Introduction Asset Pricing

ASSIGNMENT 2

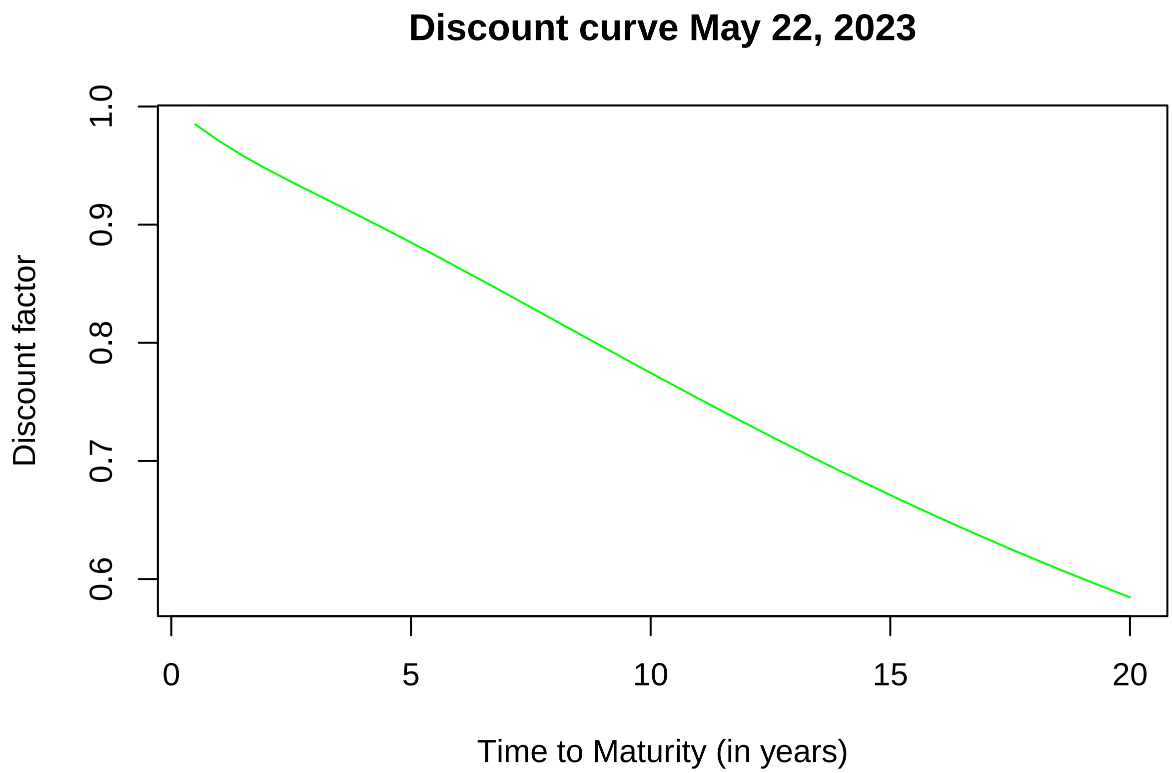
- Group 18:
  - Nikodem Baehr 2076515
  - Sam Friedlaender 2070091
  - Marcin Pietruk 2075739

