Assignment 4

Nitish Ramkumar, Ian Laker, Prasanth Kumar

Extract the portfolio and risk free data. Apply the date constraints and remove the invalid columns. Sample data is for excess returns is as below.

1 Time Series regression

Time series regression can be performed by regressing the monthly returns of the 25 porfolios with 5 FF Factors. As we use lm, it will automatically include an intercept.

```
data1 <- cbind(portfolio.bm.excess,portfolio.size.excess,portfolio.opprofit.excess,
               portfolio.inv.excess,portfolio.beta.excess)
regressions.TimeSeries <- apply(data1,2,function(x){lm(x~FFfactors.data)})

alphas <- unlist(lapply(regressions.TimeSeries,function(x){x$coefficients[1]}))
residuals <- sapply(regressions.TimeSeries,function(x){x$residuals})
```

The alpha test can either be a asymptotic (chi-squared) or a exact sample size (F) test.

F-Test

In this test, we use don't know the standard deviation of errors. So we use the variance covariance matrix of te residuals.

```
fmean <- apply(FFfactors.data,2,mean)
sigmafinv <- chol2inv(chol(cov(FFfactors.data)))
fSharpeRatioSq <- t(fmean)%*%sigmafinv%*%fmean

sigmaeinv <- chol2inv(chol(cov(residuals)))
alphaterm <- t(alphas)%*%sigmaeinv%*%alphas

#FTerm
fterm <- ((T-N-K)/N)*alphaterm *chol2inv(chol(1+fSharpeRatioSq))
fterm
```

```
##          [,1]
## [1,] 2.094912
```

```
qf(0.95,N,T-N-K)
```

```
## [1] 1.524511
```

We can reject the Ftest null hypothesis, which is that alpha is significantly not 0.

Chi-Squared

In this case, we need to use the known standard deviation of errors. This can be retrieved out of all the regressions and converted into a diagonal matrix

```r
errorSEs <- sapply(regressions.TimeSeries,function(x){summary(x)$sigma})
errorSEMatrix <- diag(errorSEs)
alphaterm <- t(alphas)%*%(chol2inv(chol(errorSEMatrix)))%*%alphas

#chiSq
chisq <- T* alphaterm *chol2inv(chol(1+fSharpeRatioSq))
chisq
```

```
##          [,1]
## [1,] 48.23565
```

```r
qchisq(0.95,N)
```

```
## [1] 37.65248
```

We can reject the chi-squared null hypothesis, which is that alpha is significantly not 0.

2 Cross Sectional Regression

OLS Cross Sectional
Price of risk estimate

```r
OLSReg <- lm(apply(data1,2,mean)~ betas)
lambda.OLS.2 <- OLSReg$coefficients
lambda.OLS.2
```

```
##              (Intercept) betasFFfactors.dataMkt.RF
##                0.2079062                 0.2904833
##    betasFFfactors.dataSMB    betasFFfactors.dataHML
##                0.2531712                 0.2574591
##    betasFFfactors.dataRMW    betasFFfactors.dataCMA
##                0.2134350                 0.1711276
```

Price of risk error

```r
#lambda Errors OLS
allFactors <- cbind(1,FFfactors.data)
allBetas <- t(sapply(regressions.TimeSeries,function(x){x$coefficients}))
sandwich <- chol2inv(chol(crossprod(allBetas)))%*%t(allBetas)
lambdaVar <- (sandwich%*%cov(residuals)%*%t(sandwich) + cov(allFactors))/T
sqrt(diag(lambdaVar))
```

```
##         X1     Mkt.RF        SMB        HML        RMW        CMA
## 0.21395540 0.17739072 0.12311946 0.11887854 0.09830612 0.09664883
```

Price of risk error with Shanken correction

```r
#Errors Shanken correction
lambdaVarShanken <- lambdaVar + sandwich%*%cov(residuals)%*%t(sandwich)*((t(lambda.OLS.2)
                                %*%cov(allFactors)%*%lambda.OLS.2)[1,1])/T
sqrt(diag(lambdaVarShanken))
```

```
##        X1    Mkt.RF       SMB       HML       RMW       CMA
## 0.4012440 0.1782330 0.1268437 0.1346681 0.1179232 0.1292117
```

We notice a bigger change in the intercept value as compared to the factors because the shanken correction is applied primarily on the errors covariance matrix and not on the factors covariance matrix

