MSCA32001 Assignment 2

Duo Zhou

7/6/2020

Create efficient frontier, CAPM model and APT model for a group of stocks representing health care sector and industrial sector.

The names of the selected companies are in the file Industrials_Health_Names.csv.

The period of observation is from="2014-7-1",to="2015-7-1".

For the sector indices use SPDR XLV (health care sector) and XLI (industrial sector).

For the broad market index use SPY.

For the risk-free rate use Fed Funds effective rate.

Note that it may not be possible to find interpretation of PCA factors in terms of real assets or indices. In such cases it is possible to use PCA factors without interpretation.

```r
# Loading quantmod
suppressWarnings(suppressMessages(library(quantmod)))
```

```r
# reading the names of selected stocks
datapath<-"C:/Users/zd000/Desktop/MSCA/Financial Analytics/Assignments/week2/"
SP500.IH <-read.csv(
  file=paste(datapath,"Industrials_Health_Names.csv",sep="/"),header=F)
SP500.IH.names <- as.character(SP500.IH[,1])
FedFunds.RF<-read.csv(file=paste(datapath,"RIFSPFF_NB.csv",sep="/"))
# FedFunds From "2014-7-1" to "2015-7-1"
FedFunds.RF<-FedFunds.RF[15094:15345,]
```

```r
# Loading prices of selected stocks, sectors and SNP500
suppressWarnings(suppressMessages((getSymbols(SP500.IH.names, from="2014-7-1",to="2015-7-1"))))
```

```
##  [1] "CAT" "FDX" "GE"  "HON" "LMT" "NOC" "UNP" "UPS" "UTX" "WM"  "ABT" "AET"
## [13] "HUM" "JNJ" "MDT" "PFE"
```

```r
suppressWarnings(getSymbols("XLV",from="2014-7-1",to="2015-7-1"))
```

```
## [1] "XLV"
```

```r
suppressWarnings(getSymbols("XLI",from="2014-7-1",to="2015-7-1"))
```

```
## [1] "XLI"
```

```r
suppressWarnings(getSymbols("SPY",from="2014-7-1",to="2015-7-1"))
```

```
## [1] "SPY"
```

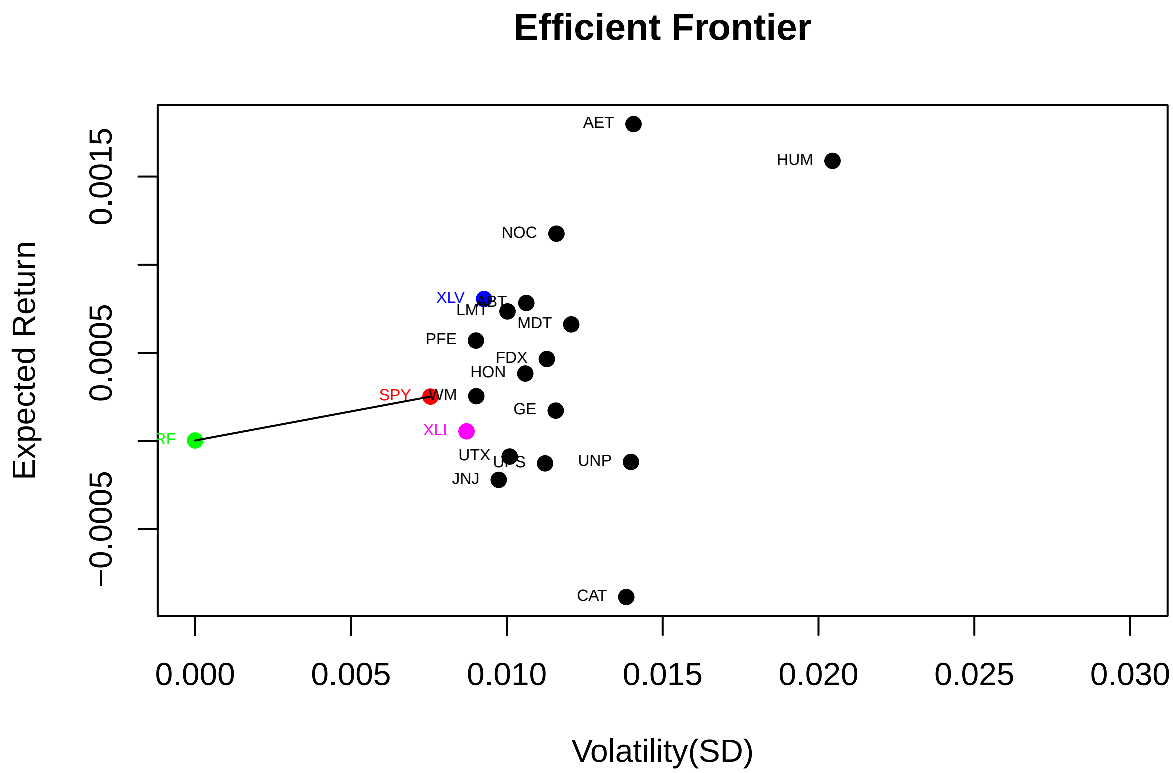
1. Efficient Frontier

Calculating Mean and SD for each stock, sector and SPY

```r
# Calculate mean and standard deviation for each stock
Mean.Sd.SP500.companies<-
cbind(sd=sapply(SP500.IH.names,function(z) sd(diff(log(get(z)[,6])),na.rm=TRUE)),
mean=sapply(SP500.IH.names,function(z) mean(diff(log(get(z)[,6])),na.rm=TRUE)))
# Calculate the mean and standard deviation for XLV
Mean.Sd.XLV<-c(sd(diff(log(XLV[,6])),na.rm=TRUE),mean(diff(log(XLV[,6])),na.rm=TRUE))
# Calculate the mean and standard deviation for XLI
Mean.Sd.XLI<-c(sd(diff(log(XLI[,6])),na.rm=TRUE),mean(diff(log(XLI[,6])),na.rm=TRUE))
# Calculate the mean and standard deviation for SPY
Mean.Sd.SPY<-c(sd(diff(log(SPY[,6])),na.rm=TRUE),mean(diff(log(SPY[,6])),na.rm=TRUE))
# Calculate the mean of Fed Funds
Mean.FedFunds<-mean(FedFunds.RF[,2])/100/365
```

Plot the SPY companies on SD vs. Mean Diagram, with SPY, XLV, XLI and risk-free rate

```r
plot(Mean.Sd.SP500.companies,ylab="Expected Return",
xlab="Volatility(SD)",main="Efficient Frontier", pch=19,xlim=c(0,.03))
points(Mean.Sd.SPY[1],Mean.Sd.SPY[2],col="red",pch=19)
points(Mean.Sd.XLV[1],Mean.Sd.XLV[2],col="blue",pch=19)
points(Mean.Sd.XLI[1],Mean.Sd.XLI[2],col="magenta",pch=19)
points(0,Mean.FedFunds,col="green",pch=19)
lines(c(0,Mean.Sd.SPY[1]),c(Mean.FedFunds, Mean.Sd.SPY[2]))
text(Mean.Sd.SP500.companies,labels=rownames(Mean.Sd.SP500.companies),cex=.5,pos=2)
text(Mean.Sd.SPY[1],Mean.Sd.SPY[2],labels="SPY",cex=.5,col="red",pos=2)
text(Mean.Sd.XLV[1],Mean.Sd.XLV[2],labels="XLV",cex=.5,col="blue",pos=2)
text(Mean.Sd.XLI[1],Mean.Sd.XLI[2],labels="XLI",cex=.5,col="magenta",pos=2)
text(0,Mean.FedFunds,labels="FF.RF",cex=.5,col="green",pos=2)
```



From above graph I would choose stocks above the security market line in my profolio. We can also see that the health(XLV) sector did better than SPY and Industrial sector(XLI) performed worse than SPY during the sample period.

