GSoC 2017 Project Proposal

April 2, 2017

Project Information

Project Title: Advanced factorAnalytics Package

Project Short Title (30 characters) : Advancing factorAnalytics

Short Description:

This proposal explains the key deliverables and improvements in the functionality, usability, and graphics to the Fundamental Factor Model (FFM), the Time Series Factor Model (TSFM), and the Risk Management and Risk Budgeting (RM&RB) together with the timeline and detailed plan each week. The main ideas follow from the project description and there are my own ideas added based on those main ideas.

Project URL:

https://github.com/rstats-gsoc/gsoc2017/wiki/advancing-factorAnalytics

Bio of Student

My name is Chindhanai Uthaisaad, a second year Master's student in Computational Finance and Risk Management (CF&RM) at the University of Washington with Bachelor's degree in Mathematics and Computer Science from Chulalongkorn University, Thailand. I am currently a research assistant in Applied Mathematics at University of Washington under the supervision of Professor Douglas Martin in the large sample skew-t distribution MLEs, in which we explore the asymptotic properties of the 300 stocks monthly returns assuming skew-t distribution. My research/career interests include Asymptotic Statistics, Quantitative Research in Risk Analysis, Portfolio Optimization, and Factor Model Analysis.

Professor Martin amazed me several times with the use of factorAnalytics package in R to illustrate how powerful this package was, and how we could improve its performance in the factor model analysis, which is the essential of the portfolio management. I also heard about the GSoC summer project from Avinash Archaya, my classmate and the student who was awarded the GSoC 2016 project and developed the factorAnalytics package last year.

Regarding to my work experiences, I was a model risk analyst intern at Federal Home Loan Bank of Des Moines. I worked in multiple projects with Market Risk and Portfolio Strategy teams to identify and gauge risks in the models used by the Bank, and it has trained me to work under pressure within the given deadline, while not compromising in the quality of the model risk reports. During this internship, I developed my SQL and VBA proficiency in extracting and analyzing model risks in the Bank, together with time management in dealing multiple projects at a time. After that, I came back to UW as a Teaching Assistant in the Financial data access and analysis with SQL, Excel and VBA, a graduate class in CF&RM program at UW, in the first quarter of my second year. Even though I mainly used SQL and VBA throughout the year, I learned how to be detail-oriented and how to manage my time even more effectively as the course is intensive and a lot of materials have to be covered in a short period of time. I strongly believe the experiences I have had so far are sufficient to carry out high quality work, and this is a wonderful opportunity to present the work that will exceed the expectations. Given that CF&RM program highly focuses in R and the time management skill I have, I am confident that I will be able to accomplish the project goals within the tight timeline while broadening my knowledge as an R developer. Also, I learn how to use Github from the department of Applied Mathematics training. Last but not least, this opening is the great chance for me to learn even more under the guidance of Professors Douglas Martin.

Contact Information

Student Name: Chindhanai Uthaisaad

Student postal address: 5319 24th Ave NE, Apt #205, Seattle, Washington - 98105

Telephone(s): 206-953-7833

Email(s): chindu@uw.edu, chindhanaipom@gmail.com

Student Affiliation

Institution : University of Washington

Program: Computational Finance and Risk Management (CF&RM)

Expected date of Graduation: December 2017

Contacts to verify enrollment: Laurie Feldman lf23@uw.edu

Schedule Conflicts

There is no schedule conflicts. I will treat the proposed GSoC project as a full time job in the summer.

Mentors

Mentor names:

1. Douglas Martin[1], an Emiritus Professor and the Director of CF&RM at the University of Washington

2. Eric Zivot[2], Robert Richards Chaired Professor of Economics, Co-Director of the MS Program in CF&RM, Adjunct Professor of Finance, Foster Business School, Adjunct Professor of Statistics and Adjunct Professor of Applied Mathematics

3. Brian Peterson[3], Partner, Head Trader, Automated Trading at DV Trading

4. Thomas Philips[4], Global Head of Front Office at BNP Paribas Investment Partners Institutional Division and Adjunct Professor at NYU Tandon School of Engineering

5. Kjell Konis[5], Acting Assistant Professor in CF&RM at the University of Washington

Mentor emails:

1. martinrd@comcast.net

2. ezivot@uw.edu

3. brian.peterson@r-project.org

4. tkpmep@gmail.com

5. kjellk@uw.edu

Have you been in touch with the mentors? When and how?