R-squared

```
## [1] 0.8284747
```

Fama Macbeth Cross Sectional
Price of risk Estimate

```
FMBReg <- apply(data1,1,function(x){lm(x~betas)})
allLambdas.FMB <- t(sapply(FMBReg,function(x){x$coefficients}))
lambdaEst.FMB.2 <- apply(allLambdas.FMB,2,mean)
cbind(lambda.OLS.2,lambdaEst.FMB.2)
```

```
##                              lambda.OLS.2 lambdaEst.FMB.2
## (Intercept)                     0.2079062       0.2079062
## betasFFfactors.dataMkt.RF      0.2904833       0.2904833
## betasFFfactors.dataSMB         0.2531712       0.2531712
## betasFFfactors.dataHML         0.2574591       0.2574591
## betasFFfactors.dataRMW         0.2134350       0.2134350
## betasFFfactors.dataCMA         0.1711276       0.1711276
```

The estimated price of risk from both the OLS and Fama-MacBeth cross sectional regression are the same, as beta doesn't change with time.

Price of risk Error

```
#Fama Macbeth Error
#lambda.Error.FMB <- apply(allLambdas,2,function(x){sum((x-mean(x))^2)/(T^2)})
lambdaError.FMB.2 <- apply(allLambdas.FMB,2,sd)/sqrt(T)
lambdaError.FMB.2
```

```
##               (Intercept) betasFFfactors.dataMkt.RF
##                0.22171027                0.28305993
##    betasFFfactors.dataSMB    betasFFfactors.dataHML
##                0.12290151                0.11756676
##    betasFFfactors.dataRMW    betasFFfactors.dataCMA
##                0.09679477                0.08829092
```

R-squared

```
#Fama Macbeth Rsquared
Rsquared.FMB.2 <- mean(sapply(FMBReg,function(x){summary(x)$r.squared}))
Rsquared.FMB.2
```

```
## [1] 0.6647251
```

3a

Retrieve the necessary data from Ken Fench website.

Fama Macbeth and comparison with 2)

Estimates of price of risk

```
##                             lambdaEst.FMB.2 lambdaEst.FMB.3
## (Intercept)                       0.2079062       0.5261441
## betasFFfactors.dataMkt.RF         0.2904833       0.4259700
## betasFFfactors.dataSMB            0.2531712       0.1288284
## betasFFfactors.dataHML            0.2574591      -0.5752904
## betasFFfactors.dataRMW            0.2134350       0.1579265
## betasFFfactors.dataCMA            0.1711276       0.3888467
```

When we compare the estimates with the Fama MacBeth calculations from question 2, we can notice a higher beta on the market factor and Conservative Minus Aggressive factor, whereas it has reduced for other 3 factors (actually a negative for High Minus Low Book/Market ratio)

error of price of risk

```
##                             lambdaError.FMB.2 lambdaError.FMB.3
## (Intercept)                        0.22171027         0.3020768
## betasFFfactors.dataMkt.RF          0.28305993         0.3552286
## betasFFfactors.dataSMB             0.12290151         0.1824995
## betasFFfactors.dataHML             0.11756676         0.1759233
## betasFFfactors.dataRMW             0.09679477         0.1948873
## betasFFfactors.dataCMA             0.08829092         0.2329114
```

The errors have increased for every factor compared to the previous case.

R-squared

```
##      Rsquared.FMB.2 rsquared.FMB.3
## [1,]      0.6647251      0.4881979
```

R squared has reduced in this case as compared to the previous case.

3bi

Price of risk with Intercept

```
##   (Intercept) betaReg3.Capm
##     1.2591451    -0.3338983
```

