new R package EstimatorStandardError into the PerformanceAnalytics package for the purpose of computing SE's for those approximately 50 risk and performance measure estimator functions. The resulting SE's will be accurate for non-normal as well as normal returns distributions, and for serially correlated as well as uncorrelated returns. The result will make the PerformanceAnalytics package's risk and performance measure estimators much more valuable by providing estimator uncertainty measures in the form of SE's.

2 Background

The foundations for this project were laid by Xin Chen's PhD dissertation on "Standard Errors for Risk and Performance Measure Estimators", due to be completed in September 2018 in the Department of Applied Mathematics at the University of Washington, with R. Douglas Martin as his PhD adviser. The method of computing standard errors is quite new and very general, being applicable in principle to any risk or performance measure estimators, and a detailed description may be found in the paper "Standard Errors of Risk and Performance Estimators for Serially Correlated Returns" by Chen et al. (2018), available at http://ssrn.com/abstract=3085672, and henceforth referred to as CM2018. The key elements of the method are as follows.

2.1 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 $T_n = T(F_n) = T(r_1, r_2, \cdots, r_n)$ 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 $r_1, r_2, \cdots, r_n$.

For example the sample mean and sample volatility have the large sample functional representations

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

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

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:

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

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

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

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 (SD), Sharpe ratio (SR) and expected short fall (ES) estimators in the figures below are derived in CM2018 for the case of monthly returns with mean 0.12 and standard deviation (volatility) 0.24.

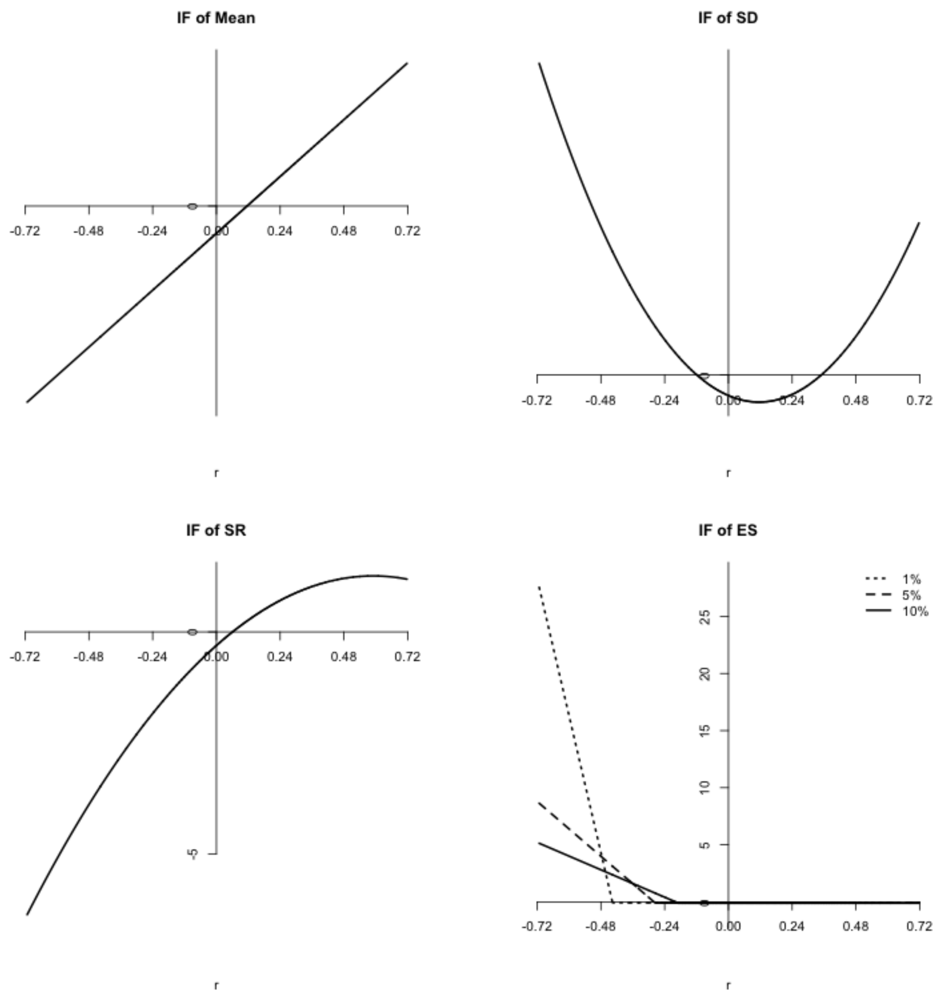

Figure 1: Influence Functions of Some Risk and Performance Measures

2.3 The Key Influence Function Property

The key influence function property for our method 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 probabilistic 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:

$$V_n(T(F_n)) = \frac{1}{n^2} \cdot var\left[\sum_{t=1}^{n} IF(r_t; T, F)\right]. \tag{8}$$