2. CAPM model

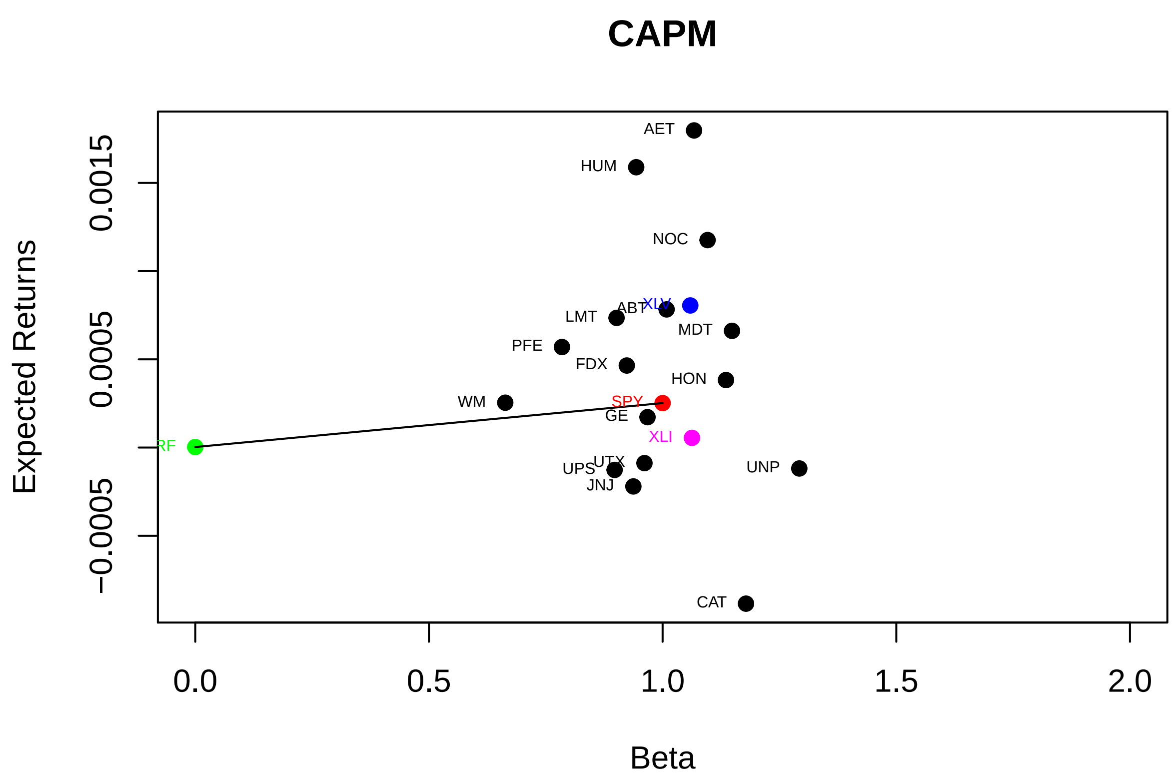
```
#Normalize fed funds to daily values
FedFunds.daily<-FedFunds.RF[,2]/100/365
# Extract the coefficients from linear models. Y is the daily excess return and X
SP500.companies.sector.betas<-
as.matrix(sapply(c(SP500.IH.names,"XLV","XLI"),function(z)
lm(I(diff(log(get(z)[,6]))-FedFunds.daily)~-1+I(diff(log(SPY[,6]))-FedFunds.daily))$coefficients))
```

```
# Given Row Names for betas
rownames(SP500.companies.sector.betas)<-c(SP500.IH.names,"XLV","XLI")
# Combine Company and Sector Means
Mean.company.sector<-c(Mean.Sd.SP500.companies[,2],"XLV"=Mean.Sd.XLV[2],"XLI"=Mean.Sd.XLI[2])
```

Plot CAPM diagram

```
plot(SP500.companies.sector.betas[1:16], Mean.company.sector[1:16],pch=19,
     main="CAPM", xlim=c(0,2), ylab= "Expected Returns", xlab="Beta")
points(1,Mean.Sd.SPY[2],col="red",pch=19)
points(0,Mean.FedFunds,col="green",pch=19)
```

```r
points(SP500.companies.sector.betas[17],Mean.company.sector[17],col="blue",pch=19)
points(SP500.companies.sector.betas[18],Mean.company.sector[18],col="magenta",pch=19)
lines(c(0,1),c(Mean.FedFunds,Mean.Sd.SPY[2]))
text(SP500.companies.sector.betas[1:16],Mean.company.sector[1:16],
     labels=rownames(SP500.companies.sector.betas)[1:16],cex=.5,pos=2)
text(1,Mean.Sd.SPY[2],labels="SPY",cex=.5,col="red",pos=2)
text(0,Mean.FedFunds,labels="FF.RF",cex=.5,col="green",pos=2)
text(SP500.companies.sector.betas[17],Mean.company.sector[17],
     labels="XLV",cex=.5,col="blue",pos=2)
text(SP500.companies.sector.betas[18],Mean.company.sector[18],
     labels="XLI",cex=.5,col="magenta",pos=2)
```



From above graph, we can come to the same conclusion as the efficient frontier model.