Question 1

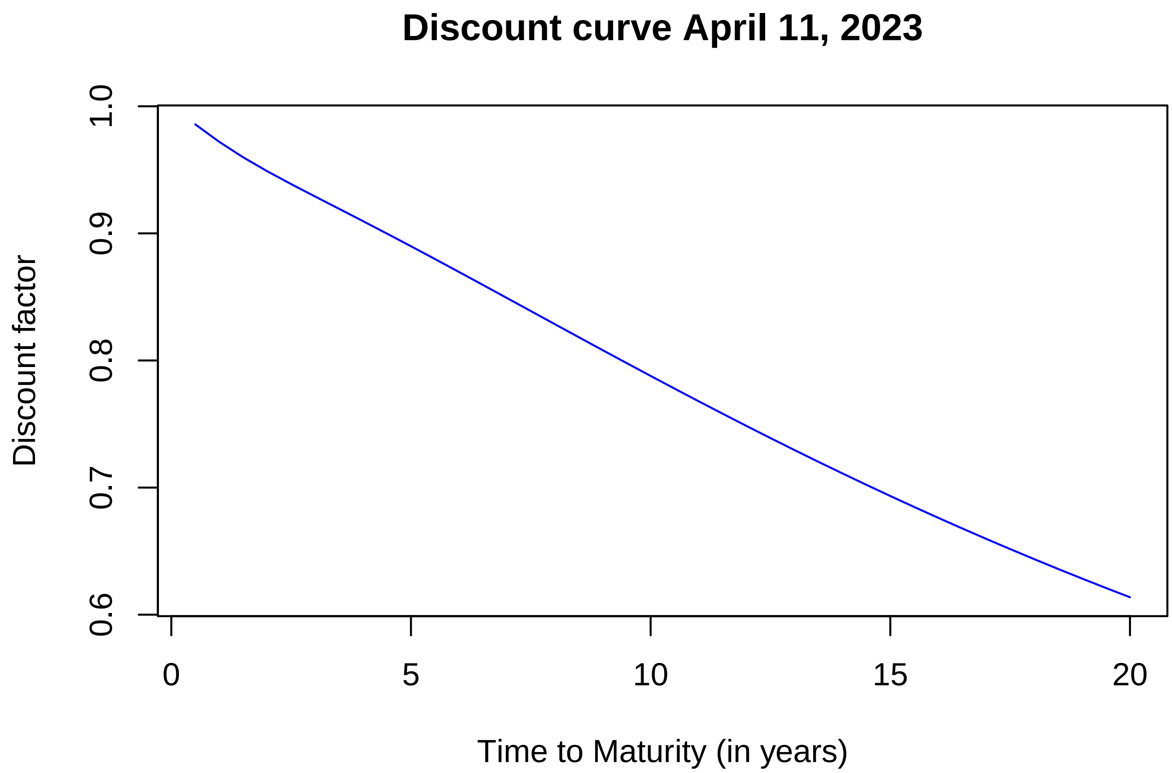
We decided to choose 22nd of May as the day we computed the NSS yield curve for.We will compare the resulting graph with the graph calculated on the 11th of April.

```
#Time measured in years, yields calculated with six month maturities
deltat<-1/2
ttm=seq(from=deltat,to=20, by=deltat)

# Parameter values ECB May 22, 2023
tau1=0.737992
tau2=12.486127
beta0=1.043188
beta1=1.885881
beta2=2.431911
beta3 = 4.988250



# Parameter values ECB April 11, 2023
tau1_11=0.682621
tau2_11=11.692957
beta0_11=1.189183
beta1_11=1.447696
beta2_11=2.518072
beta3_11 = 3.744450



#Setting up the four terms of the NSS specification
exph1<-exp(-ttm/tau1)
```

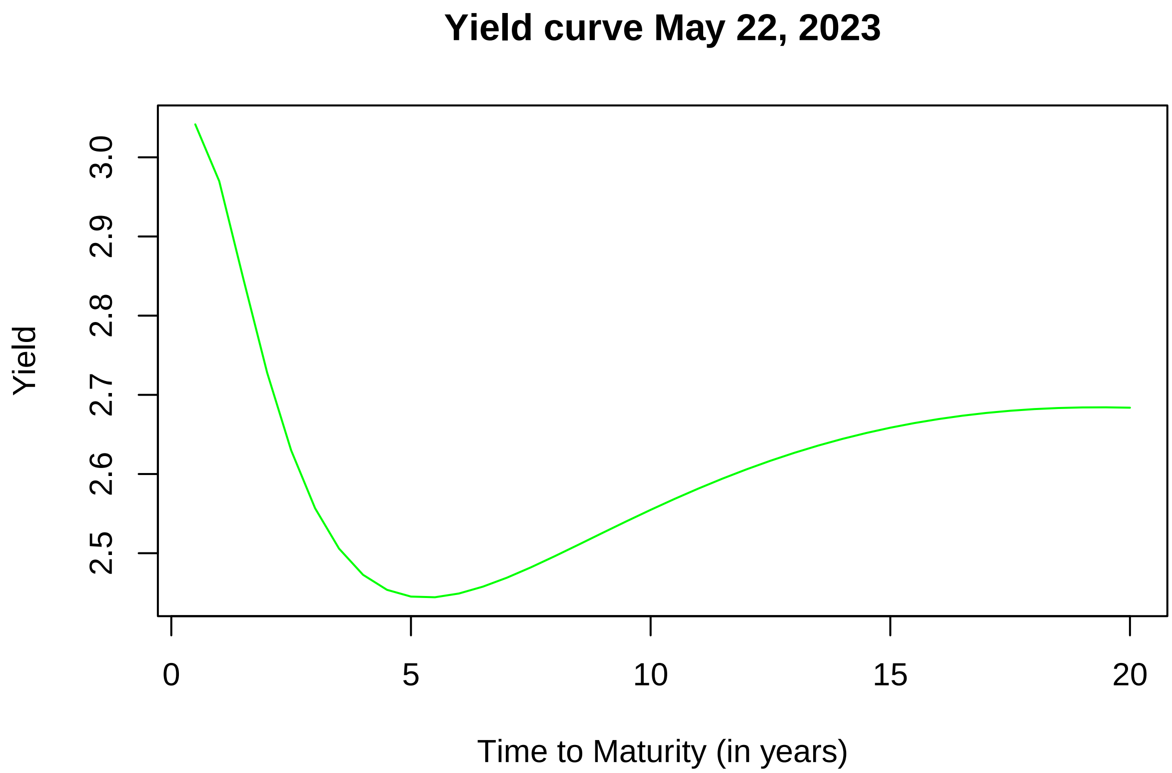
```
exph2<-exp(-ttm/tau2)
exph1_11<-exp(-ttm/tau1_11)
exph2_11<-exp(-ttm/tau2_11)

NSS0<-1
NSS1<-(1-exph1)/(ttm/tau1)
NSS2<-NSS1-exph1
NSS3<-(1-exph2)/(ttm/tau2)-exph2


NSS1_11<-(1-exph1_11)/(ttm/tau1_11)
NSS2_11<-NSS1_11-exph1_11
NSS3_11<-(1-exph2_11)/(ttm/tau2_11)-exph2_11

# Calculating the yield curve (in %)
yield<-beta0*NSS0+beta1*NSS1+beta2*NSS2+beta3*NSS3
yield_11<-beta0_11*NSS0+beta1_11*NSS1_11+beta2_11*NSS2_11+beta3_11*NSS3_11
# Calculating the discount curve (based on continuous compounding)
discount=exp(-ttm*yield/100)
discount_11=exp(-ttm*yield_11/100)

# Constructing the figures

### plot discount factor - NSS

plot(ttm,discount,type="l",main="Discount curve May 22, 2023",col="green",xlab="Time to Maturity (in yea
```