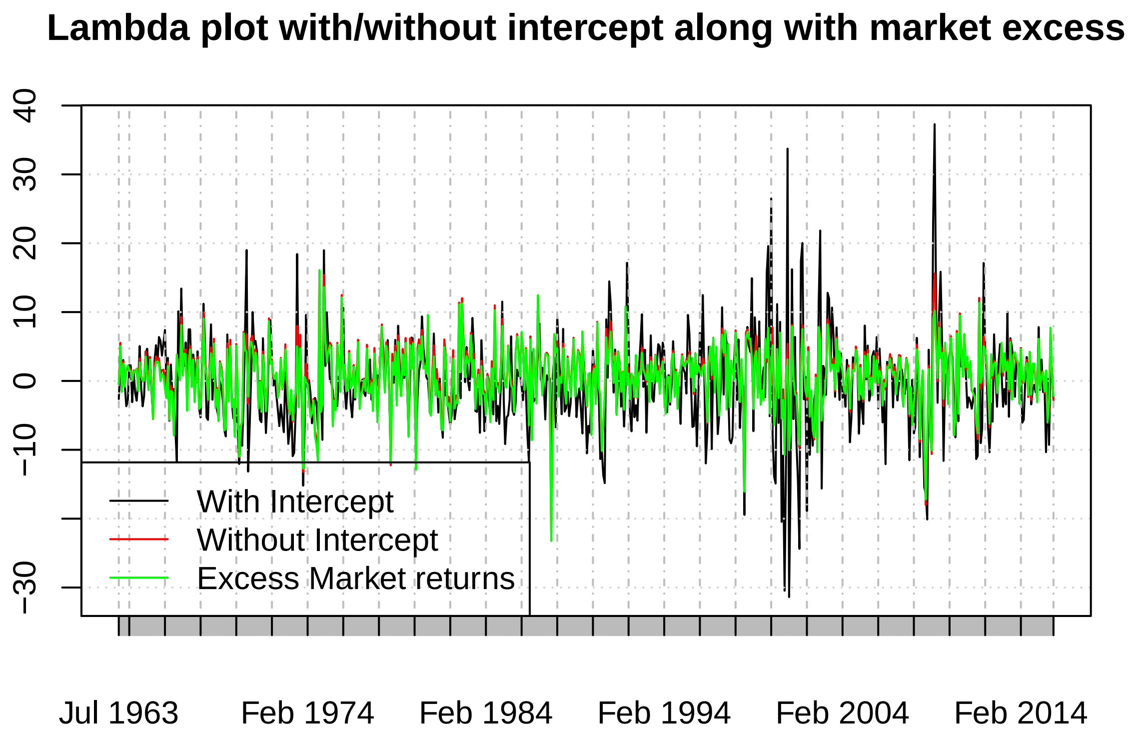
Fama Macbeth regression with intercept results in lambda of **-0.3338983**

Price of risk without Intercept

```
## [1] 0.8533837
```

Fama Macbeth regression without intercept leads to a lambda of **0.8533837**

3bii



It can be seen that the market excess returns are change similar to the lambda graph without intercept.

```
##                   With.Intercept Without.Intercept    Mkt.RF
## With.Intercept         1.0000000         0.6803602 0.6346203
## Without.Intercept      0.6803602         1.0000000 0.9885109
## Mkt.RF                 0.6346203         0.9885109 1.0000000
```

Our intuition from the graph is further justified from the correlation matrix, where there is high correlation between Market excess returns and lambdas without intercept. So, if we want to be close to the market excess returns (in this case, this was the only independent variable) and make CAPM valid, we need to use without intercept method.