3. Arbitrage Pricing Theory

```r
# Calculate the log differences for each stock
Stock.Portfolio.Returns <- as.data.frame(matrix(NA,nrow=dim(SPY)[1]-1,
                    ncol=length(SP500.IH.names)))
colnames(Stock.Portfolio.Returns) <- c(SP500.IH.names)

for (i in colnames(Stock.Portfolio.Returns)){
  Stock.Portfolio.Returns[,i] <- diff(log((get(i)[,6])))[-1,]
}
dim(Stock.Portfolio.Returns)
```

```
## [1] 251  16
```

```
head(Stock.Portfolio.Returns)
```

```
##             CAT           FDX            GE           HON          LMT
## 1  0.003566067 -0.0007222551  0.007923160 -0.003282574 -0.010387138
## 2  0.013778205  0.0087616209  0.009351131  0.006343826  0.005770424
## 3 -0.008316728 -0.0126436005 -0.004103718 -0.005283780 -0.010056953
## 4 -0.006374740 -0.0033683951 -0.014307703 -0.001803033 -0.006909281
## 5  0.006193029  0.0009921098 -0.001897515  0.003497139  0.004569466
## 6 -0.007107138 -0.0070298098 -0.004569892 -0.002754152  0.001581634
##             NOC          UNP           UPS          UTX           WM
## 1 -0.005745934 -0.000299681  0.0067831337 -0.006837770 -0.008062727
## 2  0.012036741  0.009052628  0.0031816648  0.003381393  0.005382170
## 3 -0.008951484 -0.007155825 -0.0061801763 -0.003728780 -0.003809263
## 4 -0.006347700 -0.001697185 -0.0032983096 -0.008989139  0.001345956
## 5  0.003763498  0.005181958  0.0007769025 -0.002018514 -0.008104451
## 6  0.001000873 -0.009989076 -0.0086802160 -0.002902729 -0.001357412
##              ABT           AET          HUM           JNJ           MDT
## 1  0.0147041154  0.0104257146  0.002394738 -9.431279e-05  0.0000000000
## 2  0.0023900034  0.0085259639  0.004696030 -4.165108e-03  0.0099643191
## 3 -0.0091124533 -0.0142102067 -0.015011276  9.910922e-03 -0.0146693665
## 4 -0.0111435951 -0.0141462579 -0.006727404 -7.069123e-03  0.0001569921
## 5  0.0053448564  0.0014788880  0.002273608  3.022256e-03 -0.0003144911
## 6 -0.0007270034  0.0007387598 -0.004867000 -2.265880e-03 -0.0020460250
##             PFE
## 1  0.009594780
## 2  0.005254702
## 3 -0.001311252
## 4 -0.011213573
## 5 -0.003322393
## 6 -0.001665345
```

```
# Calculate the returns for SPY
SPY.returns<-as.matrix(diff(log(SPY$SPY.Adjusted))[-1])
# Calculate the returns for XLV
XLV.returns<-as.matrix(diff(log(XLV$XLV.Adjusted))[-1])
# Calculate the returns for XLI
XLI.returns<-as.matrix(diff(log(XLI$XLI.Adjusted))[-1])
```

Step 1: selection of factors Start the process of factors selection by doing PCA on the stock portfolio.

```
Stock.Portfolio.Returns.PCA <- princomp(Stock.Portfolio.Returns)
# cumsum all the standard deviations over the sum of standard deviations
cumsum(Stock.Portfolio.Returns.PCA$sdev/sum(Stock.Portfolio.Returns.PCA$sdev))
```

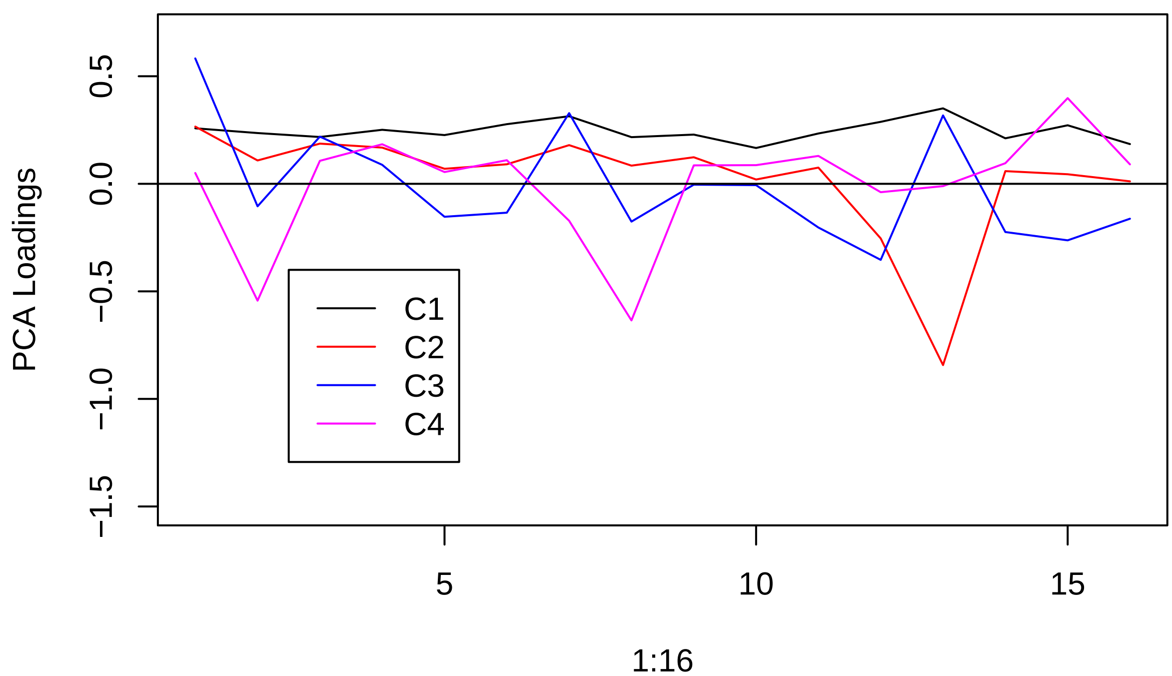
```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 0.2071119 0.3274437 0.4054387 0.4718642 0.5340896 0.5916219 0.6468114 0.6958453
##    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14   Comp.15   Comp.16
## 0.7441690 0.7860761 0.8266210 0.8654923 0.9038541 0.9383781 0.9703417 1.0000000
```

Observe the vector of cumulative explanatory power. In order to fit all the stock returns well we would need at least 13 factors(Over 90%). But in this case we will use only 4 which accounts for about 50% of variability, interpreting the rest as idiosyncratic diversifiable risk.

```r
# Create 5 factors and 5 loadings
Stock.Portfolio.Returns.PCA.factors <- as.matrix(Stock.Portfolio.Returns.PCA$scores[,1:4])
Stock.Portfolio.Returns.PCA.loadings<-Stock.Portfolio.Returns.PCA$loadings[,1:4]
Stock.Portfolio.Returns.PCA.zero.loading<-Stock.Portfolio.Returns.PCA$center
head(Stock.Portfolio.Returns.PCA.loadings)
```