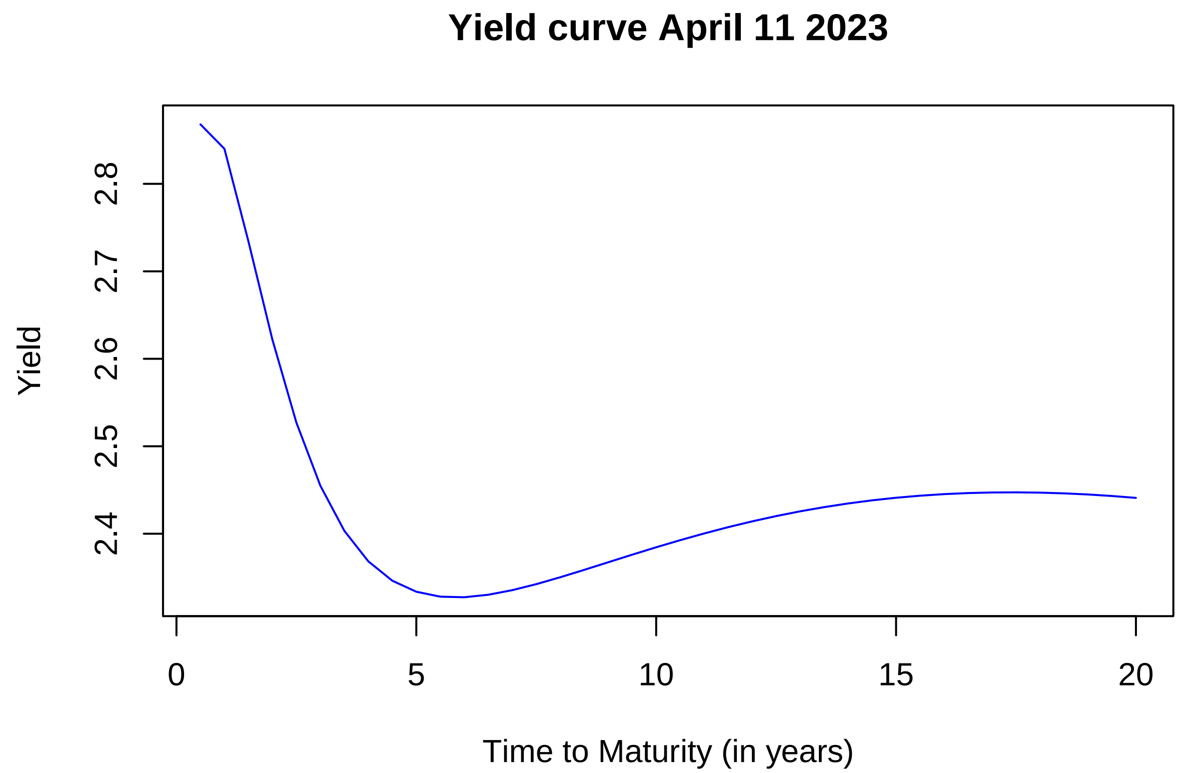

```
plot(ttm,discount_11,type="l",main="Discount curve April 11, 2023",col="blue",xlab="Time to Maturity (in
```




```
### plot yield curve - NSS

plot(ttm,yield,type="l",main="Yield curve May 22, 2023",col="green",xlab="Time to Maturity (in years)",y
```




```
plot(ttm,yield_11,type="l",main="Yield curve April 11 2023",col="blue",xlab="Time to Maturity (in years)
```