I have always been in touch with Prof. Douglas Martin to discuss the scope and details of the project whilst working on the proposal submission and acquire his guidance and thought to approach the GSoC project implementation. He also provides me the plans just in case that the timelines or goals change. I regularly have either a Skype call every Friday or schedule a meeting in person to have the touch base. I will be in touch with Prof. Eric, Brian, Thomas and Kjell in the summer via email and/or Skype call to discuss more about the further ideas and functionalities they might have on the package.

Coding Plan and Methods

In this project, I will apply an incremental software development scheme with the sets of weekly sprints according to the project milestones. I will inform my plan for the sprint at the beginning of each week and get feedback from the mentors. The document regarding the implementation I plan to accomplish in the following week will be presented with the list of all the written functions, relevant testing and the vignette documentation to be completed, so that they can skim through my plan, make comments and inform any changes as appropriate. Weekly sprint will lead to finished portions of the coding project and vignette, and hence minimizing the chances of work getting piled up in the end. Once a sprint is completed, I will review the progress of the sprint, document any tasks that are not yet completed, pass it to the mentors for review, get their feedbacks and document any changes suggested. The next sprint will imcorporate those leftovers. At the end of each phase, I will summarize and present the features added, test results and the additional features we might implement if time permits.

The project plan is to complete the advanced features of the fundamental factor models (FFM's) and the time series factor models (TSFM's) functionality of the factorAnalytics package developed in the GSoC 2016 project, and also implement additional features in Risk Models and Risk Budgeting (RMRB). According to the project description and project milstones, we list the project plan under *Key Deliverables*. The additional features I find interesting are listed under *Possible Additional Functionalities and Extensions*

Key Deliverables:

1. Fundamental Factor Models - Hybrid Factor Model

    (a) Implement the hybrid factor model to extract the significant (single) factor omitted from the model. The user can specify whether they want to apply the zero noise or finite noise to the model. The results include the correlation between the true Gram-Schmidt returns and the estimated hybrid factor returns, and the correlation between the true Gram-schmidt exposures to the estimated hybrid factor exposures in both small sample and large sample cases. These correlations will inform how well the estimated factor returns and exposures align with the true factor returns and exposures

2. Fundamental Factor Models - The Multi-Factor Model method

    (a) Implement the new exposure standardization method

    (b) Compute the quadratic utility optimal information ratio

    (c) Implement the Monte Carlo method for the multi-factor model analogous to the implemented single-factor model

3. Fundamental Factor Models - The Multi-dimensional outlier detection and visualization with robust squared distances based on robust covariance matrices

4. Fundamental Factor Models - EWMA and GARCH models

   (a) Enhance the models to handle the time-varying volatility of the residuals

5. Time Series Factor Models - Portfolio level returns and risk reports

   (a) Create tabular reports and graphs for risk decomposition and performance analysis

6. Time Series Factor Models - Multi-dimensional outlier detection and visualization

   (a) Implement the detection of outliers on a time-period by time-period basis, using the entire multivariate time series of portfolio returns on a moving window.

7. Time Series Factor Models - Quadratic and Interaction model terms, Lagged risk factors and lagged returns models, and models with ARIMA errors

   (a) Create the convenient user specification method interface for the quadratic interaction model terms

   (b) Lagged risk factors and lagged returns models, and models with ARIMA errors including use selected functionality in the robustarima package.

8. Time Series Factor Models

   (a) Implement EWMA and GARCH models for residuals for TSFM's

(b) Implement the CUSUM procedure in sequential analysis and statistical process control (SPC) to rapidly detect the changes in the mean of a noisy random process and use this to assess the active manager's performance. (IR is used as a measure of performance) The users can specify the threshold if the alarm will be raised so that the users have to reevaluate the active portfolio's performance.

(c) Implement parallel processing using R and Rcpp to assuage the computational workloads by utilizing multiple threads in `async()` function and `future` objects. This is most likely applied to the multi-factor model case where we might have to run heavy Monte Carlo simulations.