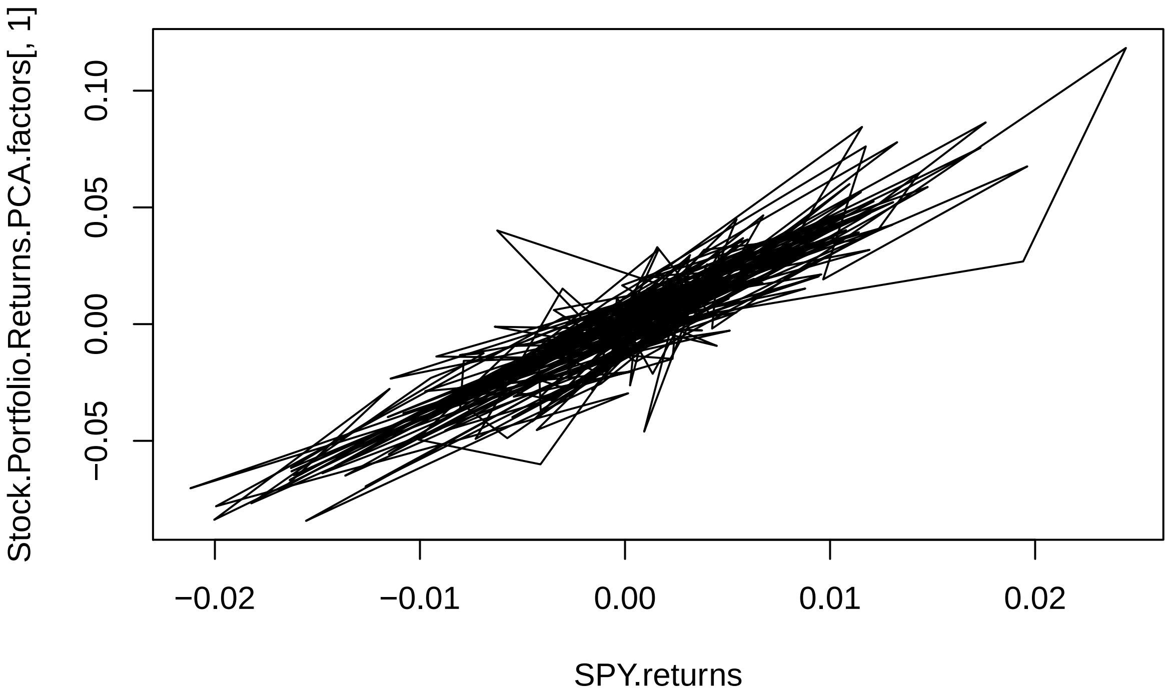
```
##        Comp.1     Comp.2     Comp.3      Comp.4
## CAT 0.2579835 0.26613878  0.5828886  0.05026304
## FDX 0.2358774 0.10857890 -0.1046452 -0.54321775
## GE  0.2176063 0.18684138  0.2192661  0.10683302
## HON 0.2510496 0.16829214  0.0881241  0.18383502
## LMT 0.2264649 0.06984808 -0.1531892  0.05474233
## NOC 0.2772688 0.09075939 -0.1342891  0.10977691
```

```r
# Plot the loadings
matplot(1:16,Stock.Portfolio.Returns.PCA.loadings,col=c('black','red','blue','magenta'),
        type="l",lty=1,ylab='PCA Loadings' , ylim = c(-1.5,0.7))
legend(2.5,-0.4,legend=c("C1","C2","C3","C4"),col=c('black','red','blue','magenta'),
       lty=1)
abline(h=0)
```



Since all 16 stocks are traded as part of the SNP500 we can say that the first component may be positively correlated to SPY. For the second component, we can see that most of the Health stocks loadings are negative, while most of the Industrial stocks loadings are positive. It means that the two sectors performed oppositely over the sample period, which is consistant with Efficient Frontier and CAPM model results. The loading plots of the 3rd and 4th components don't show much that can be interpreted.

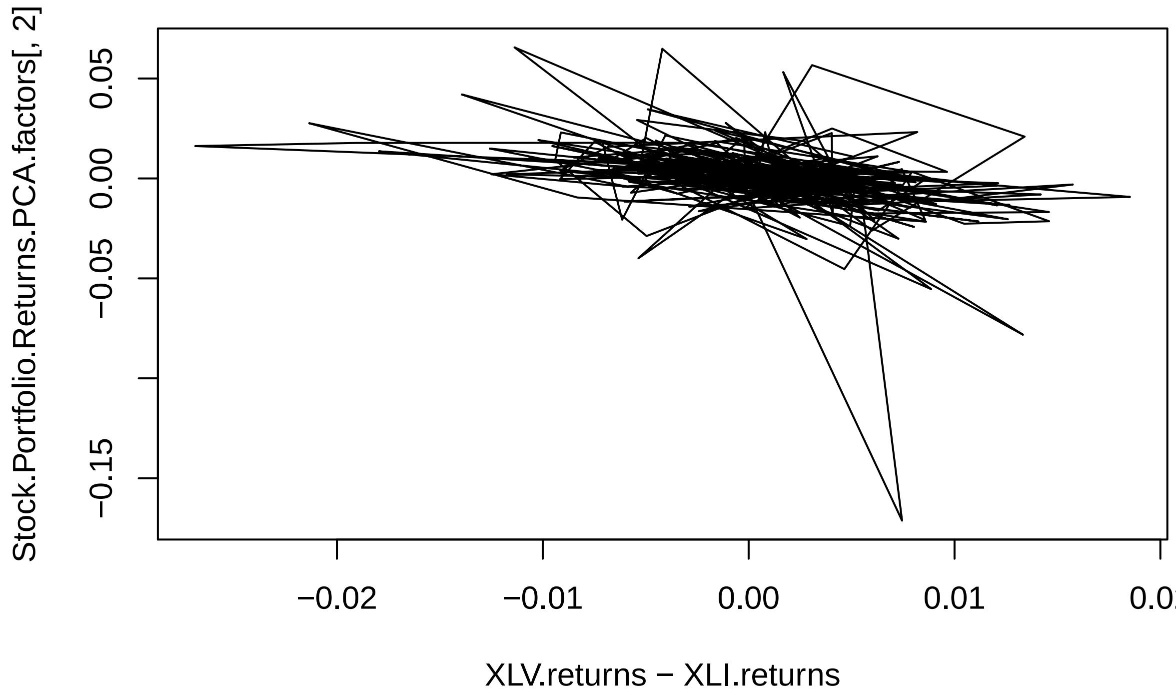
```r
# Plot First Factors with SPY returns
plot(SPY.returns,Stock.Portfolio.Returns.PCA.factors[,1],type="l")
```



From the plot, it is clear that first facor is positively correlated with SPY returns

The second loading has opposite signs for the stocks from Health sector and Industrial sector. This may mean that the second factor may be interpreted as the spread between XLV and XLI.

```r
# Plot The second factor vs the differences in the 2 sectors
plot(XLV.returns-XLI.returns, Stock.Portfolio.Returns.PCA.factors[,2], type = "l")
```



This relationship is not as strong as interpretation of the first factor.