For independent and identically distributed returns r_t, $t = 1, 2, \cdots, n$, and using the fact that $E_F[IF(r_1; T, F)] = 0$, this expression reduces to

$$V_n(T(F_n)) = \frac{1}{n} var[IF(r_1; T, F)] = \frac{1}{n} E\left[IF^2(r_1; T, F)\right] \tag{9}$$

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

But when the r_t, $t = 1, 2, \cdots, n$ 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 $IF_t = IF(r_t; T, F)$, $t = 1, 2, \cdots, n$ evaluated at frequency zero, i.e., $S_{IF_t}(f)|_{f=0}$.

2.4 The Generalized Linear Model Computational Method with Elastic Net

The details of this important computational engine for obtaining standard errors of risk and performance measures are described in CM2018, and here is the basic idea. The above spectral density at frequency zero is computed by first computing the periodogram of the series IF_t, $t = 1, 2, \cdots, n$ with a fast-Fourier Transform (FFT) and then fitting a polynomial generalized linear model (GLM) to the periodogram values, where the GLM is based on the exponential distribution of the periodogram values. An added feature is that an elastic net variation of the the GFM is used to obtain a sparse representation of the polynomial approximation.

We refer to the overall method for computing standard errors described above as the "seCorIF" method, and the method is implemented in the R package "EstimatorStandardError", which is available at https://github.com/chenx26/EstimatorStandardError. This package requires the R package "glmnetRcpp" which implements the optimization method for fitting the GLM model with elastic net, and is available at https://github.com/chenx26/glmnetRcpp. For instructions on installing these packages, see Appendix A4 of CM2018.

3 Project Goals

The overall goal is to integrate the "EstimatorStandardError" (ESE) package with "PerformanceAnalytics" (PA) so as to compute standard errors (SE's) for all of the risk and performance measure estimators (RPME's) for which computation of SE's are appropriate, with the following as primary features: (1) The interface between ESE and the PA RPME functions shall be as consistent as is reasonably possible, with added arguments as needed to specify the SE output information, and (2) The user shall be able to specify SE output information content and format, to include the information in the columns of Table 3 in CM2018 or subsets thereof, among other to be determined at the beginning of the project in consultation with the project mentors.

3.1 Specification of PA RPME's

An initial goal will be to identify which of the approximately 50 RPME's, most of which are listed in the the table below, that will have ESE based SE computation provided, and document these as the "rpmeList". This list should include at least half of the existing PA RPME's as a minimal goal.

```
 [1] "AdjustedSharpeRatio"    "AverageDrawdown"
 [3] "BernardoLedoitRatio"    "BurkeRatio"
 [5] "CalmarRatio"            "CVaR"
 [7] "DownsideDeviation"      "DownsideFrequency"
 [9] "DownsidePotential"      "DRatio"
[11] "DrawdownDeviation"      "ES"
[13] "ETL"                    "Frequency"
[15] "kurtosis"               "MartinRatio"
[17] "maxDrawdown"            "mean.geometric"
[19] "mean.LCL"               "mean.stderr"
[21] "mean.UCL"               "MeanAbsoluteDeviation"
[23] "Omega"                  "OmegaSharpeRatio"
[25] "PainIndex"              "PainRatio"
[27] "Return.annualized"      "Return.cumulative"
[29] "sd.annualized"          "sd.multiperiod"
[31] "SemiDeviation"          "SemiVariance"
[33] "SharpeRatio"            "SharpeRatio.annualized"
[35] "skewness"               "SkewnessKurtosisRatio"
[37] "SmoothingIndex"         "SortinoRatio"
[39] "StdDev"                 "StdDev.annualized"
[41] "SterlingRatio"          "UPR"
[43] "UpsideFrequency"        "UpsidePotentialRatio"
[45] "UpsideRisk"             "VaR"
[47] "VolatilitySkewness"
```

Figure 2: Approximate List of RPME's in Current Version of PerformanceAnalytics

3.2 Influence Functions Package

The SE method relies critically on analytic formulas for the influence functions of estimators, i.e., the $IF_t = IF(r_t; T, F)$ in Section 2.3, in the "rpmeList". Thus an important goal to develop an R package that includes functions to evaluate all those formulas at returns r_t, including complete documentation and a vignette.