When comparing the parameter values of 11th of April and 22nd of May we can see that in the latter case the values of τ_1 and τ_2 are bigger than those in April. This causes the location of the "humps" to be shifted more to the right. Furthermore, β_0, β_2, are bigger in the April case. This means that on 11th of April "Long rates" were higher than on 22nd on may. After looking at β_0 $+\beta_1$ in both cases, we can conclude that the "short. rate" was bigger in May than in April. However, β_3 was bigger in May meaning that the second "hump" is larger then.

Question 2

We downloaded the Fama French Five Factors in a time period from January 2000 to March of 2023 with monthly frequency. We also downloaded 10 portfolios formed on Size (ME) and 10 portfolios formed Book to Market (BE-ME), both excluding dividends. The ME portfolios are constructed at the end of each June using the June market equity and NYSE breakpoints. BE/ME is book equity at the last fiscal year end of the prior calendar year divided by ME at the end of December of the prior year.

```
#Load the packages
#install.packages("sandwich")
#install.packages("lmtest")
#install.packages("ggplot2")
library("sandwich", quietly=TRUE)
library("lmtest", quietly=TRUE)
```

```
#library("ggplot2", quietly=TRUE)

rm(list=ls())
FF5US <- read.table("C:/Users/nikod/Downloads/fama5us.txt",quote="\"", comment.char="")
BEME <- read.table("C:/Users/nikod/Downloads/Portfolios_Formed_on_BE-ME_Wout_Div.txt", quote="\"", comme
ME <- read.table("C:/Users/nikod/Downloads/Portfolios_Formed_on_ME_Wout_Div.txt", quote="\"", comment.cl



# FF5Emerging factors

rme<-FF5US[,2]
f1<-rme
f2<-FF5US[,3]
f3<-FF5US[,4]
f4<-FF5US[,5]
f5<-FF5US[,6]
rf<-FF5US[,7]

# Transforming the data to appropriate format: Total Returns
# (i.e., Payoff/Price)
rme<-rme/100
rf<-rf/100
rm<-1+(rme+rf)
mrm<-mean(rm)

# Fama 5 matrix
fama<-FF5US[,2:7]


R_beme<-BEME[,10:19]/100
# R_beme = R(eturns) Dec(iles) BE(ME)
# Gross returns
R_beme<-1+R_beme;


R_me<-ME[,10:19]/100
# R_me = Returns Deciles Size
# Gross returns
R_me<-1+R_me

# We collect all the data
rdecg<-data.frame(R_beme,R_me,rm)
```

We present summary descriptive statistics of the portfolios and Fama-French Five factors.

FF5Emerging: Descriptive Statistics of the 7 portfolios:

```
##        V2                 V3                 V4                 V5         
##  Min.   :-17.2300   Min.   :-15.3500   Min.   :-13.9500   Min.   :-18.7300  
##  1st Qu.: -2.0200   1st Qu.: -1.5450   1st Qu.: -1.6850   1st Qu.: -1.0350  
##  Median :  1.1700   Median :  0.1900   Median : -0.0300   Median :  0.4400  
##  Mean   :  0.5204   Mean   :  0.2528   Mean   :  0.2009   Mean   :  0.4534  
##  3rd Qu.:  3.2350   3rd Qu.:  2.0150   3rd Qu.:  1.8000   3rd Qu.:  1.6000  
##  Max.   : 13.6500   Max.   : 18.3400   Max.   : 12.7500   Max.   : 13.0900  
##        V6                V7        
##  Min.   :-6.9200   Min.   :0.0000  
##  1st Qu.:-1.0350   1st Qu.:0.0100  
##  Median : 0.0100   Median :0.0800  
##  Mean   : 0.3209   Mean   :0.1265  
##  3rd Qu.: 1.4450   3rd Qu.:0.1900  
##  Max.   : 9.0500   Max.   :0.5600  
```