Fit linear models explaining the interpretation of both factors.

```
lm.fit.factor1<-lm(Stock.Portfolio.Returns.PCA.factors[,1]~SPY.returns)
lm.fit.factor2<-lm(Stock.Portfolio.Returns.PCA.factors[,2]~I(XLV.returns-XLI.returns))
summary(lm.fit.factor1)
```

```
##
## Call:
## lm(formula = Stock.Portfolio.Returns.PCA.factors[, 1] ~ SPY.returns)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.049470 -0.008062  0.000629  0.008060  0.065931
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0010037  0.0008729   -1.15    0.251
## SPY.returns  3.9820621  0.1157679   34.40   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01382 on 249 degrees of freedom
## Multiple R-squared:  0.8261, Adjusted R-squared:  0.8254
## F-statistic:  1183 on 1 and 249 DF,  p-value: < 2.2e-16
```

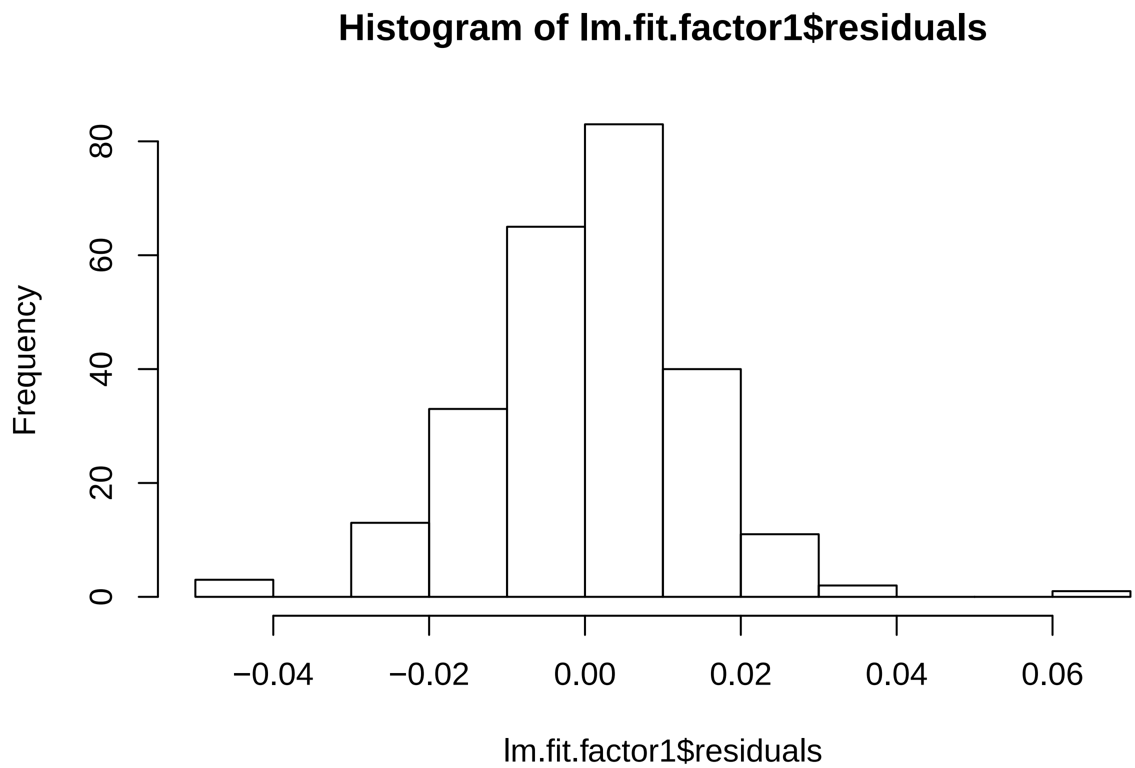
```r
summary(lm.fit.factor2)
```

```
##
## Call:
## lm(formula = Stock.Portfolio.Returns.PCA.factors[, 2] ~ I(XLV.returns -
##     XLI.returns))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.163089 -0.006035 -0.000045  0.007709  0.059480
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.0009055  0.0011183   0.810    0.419
## I(XLV.returns - XLI.returns) -1.2064368  0.1720552  -7.012  2.2e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0176 on 249 degrees of freedom
## Multiple R-squared:  0.1649, Adjusted R-squared:  0.1615
## F-statistic: 49.17 on 1 and 249 DF,  p-value: 2.203e-11
```

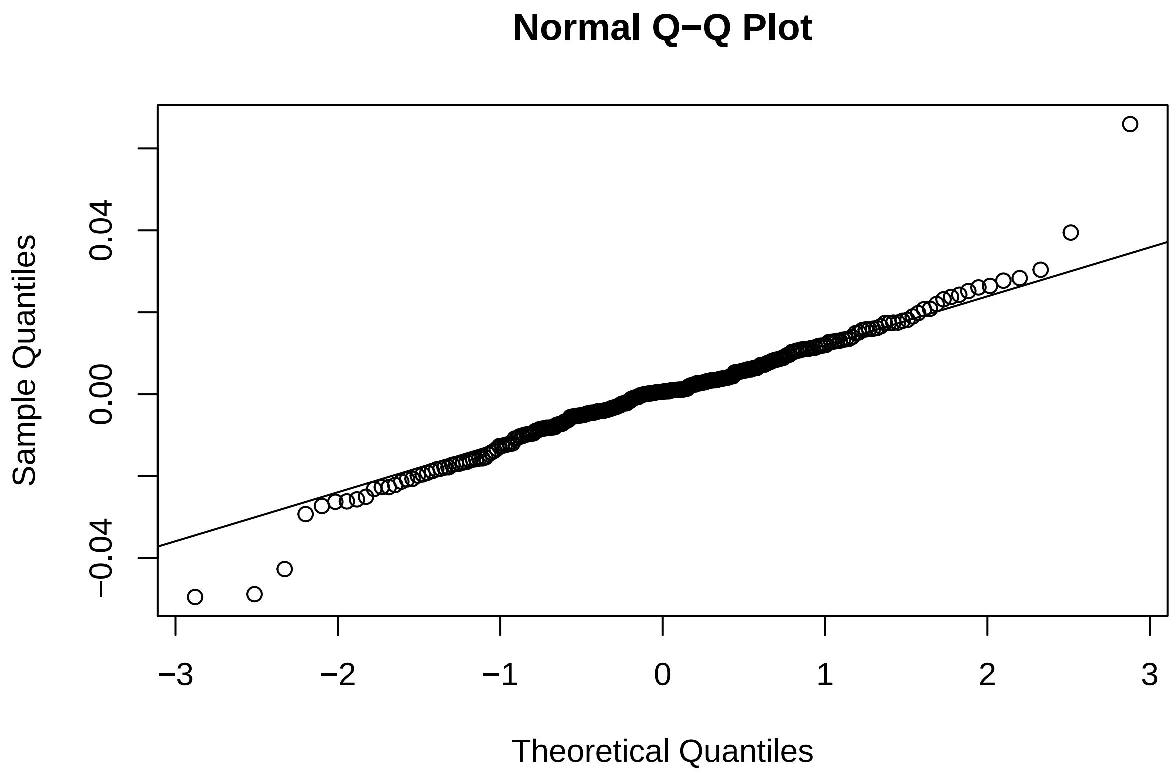
In both fits intercepts are practically insignificant, but both slopes are significant. The first factor fit has pretty good R^2, the second is not strong.