9. Fundamental Factor Models - Multi-currency risk modeling

10. Risk Models and Risk Budgeting - Risk parity portfolios for volatility risk

11. Risk Models and Risk Budgeting - Simple and robust risk budgeting with expected shortfall

12. Risk Models and Risk Budgeting - Russian Doll risk models

Possible Additional Functionalities and Extensions:

1. From the Hybrid Factor Models from Menchero and Mitra (2008) paper, they used one world factor and two style factors: one known factor and one omitted factor estimated by Gram-Schmidt model. We can, in gen-

eral, extend the model to handle the arbitrary number of style factors and omitted factors by iterative Gram-Schmidt model to higher dimentions.

2. Since CUSUM is sensitive to the correlation in the sequence of excess returns, we might need to implement a method to evaluate the choice of the benchmark to ensure that the excess returns are uncorrelated.

Timeline

Sprint 1: 5/30 - 6/05 Implement Key Deliverables point 1

Sprint 2: 6/06 - 6/12 Implement Key Deliverables point 2

Sprint 3: 6/13 - 6/19 Implement Key Deliverables point 3

Sprint 4: 6/20 - 6/26 Implement Key Deliverables point 4

Sprint 5: 6/27 - 7/03 Implement Key Deliverables point 5

Sprint 6: 7/04 - 7/10 Implement Key Deliverables point 6

Sprint 7: 7/11 - 7/17 Implement Key Deliverables point 7

Sprint 8: 7/18 - 7/24 Implement Key Deliverables point 8

Sprint 9: 7/25 - 7/31 Implement Key Deliverables point 9

Sprint 10: 8/01 - 8/07 Implement Key Deliverables point 10

Sprint 11: 8/08 - 8/14 Implement Key Deliverables point 11

Sprint 12: 8/15 - 8/21 Implement Key Deliverables point 12

Wrap-up: 8/22 - 8/29 Wrap up the remaining issues and incorporate mentor feedbacks to mitigate the issues.

Management of Coding Project

How do you propose to ensure code is submitted / tested?

According to the aforementioned plan, I will meet one or more of the mentors at least every week to plan and review the tasks and functionality needed to be implemented in each sprint. The factorAnalytics package used to be hosted on R-forge and the development process will take advantage of these facilities for version control via SVN, but this year we will use Github as the medium of the open-source R package development and version control. The quality and completeness of the work will be reviewed and assessed by the mentors at the end of each sprint. Any feedback from mentors, or failing test case scenarios will be documented in order to be fixed and reinvestigated in the next sprint. With the comments from each mentor, I can prepare the checklist in each phase for the functionalities to be implemented in the next sprint to make sure all the functionalities of the function are correctly built and properly tested in the testing phase. Any delays will be addressed every week and thus avoiding falling behind the schedule. Also, we follow an adaptive method so that necessary changes can be incorporated on the go when required and make appropriate corrections to the future schedules. I am also prepared to continue working on the project past the end date if necessary to deliver the key objectives.

The testing can be differentiated as the white box testing and black box testing. The white box testing is performed during the implementation where I check each line of code through unit testing. Then the black box testing is performed at the end of the sprint to check the overall functionalities of the software. Each outcomes will be documented and reviewed by the mentors at the end of each sprint. All functions and datasets will be documented using Roxygen2. For better understanding to the users, I will write code examples of function usage, and a vignette describing the equations and some relevant theory will be written to guide the users in the overall use of the function.

How often do you plan to commit? What changes in commit behavior would indicate a problem?

I plan to commit approximately when every solid logic is added to update the code base. The main reason is to keep track on important changes, keep my activity active to the mentors, and keep the idea simple. Also it is not too rarely for the changes to pile up and make mentors lost track on what is added.

I will use the `-m` flag to record a succint, descriptive, and specific comment and summary of changes made in the commit that will help me and mentors remember later on what I did and why. The changes from the former version and the recent version can be accessed by the function `$ git diff`, which informs us not only the change that has been made but also the error messages. We could also use `$ git show` which shows us what changes we made at an older commit as well as the commit message, rather than the differences between a commit and our working directory that we see by using `$ git diff`. The problem in commit bahavior stems from the overlapping changes. This is indicated when Git won't let us push it to GitHub even though we can commit the change locally since Git detects that the changes made in one copy overlap with those made in the other and stops me from trampling on my previous work. I will resolve this by pulling the changes from GitHub, merging them into the copy I am currently working on, and then pushing that to GitHub. Git keeps track of what we have merged, so we don't have to fix things by hand again when the I or mentors who made the first change pull again.