ME: Descriptive Statistics of 10 portfolios based on size:

```
##       V10              V11              V12              V13        
##  Min.   :0.7018   Min.   :0.8435   Min.   :0.8348   Min.   :0.8460  
##  1st Qu.:0.9727   1st Qu.:0.9797   1st Qu.:0.9809   1st Qu.:0.9808  
##  Median :1.0171   Median :1.0083   Median :1.0089   Median :1.0104  
##  Mean   :1.0076   Mean   :1.0056   Mean   :1.0062   Mean   :1.0066  
##  3rd Qu.:1.0451   3rd Qu.:1.0352   3rd Qu.:1.0352   3rd Qu.:1.0329  
##  Max.   :1.2114   Max.   :1.1510   Max.   :1.1315   Max.   :1.1454  
##       V14              V15              V16              V17        
##  Min.   :0.8262   Min.   :0.8218   Min.   :0.8182   Min.   :0.7550  
##  1st Qu.:0.9808   1st Qu.:0.9822   1st Qu.:0.9819   1st Qu.:0.9819  
##  Median :1.0109   Median :1.0121   Median :1.0105   Median :1.0073  
##  Mean   :1.0060   Mean   :1.0065   Mean   :1.0069   Mean   :1.0032  
##  3rd Qu.:1.0325   3rd Qu.:1.0346   3rd Qu.:1.0337   3rd Qu.:1.0316  
##  Max.   :1.1327   Max.   :1.1325   Max.   :1.1682   Max.   :1.1575  
##       V18              V19        
##  Min.   :0.7500   Min.   :0.7277  
##  1st Qu.:0.9828   1st Qu.:0.9773  
##  Median :1.0109   Median :1.0152  
##  Mean   :1.0065   Mean   :1.0079  
##  3rd Qu.:1.0388   3rd Qu.:1.0435  
##  Max.   :1.1673   Max.   :1.2146  
```

BEME: Descriptive Statistics of 10 portfolios based on book to market:

```
##       V10              V11              V12              V13
##  Min.   :0.8394   Min.   :0.7774   Min.   :0.7624   Min.   :0.7565
##  1st Qu.:0.9819   1st Qu.:0.9721   1st Qu.:0.9695   1st Qu.:0.9688
##  Median :1.0098   Median :1.0113   Median :1.0101   Median :1.0155
##  Mean   :1.0046   Mean   :1.0083   Mean   :1.0082   Mean   :1.0086
##  3rd Qu.:1.0315   3rd Qu.:1.0452   3rd Qu.:1.0525   3rd Qu.:1.0494
##  Max.   :1.1313   Max.   :1.2946   Max.   :1.2565   Max.   :1.2115
##       V14              V15              V16              V17
##  Min.   :0.7739   Min.   :0.7875   Min.   :0.7858   Min.   :0.7775
##  1st Qu.:0.9698   1st Qu.:0.9685   1st Qu.:0.9763   1st Qu.:0.9805
##  Median :1.0127   Median :1.0129   Median :1.0127   Median :1.0123
##  Mean   :1.0073   Mean   :1.0074   Mean   :1.0076   Mean   :1.0075
##  3rd Qu.:1.0461   3rd Qu.:1.0441   3rd Qu.:1.0416   3rd Qu.:1.0410
##  Max.   :1.1816   Max.   :1.1954   Max.   :1.1666   Max.   :1.1707
##       V18              V19
##  Min.   :0.7944   Min.   :0.7852
##  1st Qu.:0.9800   1st Qu.:0.9818
##  Median :1.0112   Median :1.0107
##  Mean   :1.0077   Mean   :1.0070
##  3rd Qu.:1.0406   3rd Qu.:1.0352
##  Max.   :1.1446   Max.   :1.1413
```

Question 3

```r
# Collecting the total return data: dim  Rvect = number of observations
# times number of returns
rvect<-data.frame(R_beme,R_me)
# Matrix size.
dimObs<-dim(rvect)
dimT<-dimObs[1]
dimJ<-dimObs[2]

# Vectors of ones
iotaT<-rep(1,dimT)
iotaJ<-rep(1,dimJ)

# Required data related to SDF
consfama<-cbind(iotaT,fama)
consfama_numeric <- apply(consfama, 2, as.numeric)

# We define a and B
```

```r
a<--iotaJ
B<--(1/dimT)*(t(rvect)%*%consfama_numeric)

# first round W (identity matrix
W<-diag(dimJ)

# First round estimates

x<-solve(t(B)%*%W%*%B)%*%(t(B)%*%W%*%a)

# Resulting values SDF

SDF<- consfama_numeric%*%x

# Resulting values moments (appearing in moment condition)

Moment1<-as.matrix(iotaT%*%t(iotaJ)-(SDF%*%t(iotaJ))*rvect)

# Estimate of the variance of the moments
S<-(1/dimT)*(t(Moment1)%*%Moment1)

# V of the first round
Vfirst<-solve(t(B)%*%W%*%B)%*%(t(B)%*%W%*%S%*%W%*%B)%*%solve(t(B)%*%W%*%B)

# standard errors of x

sxfirst<-sqrt(diag(Vfirst/dimT))

print("estimates, standard errors, t-statistics")
```

```
## [1] "estimates, standard errors, t-statistics"
```

```r
print(cbind(x,sxfirst, x/sxfirst))
```

```
##                        sxfirst
## iotaT   2.803525e+00 1.1503003  2.4372118505
## V2      1.977315e-02 0.1323342  0.1494182436
## V3      1.670040e-05 0.1040592  0.0001604894
## V4     -6.817621e-02 0.1332583 -0.5116096518
## V5      1.030645e-01 0.2290227  0.4500189975
## V6      3.257378e-01 0.3164613  1.0293133288
## V7     -1.549369e+01 9.3787182 -1.6520050930
```