Check the residuals of both fits

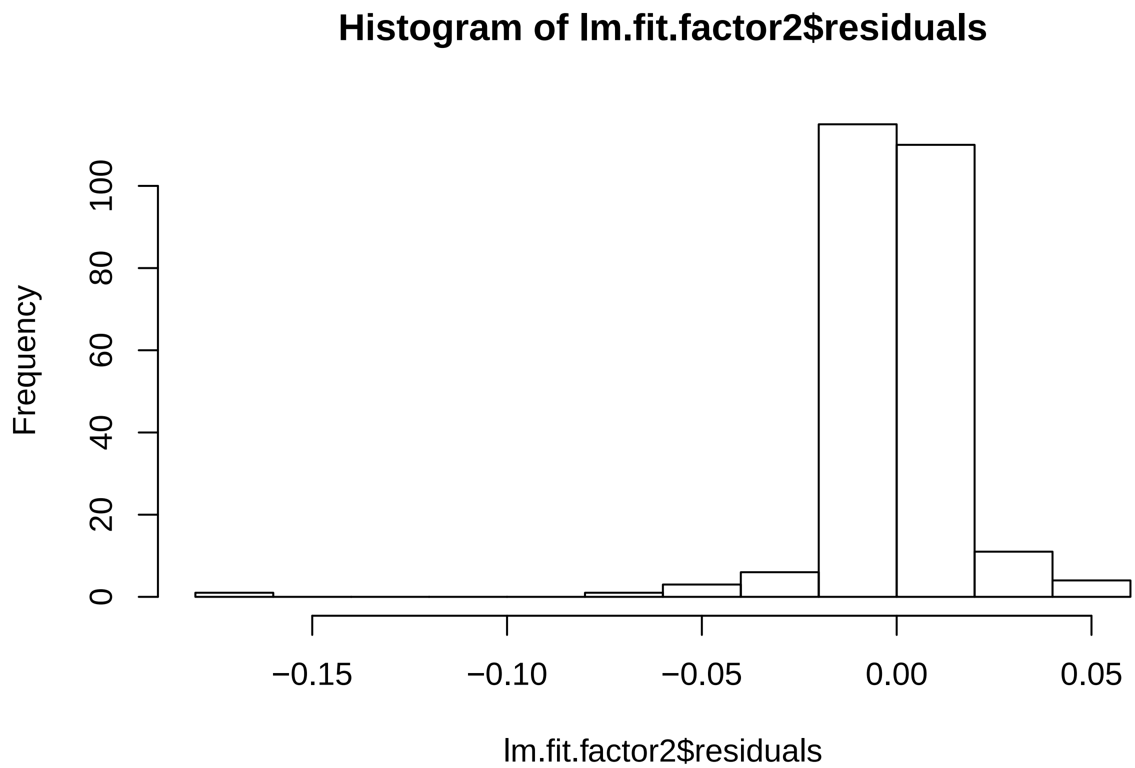
```r
# Residuals of factor 1 fit
hist(lm.fit.factor1$residuals)
```



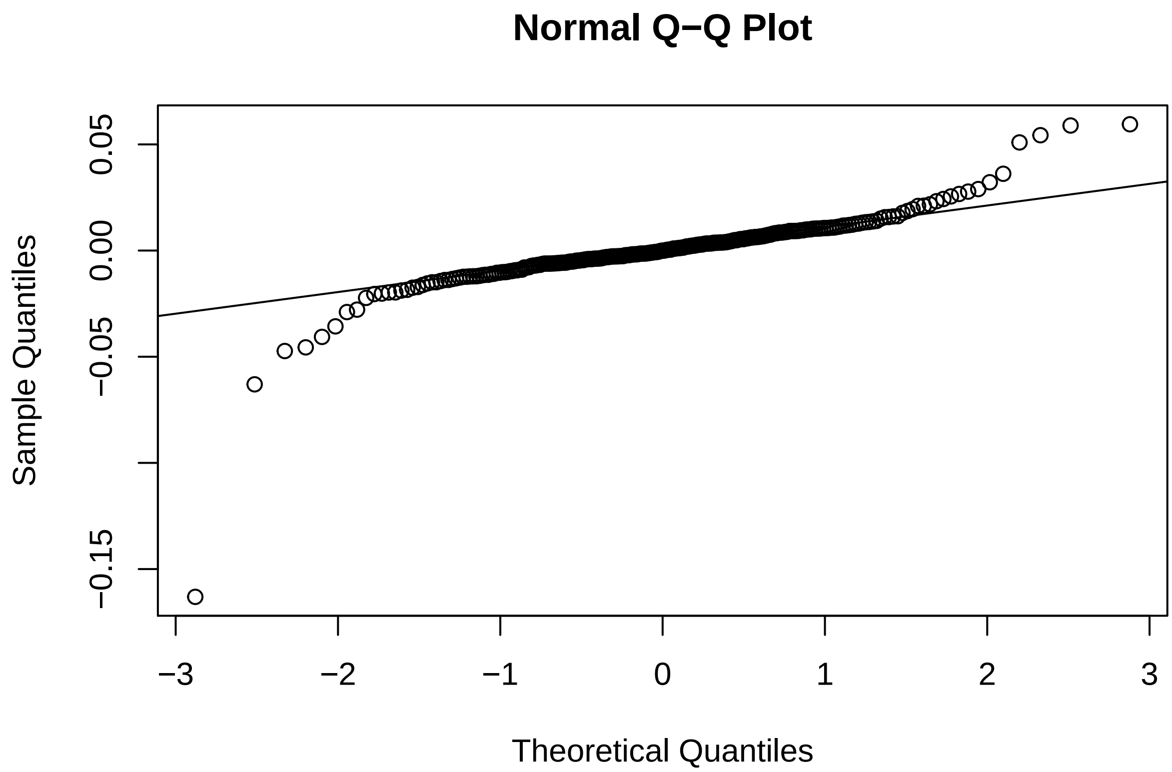

```
qqnorm(lm.fit.factor1$residuals)
qqline(lm.fit.factor1$residuals)
```




```r
# Residuals of factor 2 fit
hist(lm.fit.factor2$residuals)
```




```
qqnorm(lm.fit.factor2$residuals)
qqline(lm.fit.factor2$residuals)
```