R Code

```
##DATA RETRIEVAL
options(warn = -1)
suppressMessages(library(xts))
options(warn = 0)
setwd("C:/_UCLA/237E_Empirical/Assigments/Assignment4")

FFfactors.data <- read.csv("F-F_Research_Data_5_Factors_2x3.csv",header=TRUE,sep = ","
                           ,stringsAsFactors = FALSE,skip = 3,nrows = 630)
FFfactors.data$X <- as.yearmon(as.character(FFfactors.data$X),format="%Y%m")
FFfactors.data <- xts(FFfactors.data[,-1],order.by = FFfactors.data$X)[,c(1,2,3,4,5)]
```

```r
riskfree.data <- read.csv("F-F_Research_Data_Factors.csv",header=TRUE,sep = ",",
                          stringsAsFactors = FALSE,skip = 3,nrows=1086)
riskfree.data$X <- as.yearmon(as.character(riskfree.data$X),format="%Y%m")
riskfree.data <- xts(riskfree.data[,-1],order.by = riskfree.data$X)
riskfree.data <- riskfree.data[index(riskfree.data)>="1963-07-01"
                               & index(riskfree.data)<="2015-12-31",]

portfolio.beta.data <- read.csv("Portfolios_Formed_on_BETA.csv",header=TRUE,sep = ",",
                                stringsAsFactors = FALSE,skip = 15,nrows = 630)
portfolio.beta.data$X <- as.yearmon(as.character(portfolio.beta.data$X),format="%Y%m")
portfolio.beta.data <- xts(portfolio.beta.data[,-1],order.by = portfolio.beta.data$X)
[,c("Lo.20", "Qnt.2","Qnt.3","Qnt.4","Hi.20")]
colnames(portfolio.beta.data) <- c("Beta.Lo.20","Beta.Qnt.2","Beta.Qnt.3",
                                   "Beta.Qnt.4","Beta.Hi.20")

portfolio.opprofit.data <- read.csv("Portfolios_Formed_on_OP.csv",header=TRUE,
                                    sep = ",",stringsAsFactors = FALSE,skip = 18,nrows = 630)
portfolio.opprofit.data$X <- as.yearmon(as.character(portfolio.opprofit.data$X)
                                        ,format="%Y%m")
portfolio.opprofit.data <- xts(portfolio.opprofit.data[,-1],order.by = portfolio.opprofit.data$X)[,c("Lo
colnames(portfolio.opprofit.data) <- c("OpPft.Lo.20","OpPft.Qnt.2","OpPft.Qnt.3","OpPft.Qnt.4","OpPft.Hi

portfolio.inv.data <- read.csv("Portfolios_Formed_on_INV.csv",header=TRUE,sep = ",",
                               stringsAsFactors = FALSE,skip = 17,nrows = 630)
portfolio.inv.data$X <- as.yearmon(as.character(portfolio.inv.data$X),format="%Y%m")
portfolio.inv.data <- xts(portfolio.inv.data[,-1],order.by = portfolio.inv.data$X)
[,c("Lo.20", "Qnt.2","Qnt.3","Qnt.4","Hi.20")]
colnames(portfolio.inv.data) <- c("inv.Lo.20","inv.Qnt.2","inv.Qnt.3",
                                  "inv.Qnt.4","inv.Hi.20")

portfolio.bm.data <- read.csv("Portfolios_Formed_on_BE-ME.csv",header=TRUE,sep = ",",
                              stringsAsFactors = FALSE,skip = 23,nrows = 1086)
portfolio.bm.data$X <- as.yearmon(as.character(portfolio.bm.data$X),format="%Y%m")
portfolio.bm.data <- xts(portfolio.bm.data[,-1],order.by = portfolio.bm.data$X)
portfolio.bm.data <- portfolio.bm.data[index(portfolio.bm.data)>="1963-07-01" & index(portfolio.bm.data
colnames(portfolio.bm.data) <- c("BM.Lo.20","BM.Qnt.2","BM.Qnt.3","BM.Qnt.4",
                                 "BM.Hi.20")

portfolio.size.data <- read.csv("Portfolios_Formed_on_ME.csv",header=TRUE,sep = ",",
                                stringsAsFactors = FALSE,skip = 12,nrows = 1086)
portfolio.size.data$X <- as.yearmon(as.character(portfolio.size.data$X),format="%Y%m")
portfolio.size.data <- xts(portfolio.size.data[,-1],order.by = portfolio.size.data$X)
portfolio.size.data <- portfolio.size.data[index(portfolio.size.data)>="1963-07-01" & index(portfolio.si
colnames(portfolio.size.data) <- c("Size.Lo.20","Size.Qnt.2","Size.Qnt.3",
                                   "Size.Qnt.4","Size.Hi.20")

#Calculate Excess Returns
portfolio.beta.excess <- xts(apply(portfolio.beta.data,2,function(x){t(x - riskfree.data$RF)}),index(po