The above output contains the estimates of the SDF for the five factors of the Fama-French model. The t-value at 5%, with 18 degrees of freedom (two samples of 10 portfolios each so 20-2=18) is given as $t_{18;0.05} = 1.7344$ at 4 decimal places. Factors' estimates are very small and the corresponding standard errors are small (with the exception of the risk free rate and the CMA factor). Given small errors, but small estimates, the factors are statistically indifferent from 0 (one could argue against risk-free factor).

```r
#3.2

# optimal weighting matrix W

Wopt<-solve(S)

# GMM estimates with optimal weighting matrix

xopt<-solve(t(B)%*%Wopt%*%B)%*%(t(B)%*%Wopt%*%a)

# Resulting values SDF 2

SDFopt<-consfama_numeric%*%xopt

# Resulting values moments (appearing in moment condition)
Momentopt<-as.matrix(iotaT%*%t(iotaJ)-(SDFopt%*%t(iotaJ))*rvect)
# re-estimate Sopt using xopt
Sopt<-(1/dimT)*(t(Momentopt)%*%Momentopt)
Woptn<-solve(Sopt);

# V of second round. Since W = S^(-1), the formula for the variance
# more simplified.
Vopt<-solve(t(B)%*%Wopt%*%B)
Voptn<-solve(t(B)%*%Woptn%*%B)

# the asymptotic st. errors x
sxopt<-sqrt(diag(Vopt/dimT))
sxoptn<-sqrt(diag(Voptn/dimT))

# Output second round
print("estimate, standard error, standard error (S re-restimated), t-statistics (re-estimated)")
```

```
## [1] "estimate, standard error, standard error (S re-restimated), t-statistics (re-estimated)"
```

```r
print(cbind(xopt, sxopt, sxoptn, xopt/sxopt))
```

```
##                          sxopt     sxoptn
```

```
## iotaT   2.58234374 0.77038890 0.70098574  3.35200019
## V2      0.03049251 0.09525987 0.08713052  0.32009815
## V3     -0.00171817 0.08642118 0.07764568 -0.01988135
## V4     -0.07075698 0.09983358 0.09131349 -0.70874924
## V5      0.12766351 0.18858924 0.17508816  0.67693954
## V6      0.30423487 0.22900054 0.21025025  1.32853345
## V7    -13.92327286 6.64644932 6.06265613 -2.09484376
```

The biggest difference is the further decrease in the CMA factor coefficient. Others mostly increase/decrease marginally. As expected, when using the optimal weighting matrix the standard errors decrease.

```
#3.3
# We just follow the formulas
mMomentopt<-colMeans(Momentopt)
HansenJ<-dimT*t(mMomentopt)%*%solve(Sopt)%*%mMomentopt
# Output: J-test, df, and p-value
print("   J-test,    df,         p-value")
```

```
## [1] "   J-test,    df,         p-value"
```

```
print(c(HansenJ, dimJ-2, 1-pchisq(HansenJ,dimJ-2)))
```