NOTE: Formulas for the influence functions of some of the risk and performance measures may be found in CM2018 and in Martin et al. (2017), and many more may be found in Zhang (2009). Influence function formulas that are needed, but not found in those references, will be developed by the mentors in collaboration with student selected for this project.

3.3 ESE and PA Interface Design

Different functions in the rpmeList have been written by different individuals, and so will not have completely consistent arguments, argument styles and functionality. Thus it will be important to spend considerable initial time in designing a style for interfacing the ESE capabilities into PA.

4 Milestones

Phase 1

Study the functionality of all the RPME's in the current version of PA and create an initial "rpmeList" for approval by Mentors.

Phase 2

(a) Development of the "InfluenceFunctions" package, and (b) Specification of the ESE and PA integration interface. To be done concurrently, starting with an "InfluenceFunctions" package for about 10 initial estimators.

Phase 3

Complete integration of SE methods for estimators in the "rpmeList", including testing, and documentation in the form of a high quality vignette and any needed PA manual update.

5 Expected Impact

The current finance industry practice in reporting risk and performance measure estimate typically does not include reporting SE's. For example one seldom sees SE's reported for Sharpe ratios, and consumers of these reports will have no clue as to how accurate those estimates are, and no clue whether or not two Sharpe ratios for two different portfolio products are significantly different. The result of our proposed project will deliver SE's that are accurate for both i.i.d. and serially correlated returns and for non-normal as well as normal returns, and available for a wide range risk and performance measure estimators. There will then be no excuse for providers of asset and portfolio performance reports to continue failing to report SE's.

6 Skills Required

- Extensive experience in use of R and writing R code
- Demonstrable experience in use of github
- Knowledge of R package development, and in particular solid knowledge of this as presented in the book R Packages (2015) by Hadley Wickham
- Highly desired: MS degree level quantitative finance knowledge
- Ability to read and understand the basic methodologies in the Martin et al. (2017) paper, available at https://ssrn.com/abstract=2747179.

7 Project Proposal Requirements

The successful applicant will:

1. Define the ESE and PA integration intended functionality, including the rpmeList, and integration style as illustrated for example by the Sharpe ratio (SR) and Expected Shortfall (ES) estimator SE's.
2. Demonstrate R package writing ability by creating an initial InfluenceFunctions R package that includes influence functions for the following four estimators: mean returns, volatility, Sharpe ratio, and Sortino ratio.

3. Write a detailed development timeline for code implementation, documentation and testing.
4. Identify any personal commitments that conflicts for their time during summer 2018.

8 Mentors

- R. Douglas Martin, Professor Emeritus, Departments of Applied Mathematics and Statistics, University of Washington. Founder and former Director of the UW Computational Finance and Risk Management MS Degree program. doug@amath.washington.edu.
- Xin Chen, PhD student, Department of Applied Mathematics, University of Washington. Creator and Maintainer of the R packages "EstimatorStandardError" and "glmnetRcpp". Planned PhD completion date: September 2018. chenx26@uw.edu.

9 Collaboration

During this project we anticipate close collaboration with the following individuals:

- Brian Peterson and Peter Carl: Primary developers and maintainer (Peterson) of PerformanceAnaltyics. Their input and guidance will be important in the designing the interface between the ESE and PA packages.
- Shengyu Zhang: Co-Author of Martin et al. (2017) who has done extensive research on influence functions for risk and performance measures estimators as documented in Zhang (2009), PH.D., CFA, FRM First Vice President Quantitative Manager HomeStreet Bank, Seattle, WA 98101. syzhang0401@gmail.com

References

Chen, X. and Martin, R. D. (2018). "Standard Errors of Risk and Performance Measure Esetimators for Serially Correlated Returns". *Available at http://ssrn.com/abstract=3085672ï¿œ* (cit. on p. 2).

Martin, R. D. and Zhang, S. (2017). "Nonparametric Versus Parametric Expected Shortfall (September 10, 2017)". *Available at SSRN: https://ssrn.com/abstract=2747179* (cit. on pp. 7–9).

Zhang, S. (2009). *Statistical Analysis of Portfolio Risk and Performance Measures: the Influence Function Approach*. University of Washington (cit. on pp. 7, 9).