portfolio.opprofit.excess <- xts(apply(portfolio.opprofit.data,2,function(x){t(x - riskfree.data$RF)}),i
portfolio.inv.excess <- xts(apply(portfolio.inv.data,2,function(x){t(x - riskfree.data$RF)}),index(port
portfolio.bm.excess <- xts(apply(portfolio.bm.data,2,function(x){t(x - riskfree.data$RF)}),index(portfo
portfolio.size.excess <- xts(apply(portfolio.size.data,2,function(x){t(x - riskfree.data$RF)}),index(po
```

```r
T <- 630
N <- 25
K <- 6

#1
data1 <- cbind(portfolio.bm.excess,portfolio.size.excess,portfolio.opprofit.excess
               ,portfolio.inv.excess,portfolio.beta.excess)
regressions.TimeSeries <- apply(data1,2,function(x){lm(x~FFfactors.data)})

alphas <- unlist(lapply(regressions.TimeSeries,function(x){x$coefficients[1]}))
residuals <- sapply(regressions.TimeSeries,function(x){x$residuals})

#F-Test
fmean <- apply(FFfactors.data,2,mean)
sigmafinv <- chol2inv(chol(cov(FFfactors.data)))
fSharpeRatioSq <- t(fmean)%*%sigmafinv%*%fmean

sigmaeinv <- chol2inv(chol(cov(residuals)))
alphaterm <- t(alphas)%*%sigmaeinv%*%alphas

fterm <- ((T-N-K)/N)*alphaterm *chol2inv(chol(1+fSharpeRatioSq))
fterm
qf(0.95,N,T-N-K)

#chi-sq
errorSEs <- sapply(regressions.TimeSeries,function(x){summary(x)$sigma})
errorSEMatrix <- diag(errorSEs)
alphaterm <- t(alphas)%*%(chol2inv(chol(errorSEMatrix)))%*%alphas

chisq <- T* alphaterm *chol2inv(chol(1+fSharpeRatioSq))
chisq
qchisq(0.95,N)

##2 Cross Sectional Regression
betas <- t(sapply(regressions.TimeSeries,function(x){x$coefficients[-1]}))

##OLS Cross Sectional**
#Price of risk estimate*
OLSReg <- lm(apply(data1,2,mean)~ betas)
lambda.OLS.2 <- OLSReg$coefficients
lambda.OLS.2

#Price of risk error*
allFactors <- cbind(1,FFfactors.data)
allBetas <- t(sapply(regressions.TimeSeries,function(x){x$coefficients}))
sandwich <- chol2inv(chol(crossprod(allBetas)))%*%t(allBetas)
lambdaVar <- (sandwich%*%cov(residuals)%*%t(sandwich) + cov(allFactors))/T
sqrt(diag(lambdaVar))

#Price of risk error with Shanken correction*
lambdaVarShanken <- lambdaVar +       sandwich%*%cov(residuals)%*%t(sandwich)
*((t(lambda.OLS.2)%*%cov(allFactors)%*%lambda.OLS.2)[1,1])/T
sqrt(diag(lambdaVarShanken))
```

```
#R-squared
summary(OLSReg)$r.squared

#Fama Macbeth Cross Sectional**
#Price of risk Estimate*
FMBReg <- apply(data1,1,function(x){lm(x~betas)})
allLambdas.FMB <- t(sapply(FMBReg,function(x){x$coefficients}))
lambdaEst.FMB.2 <- apply(allLambdas.FMB,2,mean)
cbind(lambda.OLS.2,lambdaEst.FMB.2)

#Price of risk Error*
#lambda.Error.FMB <- apply(allLambdas,2,function(x){sum((x-mean(x))^2)/(T^2)})
lambdaError.FMB.2 <- apply(allLambdas.FMB,2,sd)/sqrt(T)
lambdaError.FMB.2

#R-squared*
Rsquared.FMB.2 <- mean(sapply(FMBReg,function(x){summary(x)$r.squared}))
Rsquared.FMB.2

##3a
portfolio.momentum.data <- read.csv("10_Portfolios_Prior_12_2.csv",header=TRUE,sep = ",",
                                    stringsAsFactors = FALSE,skip = 10,nrows = 1080)
portfolio.momentum.data$X <- as.yearmon(as.character(portfolio.momentum.data$X)
                                        ,format="%Y%m")
portfolio.momentum.data <- xts(portfolio.momentum.data[,-1]
                               ,order.by = portfolio.momentum.data$X)
portfolio.momentum.data <- portfolio.momentum.data[
                              index(portfolio.momentum.data)>="1963-07-01"
                              & index(portfolio.momentum.data)<="2015-12-31",]
colnames(portfolio.momentum.data) <- c("Mom.Dec1","Mom.Dec2","Mom.Dec3","Mom.Dec4","Mom.Dec5","Mom.Dec6
                                       "Mom.Dec7","Mom.Dec8","Mom.Dec9","Mom.Dec10")

portfolio.netshares.data <- read.csv("Portfolios_Formed_on_NI.csv",header=TRUE,sep = ",",
                                     stringsAsFactors = FALSE,skip = 16,nrows=630)
portfolio.netshares.data$X <- as.yearmon(as.character(portfolio.netshares.data$X)