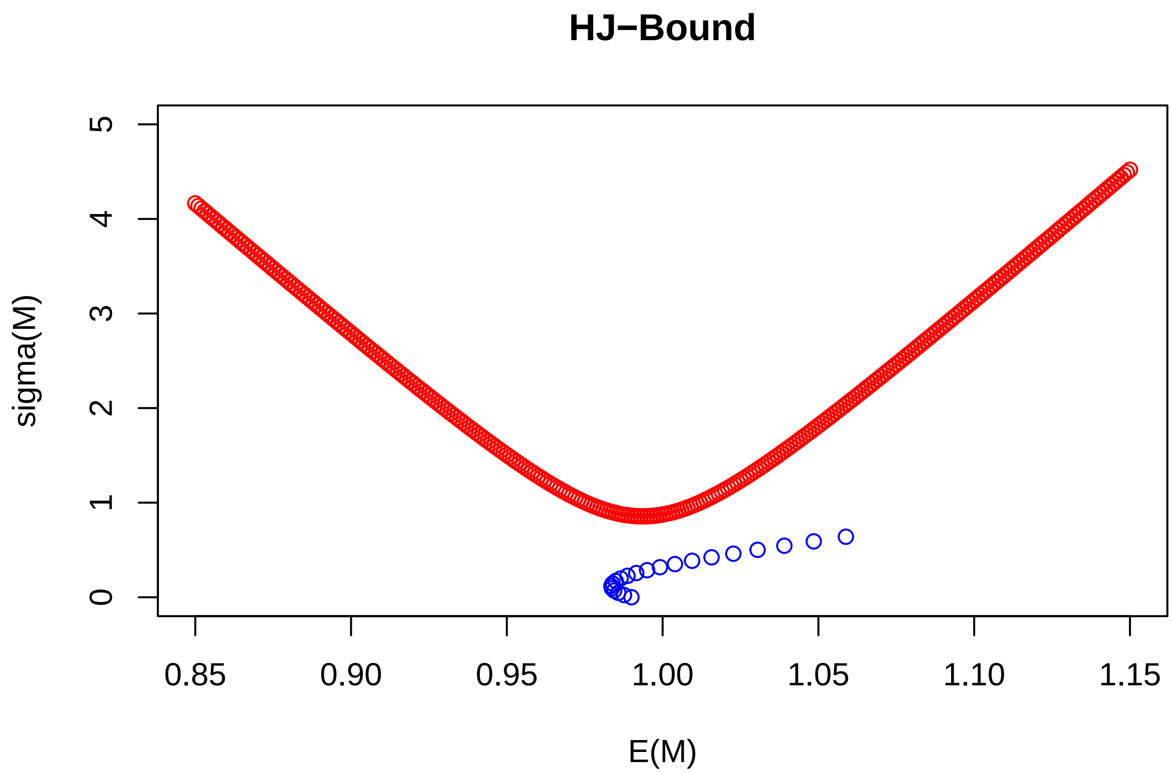
```
## [1]  1.9493747 18.0000000  0.9999991
```

From the output of the Hansen's J-test, we see that the test-statistic is insignificant due to a p-value of 99%. We are testing the null hypothesis that the Fama-French Five factor model is valid, against the alternative hypothesis that the estimates significantly violate the Hansen-Jagannathan Bound. Since we fail to reject the null hypothesis, we found statistical support for the validity of Fama-French Five factor model.

Question 4

```
#4.1
# rvect contains the gross returns
rvect<-rdecg
# Size of the matrix: dimT time dimension, dimJ = number of returns
dimJ<-ncol(rvect)

mRvect<-colMeans(rvect)
vRvect<-cov(rvect)

# We are going to determine the HJ-Bound for the following values of E(M)
mM<-seq(from=0.85,to=1.15,by=0.001)
```