Test

I have submitted the sample code for the skew-t distribution information matrix and the way to compute the asymptotic standard error for the maximum likelihood estimators of skew-t distribution. The following is an example of the function to compute the information matrix of the Apple skew-t distributed returns MLEs using the function `st.mple()` in the sn package. There are four parameters in the skew-t distribution: location parameter ξ, scale parameter ω^2, skewness parameter α, and degrees of freedom ν. Since there is no analytic formula for the information matrix, we will use the numerical quadrature to compute the information matrix by using the function `integrate()` and set the tolerance to be 10^{-10}. This function will map the infinite range onto a finite interval and use globally adaptive interval subdivision in connection with extrapolation by Wynn's Epsilon algorithm, with the basic step being Gauss-Kronrod quadrature. This information matrix is then used to obtain the asymptotic standard error for the skew-t maximum likelihood parameters and, even more useful, the asymptotic variance of the expected shortfall. For example, the asymptotic standard errors for each of the MLEs are determined by the corresponding diagonal element of the inverse of the information matrix.

```
#Necessary elements for the information matrix computation
gamma_integrand = function(u,nu=5){
dt(u,df=nu+1)*(((nu+2)*u^2)/((nu+1)*(nu+1+u^2))-log(1+u^2/(nu+1)))
}

Gamma = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
value <- integrate(gamma_integrand,-Inf, varxi, rel.tol = 1e-6)$val
return(value)
}

beta_integrand = function(u,nu=5){
dt(u,df=nu+1)*(((nu+2)*u^2)/((nu+1)*(nu+1+u^2))-log(1+u^2/(nu+1)))^2
}

Beta = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
value <- integrate(beta_integrand,-Inf,varxi,rel.tol = 1e-6)$val
return(value)
}

delta_integrand = function(u,nu=5){
```

```
dt(u,df=nu+1)*((nu*u^2-2*nu-2)*u^2)/((nu+1)*(nu+1+u^2))^2
}

Delta = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
value <- integrate(delta_integrand,-Inf,varxi, rel.tol=1e-6)$val
return(value)
}

#Elements before taking the expectation
S_z = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
value <- -tau^2*z+(alpha*tau*nu*w)/(nu+z^2)
return(value)
}

S_zz = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
w_z <- -((nu*(nu+2)*alpha^2*z*w)/((nu+z^2+alpha^2*z^2)*(nu+z^2)))-
((nu*alpha*tau*w^2)/(nu+z^2))
value <- (2*tau^2*z^2)/(nu+z^2)-tau^2-(3*alpha*tau*nu*z*w)/(nu+z^2)^2+
```

```
(alpha*tau*nu*w_z)/(nu+z^2)
return(value)
}

S_zalpha = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
w_alpha <- -((nu+2)*alpha*z^2*w)/(nu+z^2+alpha^2*z^2)-z*tau*w^2
value <- (nu*tau*(w+alpha*w_alpha))/(nu+z^2)
return(value)
}

S_znu = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
w_nu <- 0.5*w*(((nu+2)*alpha^2*z^2)/((nu+z^2+alpha^2*z^2)*(nu+z^2))-
log(1+(alpha*z)^2/(nu+z^2))-Gamma(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/
dt(varxi,nu+1))+(alpha*z*(1-z^2)*w^2)/(2*tau*(nu+z^2)^2)
value <- (z*(1-z^2))/(nu+z^2)^2+(alpha*(nu*(3*z^2-1)+2*z^2)*w)/(2*tau*(nu+z^2)^3)+
(alpha*tau*nu*w_nu)/(nu+z^2)
return(value)
}

S_xixi = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- S_zz(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/omega^2
```

```
return(value)
}

S_xiomega = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
value <- z/omega^2*S_zz(y, xi=xi, omega=omega, alpha=alpha, nu=nu)+
S_z(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/omega^2
return(value)
}

S_xialpha = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_zalpha(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/omega
return(value)
}

S_xinu = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_znu(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/omega
return(value)
}

S_omegaomega = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
value <- 1/omega^2+z^2/omega^2*S_zz(y, xi=xi, omega=omega, alpha=alpha, nu=nu)+
2*z/omega^2*S_z(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

S_omegaalpha = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
```

```
value <- -z/omega * S_zalpha(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

S_omeganu = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
value <- -z/omega * S_znu(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

S_alphaalpha = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
w_alpha <- -((nu+2)*alpha*z^2*w)/(nu+z^2+alpha^2*z^2)-z*tau*w^2
value <- z*tau*w_alpha
return(value)
}

S_alphanu = function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
w_nu <- 0.5*w*(((nu+2)*alpha^2*z^2)/((nu+z^2+alpha^2*z^2)*(nu+z^2))-
log(1+(alpha*z)^2/(nu+z^2))-Gamma(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/
dt(varxi,nu+1))+(alpha*z*(1-z^2)*w^2)/(2*tau*(nu+z^2)^2)
value <- (z*(z^2-1)*w)/(2*tau*(nu+z^2)^2)+z*tau*w_nu
```

```
}