Step 2 Estimation of betas

```
# Check that betas are given by the PCA factor loadings
Stock.portfolio.betas<-apply(Stock.Portfolio.Returns,2,
                              function(z) lm(z~Stock.Portfolio.Returns.PCA.factors[,1]+
                             Stock.Portfolio.Returns.PCA.factors[,2]
                             +Stock.Portfolio.Returns.PCA.factors[,3]
                             +Stock.Portfolio.Returns.PCA.factors[,4]
                             )$coefficients)

rownames(Stock.portfolio.betas)<-c("Alpha","Factor.1","Factor.2","Factor.3","Factor.4")
Stock.portfolio.betas<-as.data.frame(t(Stock.portfolio.betas))
Stock.portfolio.betas
```

```
##             Alpha  Factor.1    Factor.2     Factor.3    Factor.4
## CAT -8.847117e-04 0.2579835  0.26613878  0.582888590  0.05026304
## FDX  4.650241e-04 0.2358774  0.10857890 -0.104645205 -0.54321775
## GE   1.722812e-04 0.2176063  0.18684138  0.219266071  0.10683302
## HON  3.826453e-04 0.2510496  0.16829214  0.088124101  0.18383502
## LMT  7.349643e-04 0.2264649  0.06984808 -0.153189240  0.05474233
## NOC  1.175922e-03 0.2772688  0.09075939 -0.134289113  0.10977691
## UNP -1.184947e-04 0.3140942  0.17969030  0.328550707 -0.17185846
## UPS -1.268327e-04 0.2168737  0.08441479 -0.175650237 -0.63505561
## UTX -8.820731e-05 0.2287045  0.12335226 -0.003969637  0.08572903
```

```
## WM   2.543309e-04 0.1664938  0.01990691 -0.006198171  0.08664931
## ABT  7.829841e-04 0.2339982  0.07530876 -0.203273285  0.12958125
## AET  1.797413e-03 0.2879983 -0.25380621 -0.353540420 -0.03893420
## HUM  1.588894e-03 0.3507183 -0.84282284  0.317955072 -0.01082275
## JNJ -2.203000e-04 0.2115451  0.05889032 -0.224219345  0.09579692
## MDT  6.614428e-04 0.2719824  0.04437017 -0.262814652  0.39835544
## PFE  5.697635e-04 0.1847189  0.01140884 -0.162300360  0.09030429
```

```
cbind(zeroLoading=Stock.Portfolio.Returns.PCA.zero.loading,
      Stock.Portfolio.Returns.PCA.loadings[,1:4])
```

```
##       zeroLoading    Comp.1      Comp.2       Comp.3      Comp.4
## CAT -8.847117e-04 0.2579835  0.26613878  0.582888590  0.05026304
## FDX  4.650241e-04 0.2358774  0.10857890 -0.104645205 -0.54321775
## GE   1.722812e-04 0.2176063  0.18684138  0.219266071  0.10683302
## HON  3.826453e-04 0.2510496  0.16829214  0.088124101  0.18383502
## LMT  7.349643e-04 0.2264649  0.06984808 -0.153189240  0.05474233
## NOC  1.175922e-03 0.2772688  0.09075939 -0.134289113  0.10977691
## UNP -1.184947e-04 0.3140942  0.17969030  0.328550707 -0.17185846
## UPS -1.268327e-04 0.2168737  0.08441479 -0.175650237 -0.63505561
## UTX -8.820731e-05 0.2287045  0.12335226 -0.003969637  0.08572903
## WM   2.543309e-04 0.1664938  0.01990691 -0.006198171  0.08664931
## ABT  7.829841e-04 0.2339982  0.07530876 -0.203273285  0.12958125
## AET  1.797413e-03 0.2879983 -0.25380621 -0.353540420 -0.03893420
## HUM  1.588894e-03 0.3507183 -0.84282284  0.317955072 -0.01082275
## JNJ -2.203000e-04 0.2115451  0.05889032 -0.224219345  0.09579692
## MDT  6.614428e-04 0.2719824  0.04437017 -0.262814652  0.39835544
## PFE  5.697635e-04 0.1847189  0.01140884 -0.162300360  0.09030429
```

Step 3. Estimation of market price of risk

```
Market.Prices.of.risk.fit<-lm(I(Alpha-Mean.FedFunds)~.-1,data=Stock.portfolio.betas)
summary(Market.Prices.of.risk.fit)
```

```
##
## Call:
## lm(formula = I(Alpha - Mean.FedFunds) ~ . - 1, data = Stock.portfolio.betas)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -8.620e-04 -2.312e-04  1.978e-05  2.268e-04  5.700e-04
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## Factor.1  0.0019499  0.0003979   4.901 0.000365 ***
## Factor.2 -0.0016828  0.0003979  -4.230 0.001169 **
## Factor.3 -0.0012868  0.0003979  -3.234 0.007160 **
## Factor.4  0.0003845  0.0003979   0.966 0.352875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

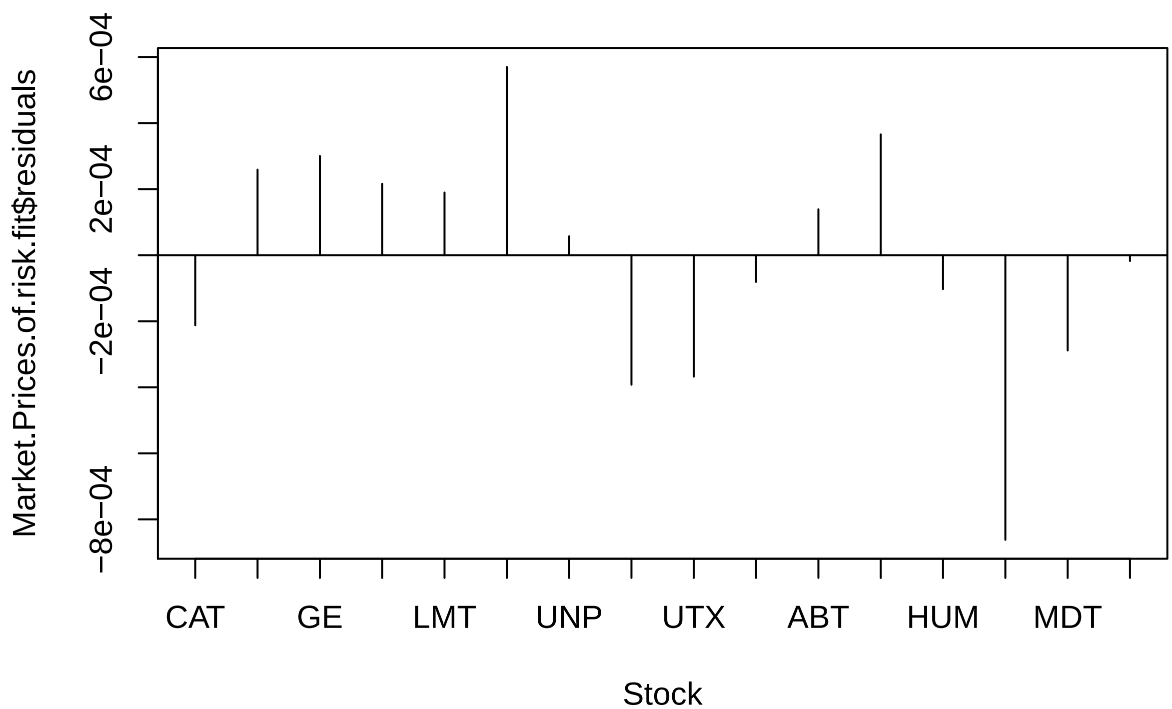
```
## Residual standard error: 0.0003979 on 12 degrees of freedom
## Multiple R-squared:  0.8163, Adjusted R-squared:  0.755
## F-statistic: 13.33 on 4 and 12 DF,  p-value: 0.000227
```

The R^2 and adjusted R^2 are not bad. Slope parameters are significant for Factor 1, 2 and 3 at 5% level. The F test can reject the utility hypothesis with 5% level. Note that slopes for factor 4 are not significant.