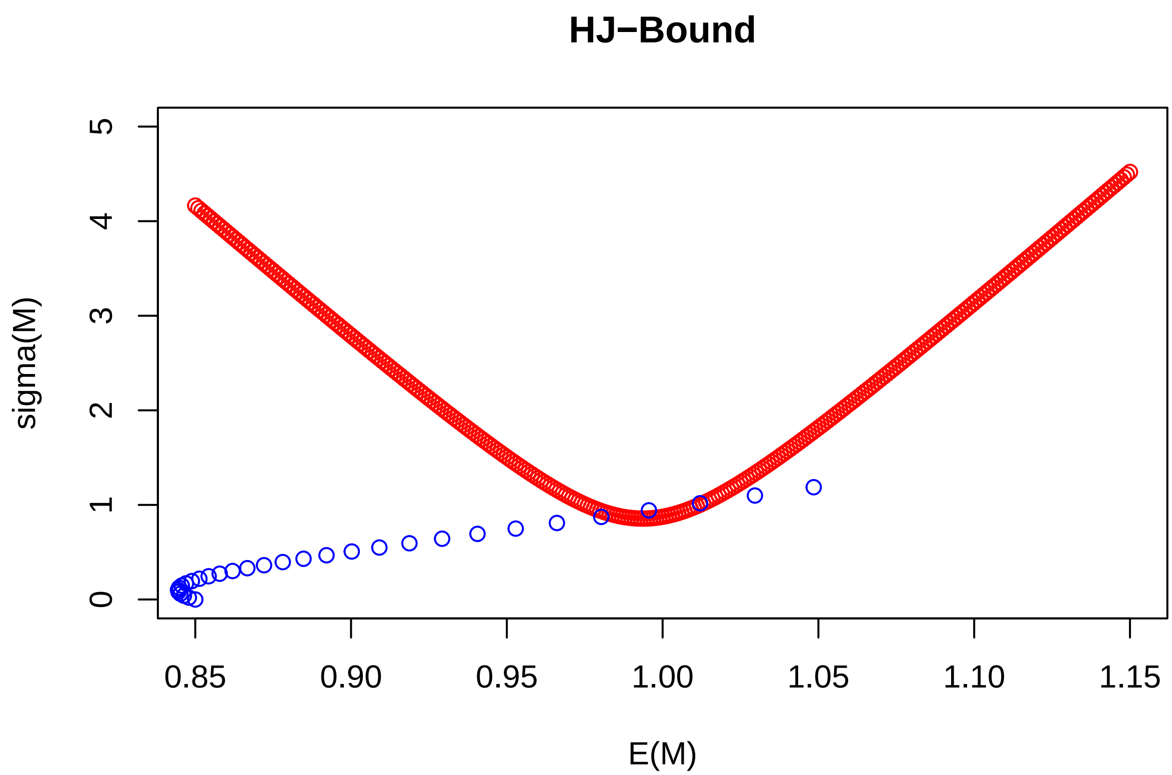
```r
HJBm<-rep(0,NROW(mM))

# define iotaJ
iotaJ<-rep(1,(dimJ))

# The Hansen-Jagannathan bound for each mM-value is then equal to
for (i in 1:NROW(mM))
{
  HJBm[i]<-sqrt(t(mM[i]*mRvect-iotaJ)%*%solve(vRvect)%*%(mM[i]*mRvect-iotaJ))
}

#4.2
# choose our range and step of gamma values
gamma<-seq(from=0,to=10,by=0.5)
# Calculate the resulting values of the SDF, size T times dim(gamma)

Rmgamma<-matrix(0,NROW(rme),NROW(gamma))
for (i in 1:NROW(gamma))
{
   Rmgamma[,i]<-0.99*(rm)^-gamma[i]
}

# calculate the means and volatilities (column-wise)
mRMgamma<-apply(Rmgamma,2,mean)
sRMgamma<-apply(Rmgamma,2,sd)



# We plot the outcomes in a figure

plot(mM,HJBm,type="p",main="HJ-Bound",col="red",xlab="E(M)",ylab="sigma(M)",ylim=c(0,5))
points(mRMgamma,sRMgamma,col="blue")
```



Beta represents the market correlation and gamma the risk aversion. The model becomes worse at explaining market prices at higher (positive) values of beta. When someone's portfolio is highly correlated with the market and the model's errors are greater at higher correlation coefficient, the portfolio's is at a higher risk of pricing errors. To minimise the risk, the person would look into hedging the portfolio in order to reduce market risk. The model also becomes worse at explaining market prices at lower values of gamma. The more risk seeking the investor is, the higher are the errors within the model. The higher the pricing errors, the more opportunities there are for arbitrage. Therefore a combination a risk loving investor in a booming market could exploit it the possibility of arbitrage. The first plot, with $\beta = 0.99$ and the γ range 0 to 10 (often accepted range of values for risk aversion). In our market, the curves doesn't cross for any of the γs with the HJ Bound, meaning that the SDF isn't validated.

```
gamma<-seq(from=0,to=15,by=0.5)
# Calculate the resulting values of the SDF, size T times dim(alpha)

Rmgamma<-matrix(0,NROW(rme),NROW(gamma))
for (i in 1:NROW(gamma))
{
  Rmgamma[,i]<-0.85*(rm)^-gamma[i]
}

# Calculate the means and volatilities (column-wise).
```

```r
# The function "apply" can take columnwise means and standard deviations
# with option "2".
mRMgamma<-apply(Rmgamma,2,mean)
sRMgamma<-apply(Rmgamma,2,sd)



# We plot the outcomes in a figure

plot(mM,HJBm,type="p",main="HJ-Bound",col="red",xlab="E(M)",ylab="sigma(M)",ylim=c(0,5))
points(mRMgamma,sRMgamma,col="blue")
```



The second plot with lower $\beta = 0.85$ and the broader γ range (0,15), the curve intersects with the HJ Bound for values around the 13-14 range of the risk aversion coefficient. Yet, these values are higher than the accepted values for the risk aversion coefficient and as a result, we can conclude that the SDF isn't validated.