S_nunu <- function(y, xi=0, omega=1, alpha=3, nu=5){
z <- (y-xi)/omega
tau <- sqrt((nu+1)/(z^2+nu))
varxi <- alpha*z*tau
w <- dt(varxi, nu+1)/pt(varxi, nu+1)
w_nu <- 0.5*w*(((nu+2)*alpha^2*z^2)/((nu+z^2+alpha^2*z^2)*(nu+z^2))-
log(1+(alpha*z)^2/(nu+z^2))-Gamma(y, xi=xi, omega=omega, alpha=alpha, nu=nu)/
dt(varxi,nu+1))+(alpha*z*(1-z^2)*w^2)/(2*tau*(nu+z^2)^2)
value <- 0.25*(trigamma(1+nu/2)-trigamma(nu/2))+(2*nu^2+2*nu+1)/(2*nu^2*(nu+1)^2)
            +z^2/(2*nu*(nu+z^2))-(z^2*(nu^2+2*nu+z^2))/(2*nu^2*(nu+z^2)^2)
            -(alpha*z*(z^2-1)*(z^2+4*nu+3)*w)/(4*tau*(nu+1)*(nu+z^2)^3)
            +(alpha*z*(1-tau^2)*w_nu)/(2*tau*(nu+z^2))-Gamma(y, xi=xi,
                omega=omega, alpha=alpha, nu=nu)^2/
                (4*pt(varxi,nu+1)^2)-(alpha*z*(z^2-1)*Gamma(y, xi=xi,
                omega=omega, alpha=alpha, nu=nu)*w)/(4*pt(varxi,nu+1)
                *tau*(nu+z^2)^2)+(2*Delta(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
                +Beta(y, xi=xi, omega=omega, alpha=alpha, nu=nu))/(4*pt(varxi,nu+1))
                +(alpha*z*(z^2-1)*w)/(4*tau*(nu+z^2)^2)*(((nu+2)*alpha^2*z^2)/
                ((nu+1)*(nu+z^2+alpha^2*z^2))-log(1+(alpha^2*z^2)/(nu+z^2)))
return(value)
}

#Numerical Quadratures
g_xixi = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_xixi(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
```

```
}

g_xiomega = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_xiomega(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_xialpha = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_xialpha(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_xinu = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_xinu(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_omegaomega = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_omegaomega(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_omegaalpha = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_omegaalpha(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
```

```
return(value)
}

g_omeganu = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_omeganu(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_alphaalpha = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_alphaalpha(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_alphanu = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_alphanu(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

g_nunu = function(y, xi=0, omega=1, alpha=3, nu=5){
value <- -S_nunu(y, xi=xi, omega=omega, alpha=alpha, nu=nu)*
dst(y, xi=xi, omega=omega, alpha=alpha, nu=nu)
return(value)
}

#Function definition
InfoMat <- function(xi, omega, alpha, nu){
```

```
I_xixi <- integrate(g_xixi,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_xiomega <- integrate(g_xiomega,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_xialpha <- integrate(g_xialpha,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_xinu <- integrate(g_xinu, xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_omegaomega <- integrate(g_omegaomega,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_omegaalpha <- integrate(g_omegaalpha,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_omeganu <- integrate(g_omeganu,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_alphaalpha <- integrate(g_alphaalpha,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_alphanu <- integrate(g_alphanu,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I_nunu <- integrate(g_nunu,xi=xi, omega=omega, alpha=alpha, nu=nu,
-Inf, Inf, rel.tol = 1e-10)$val
I <- matrix(c(I_xixi,I_xiomega,I_xialpha,I_xinu,
                I_xiomega,I_omegaomega,I_omegaalpha,I_omeganu,
                I_xialpha,I_omegaalpha,I_alphaalpha,I_alphanu,
                I_xinu,I_omeganu,I_alphanu,I_nunu), nrow = 4, ncol = 4)
return(I)
}


#Now we fit the skew-t distribution to the Apple stock monthly returns
library(sn)
```

```r
AAPL = read.table("AAPL.dat", col.names = "AAPL", sep="\t")
ret = as.numeric(AAPL[,1])
n = length(ret)
fit.stST = st.mple(as.matrix(rep(1, n)), ret)
names(fit.stST$dp) = c("location", "scale", "skew", "dof")
xi = as.numeric(fit.stST$dp[1])
omega = as.numeric(fit.stST$dp[2])
alpha = as.numeric(fit.stST$dp[3])
nu = as.numeric(fit.stST$dp[4])

#Information matrix is therefore
InfoMat(xi, omega, alpha, nu)

##              [,1]        [,2]         [,3]         [,4]
## [1,]  71.45958913 -24.2486207  3.532425144  0.019759478
## [2,] -24.24862069  95.8557530  1.465836567 -0.296719293
## [3,]   3.53242514   1.4658366  0.279276481 -0.003618516
## [4,]   0.01975948  -0.2967193 -0.003618516  0.004317889

#Inverse of the information matrix
solve(InfoMat(xi, omega, alpha, nu))

##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.2174208  0.11949397  -3.319776   4.434431
## [2,]  0.1194940  0.07993843  -1.887394   3.364739
## [3,] -3.3197755 -1.88739414  54.586238 -68.762314
## [4,]  4.4344313  3.36473892 -68.762314 384.897200
```