                                         ,format="%Y%m")
portfolio.netshares.data <- xts(portfolio.netshares.data[,-1]
                                ,order.by = portfolio.netshares.data$X)
portfolio.netshares.data <- portfolio.netshares.data[,c(3:7)]
colnames(portfolio.netshares.data) <- c("NetShares.Q1","NetShares.Q2","NetShares.Q3",
                                        "NetShares.Q4","NetShares.Q5")

portfolio.resvar.data <- read.csv("Portfolios_Formed_on_RESVAR.csv",header=TRUE,sep = ",",stringsAsFact
portfolio.resvar.data$X <- as.yearmon(as.character(portfolio.resvar.data$X)
                                      ,format="%Y%m")
portfolio.resvar.data <- xts(portfolio.resvar.data[,-1]
                             ,order.by = portfolio.resvar.data$X)
portfolio.resvar.data <- portfolio.resvar.data[
  index(portfolio.resvar.data)>="1963-07-01"
  & index(portfolio.resvar.data)<="2015-12-31",c(1:5)]
colnames(portfolio.netshares.data) <- c("ResVar.Q1","ResVar.Q2",
                                        "ResVar.Q3","ResVar.Q4","ResVar.Q5")
```

```
portfolio.10industry.data <- read.csv("10_Industry_Portfolios.csv",
          header=TRUE,sep = ",",stringsAsFactors = FALSE,skip = 11,nrows=1080)
portfolio.10industry.data$X <- as.yearmon(
  as.character(portfolio.10industry.data$X),format="%Y%m")
portfolio.10industry.data <- xts(portfolio.10industry.data[,-1]
                                 ,order.by = portfolio.10industry.data$X)
portfolio.10industry.data <- portfolio.10industry.data[
  index(portfolio.10industry.data)>="1963-07-01"
  & index(portfolio.10industry.data)<="2015-12-31",]


#Fama Macbeth and comparison with 2)**
data3 <- cbind(portfolio.momentum.data,portfolio.netshares.data,
               portfolio.resvar.data,portfolio.10industry.data)
regressions3 <- apply(data3,2,function(x){lm(x~FFfactors.data)})
betas3 <- t(sapply(regressions3,function(x){x$coefficients[-1]}))

#Estimates of price of risk*
FMBReg3 <- apply(data3,1,function(x){lm(x~betas3)})
lambdaEst.FMB.3 <- apply(t(sapply(FMBReg3,function(x){x$coefficients})),2,mean)
cbind(lambdaEst.FMB.2,lambdaEst.FMB.3)

#error of price of risk*
lambdaError.FMB.3 <- apply(t(sapply(FMBReg3,function(x){x$coefficients})),2,sd)/sqrt(T)
cbind(lambdaError.FMB.2,lambdaError.FMB.3)

#R-squared*
rsquared.FMB.3 <- mean(sapply(FMBReg3,function(x){summary(x)$r.squared}))
cbind(Rsquared.FMB.2,rsquared.FMB.3)

##3bi
#Price of risk with Intercept
TimeSeriesReg3.Capm <- apply(data3,2,function(x){lm(x~FFfactors.data[,1])})
betaReg3.Capm <- sapply(TimeSeriesReg3.Capm,function(x){x$coefficients[-1]})

FMBReg3.Capm.Int <- apply(data3,1,function(x){lm(x~betaReg3.Capm)})
lambdaFMB3.Capm.IntTS <- t(sapply(FMBReg3.Capm.Int,function(x){x$coefficients}))
lambdaFMB3.Capm.Int <- apply(lambdaFMB3.Capm.IntTS,2,mean)
lambdaFMB3.Capm.Int

#Price of risk without Intercept**
FMBReg3.Capm.WoInt <- apply(data3,1,function(x){lm(x~betaReg3.Capm-1)})
lambdaFMB3.Capm.WoIntTS <- sapply(FMBReg3.Capm.WoInt,function(x){x$coefficients})
mean(lambdaFMB3.Capm.WoIntTS)

##3bii
lambdaFMB3.Capm.Int.xts <- xts(lambdaFMB3.Capm.IntTS[,2],order.by = as.yearmon(row.names(lambdaFMB3.Capm
colnames(lambdaFMB3.Capm.Int.xts) <- "With Intercept"
lambdaFMB3.Capm.WoInt.xts <- xts(lambdaFMB3.Capm.WoIntTS,order.by = as.yearmon(index(lambdaFMB3.Capm.In
colnames(lambdaFMB3.Capm.WoInt.xts) <- "Without Intercept"

plot(lambdaFMB3.Capm.Int.xts,type="l",main="Lambda plot with/without intercept along with market excess
par(new=TRUE)
```

```
lines(lambdaFMB3.Capm.WoInt.xts,type="l",col="red")
par(new=TRUE)
lines(FFfactors.data[,1],type="l",col="green")
legend("bottomleft",lty=c(1,1,1), col=c("black","red","green"),c("With Intercept","Without Intercept","I
cor(cbind(lambdaFMB3.Capm.Int.xts,lambdaFMB3.Capm.WoInt.xts,FFfactors.data[,1]))
```