Estimating Risk and Performance measures using FMMC: factorAnalytics vignette

Rohit Arora

July 6, 2015

Abstract

The purpose of this vignette is to demonstrate the use of `fmmc` and related function `fmmc.estimate.se` in the `factorAnalytics` package.

Contents

1 Overview **2**

1.1 Load Package 2

1.2 Summary of related functions 2

1.3 Data 2

2 Fit an FMMC object to the data **3**

3 Use FMMC objects to compute risk and performance estimates **4**

1 Overview

1.1 Load Package

The latest version of the `factorAnalytics` package can be downloaded from R-forge through the following command:

```
install.packages("factorAnalytics", repos="http://R-Forge.R-project.org")
```

Load the package and it's dependencies.

```
library(factorAnalytics)
options(digits=3)
```

1.2 Summary of related functions

Here's a list of the functions and methods demonstrated in this vignette:

- `fmmc(R, factors, parallel=FALSE, ...)`: Compute fmmc objects that can be used for calcuation of estimates and their standard errors using the Factor Model Monte Carlo methodology as described in Jiang and Martin (2013). These fmmc objects can then be reused to compute different risk and performance estimates along with their standard errors.

  item `fmmc.estimate.se(fmmcObjs, fun=NULL, se=FALSE, nboot=100, parallel = FALSE)` : Compute risk and performance estimates along with thier standard errors using the fmmc objects.

1.3 Data

The following examples primarily use the `managers` dataset from the `PerformanceAnalytics` package. It's an "xts" data object with 132 observations on 10 variables; frequency is monthly.

```
data(managers)
colnames(managers)

##  [1] "HAM1"        "HAM2"        "HAM3"        "HAM4"        "HAM5"
##  [6] "HAM6"        "EDHEC.LS.EQ" "SP500.TR"    "US.10Y.TR"   "US.3m.TR"

range(index(managers))

## [1] "1996-01-31" "2006-12-31"
```

In the example below we will compute expected shortfall for the monthly returns of 3 managers, HAM2, HAM5, and HAM6 using returns from HAM3, returns on S&P500, returns on 3M and 10Y treasury. Notice that managers HAM2, HAM5, and HAM6 have missing return history at the begining.

```
sub.data <- managers[,c(2, 5, 6),drop=FALSE]
head(sub.data)

##            HAM2 HAM5 HAM6
## 1996-01-31   NA   NA   NA
## 1996-02-29   NA   NA   NA
## 1996-03-31   NA   NA   NA
## 1996-04-30   NA   NA   NA
## 1996-05-31   NA   NA   NA
## 1996-06-30   NA   NA   NA

factors.data  <- managers[,c(3, 8, 9, 10),drop=FALSE]
```

2 Fit an FMMC object to the data

Let's take a look at the arguments for `fmmc`.

```
args(fmmc)

## function (R, factors, parallel = FALSE, ...)
## NULL
```

Given a matrix of returns and factors data one can choose to parallelize the computation of fmmc objects. Additional arguments can be passed to **`fitTsfm`** such as variable.selection or fit.method. The default value for variable selection is "subsets" and fit,method is defaulted to "LS". The default criteria for model selection is "BIC". Additional arguments can be passed to **`fitTsfm`**.

```
objs <- fmmc(sub.data, factors.data, parallel=FALSE, variable.selection="subsets")

## Warning in .fmmc.proc(R = R, factors = factors, ...):  some of the betas where NA
in .fmmc.proc.  Dropping those
## Warning in .fmmc.proc(R = R, factors = factors, ...):  some of the betas where NA
in .fmmc.proc.  Dropping those
```

```
## Warning in .fmmc.proc(R = R, factors = factors, ...):  some of the betas where NA
in .fmmc.proc.  Dropping those
```

3 Use FMMC objects to compute risk and performance estimates

Our goal is to compute expected shortfall to the returns data with missing history without discarding any factor data. We will compute expected shortfall in this example. We will define a function to compute es and bind the significance at 5

```
es <- function(r, alpha = 0.05) {
    r <- sort(r)
    cutoff <- ifelse( alpha == 0, 1, round(alpha*length(r)))
    -1/cutoff * sum(r[which((1:length(r)) < cutoff)])
}
```

We will pass this function along with the previously computed fmmc object to compute the estimate and the standard error using `fmmc.estimate.se`. Since computation of standard error for multiple returns can be time consuming one can choose to parallelize and utilize all the CPU cores. However, but default the option to parallelize is turned off.

```
result <- fmmc.estimate.se(objs, fun = es, se= TRUE, nboot = 50,
                           parallel = FALSE)
result

##      estimate      se
## HAM2   0.0541 0.00607
## HAM5   0.1059 0.01176
## HAM6   0.0424 0.00468
```

Note that with a different risk/performance measure, the previously computed fmmc objects can be reused.

References

Y. Jiang and D. Martin. Better Risk and Performance Estimates with Factor Model Monte Carlo. *SSRN Electronic Journal*, July 2013. ISSN 1556-5068. doi: 10.2139/ssrn.2295602. URL `http://papers.ssrn.com/abstract=2295602`.