From the inverse information matrix above, we have that

$$S.E._{\xi} = \sqrt{0.2174208} \approx 0.466$$
$$S.E._{\omega} = \sqrt{0.0799384} \approx 0.283$$
$$S.E._{\alpha} = \sqrt{54.586238} \approx 7.388$$
$$S.E._{\nu} = \sqrt{384.89720} \approx 19.62$$

The results inform us that we estimate the location and scale parameters much more accurately than the skewness and degrees of freedom.

References

1. Chaves, D., Hsu, J., Li, F., and Shakernia, O. (2012). "Efficient Algorithms for Computing Risk Parity Portfolio Weights", *The Journal of Investing*, Fall 2012.

2. Ding, Z. and Martin, R. D. (2016). "The Fundamental Law of Active Management Redux", SSRN 2730434.

3. Green, C. and Martin, R. D. (2014). "An Extension of a Method of Hardin and Rocke, with an Application to Multivariate Outlier Detection via the IRMCDMethod of Cerioli",

4. http://students.washington.edu/cggreen/uwstat/papers/cerioli_extension.pdf.

5. Kakushadze, Z. (2015). Russian-Doll Risk Models." *Journal of Asset Management*, 16(3), pp. 170-185.

6. Menchero, J. and Mitra, I. (2008). "The Structure of Hybrid Factor Models", *Journal of Investment Management* 6(3): 35-47.

7. Philips, T. K. and Liu, M. (2012). "Simple and Robust Risk Budgeting with Expected Shortfall", *Journal of Portfolio Management*, Fall 2011.

8. Philips, T. K., Yashchin, E. and Stein, D. M. (2003). "Using Statistical Process Control to Monitor Active Managers", *Journal of Portfolio Management*, Fall 2003, pp. 86-94.