```
# Look at coefficients of the factors from the linear model
#Note: Intercept is the expected value of Risk Free Returns
Market.Prices.of.risk<-c(Mean.FedFunds,Market.Prices.of.risk.fit$coefficients)
Market.Prices.of.risk
```

```
##                    Factor.1      Factor.2      Factor.3      Factor.4
##  2.969124e-06  1.949924e-03 -1.682805e-03 -1.286838e-03  3.845239e-04
```

```
# Plot the residuals
plot(Market.Prices.of.risk.fit$residuals,type="h",xaxt="n",xlab="Stock")
abline(h=0)
axis(1, at=1:16, labels=SP500.IH.names)
```



When residual is positive it means that the mean excess return of the stock over the sample period is greater than predicted value, it means that the stock outperformed the predicted value of the model over the sample period and it is undervalued. When the residual is negative, it means that the stock gave less return than the model predicted value and it is overvalued by the model.

I noticed that if we only use the first two component loadings to generate market price of risk, the model residual diagram gives a much closer result to the Efficient frontier and CAPM models.

Fitting APT model using only the first two principle components

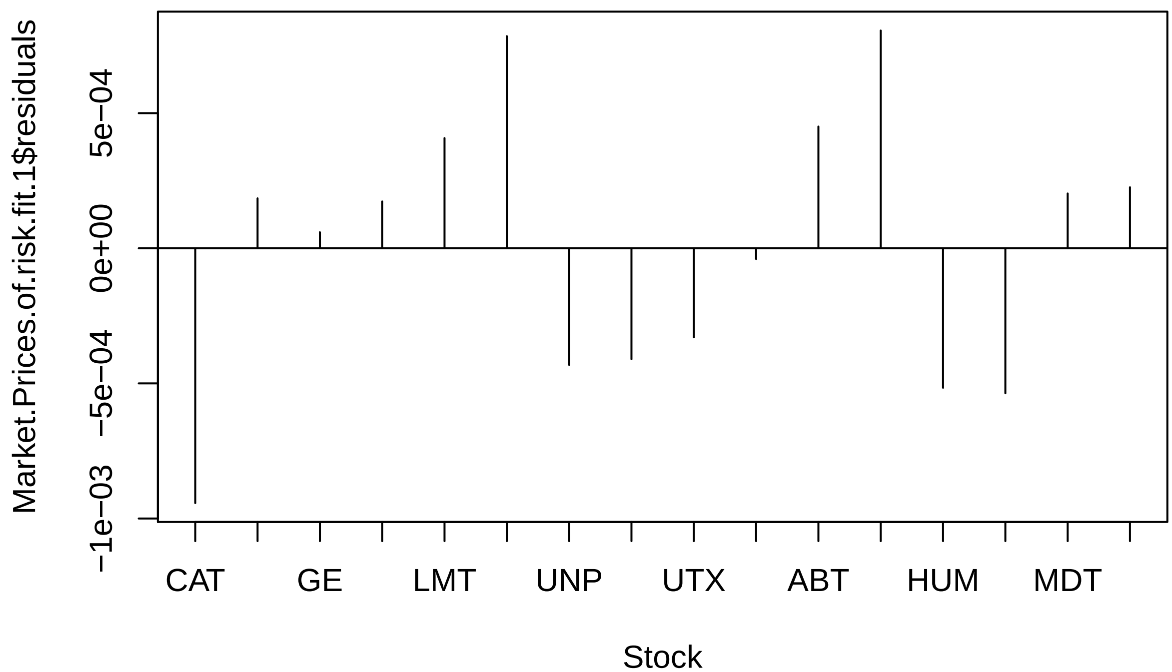
```
Stock.portfolio.betas.1<-apply(Stock.Portfolio.Returns,2,
                               function(z) lm(z~Stock.Portfolio.Returns.PCA.factors[,1]+
                              Stock.Portfolio.Returns.PCA.factors[,2]
                              )$coefficients)

rownames(Stock.portfolio.betas.1)<-c("Alpha","Factor.1","Factor.2")
Stock.portfolio.betas.1<-as.data.frame(t(Stock.portfolio.betas.1))
Stock.portfolio.betas.1
```

```
##             Alpha  Factor.1    Factor.2
## CAT -8.847117e-04 0.2579835  0.26613878
## FDX  4.650241e-04 0.2358774  0.10857890
## GE   1.722812e-04 0.2176063  0.18684138
## HON  3.826453e-04 0.2510496  0.16829214
## LMT  7.349643e-04 0.2264649  0.06984808
## NOC  1.175922e-03 0.2772688  0.09075939
## UNP -1.184947e-04 0.3140942  0.17969030
## UPS -1.268327e-04 0.2168737  0.08441479
## UTX -8.820731e-05 0.2287045  0.12335226
## WM   2.543309e-04 0.1664938  0.01990691
## ABT  7.829841e-04 0.2339982  0.07530876
## AET  1.797413e-03 0.2879983 -0.25380621
## HUM  1.588894e-03 0.3507183 -0.84282284
## JNJ -2.203000e-04 0.2115451  0.05889032
## MDT  6.614428e-04 0.2719824  0.04437017
## PFE  5.697635e-04 0.1847189  0.01140884
```

```
cbind(zeroLoading=Stock.Portfolio.Returns.PCA.zero.loading,
      Stock.Portfolio.Returns.PCA.loadings[,1:2])
```

```
##       zeroLoading    Comp.1      Comp.2
## CAT -8.847117e-04 0.2579835  0.26613878
## FDX  4.650241e-04 0.2358774  0.10857890
## GE   1.722812e-04 0.2176063  0.18684138
## HON  3.826453e-04 0.2510496  0.16829214
## LMT  7.349643e-04 0.2264649  0.06984808
## NOC  1.175922e-03 0.2772688  0.09075939
## UNP -1.184947e-04 0.3140942  0.17969030
## UPS -1.268327e-04 0.2168737  0.08441479
## UTX -8.820731e-05 0.2287045  0.12335226
## WM   2.543309e-04 0.1664938  0.01990691
## ABT  7.829841e-04 0.2339982  0.07530876
## AET  1.797413e-03 0.2879983 -0.25380621
## HUM  1.588894e-03 0.3507183 -0.84282284
## JNJ -2.203000e-04 0.2115451  0.05889032
## MDT  6.614428e-04 0.2719824  0.04437017
## PFE  5.697635e-04 0.1847189  0.01140884
```

```
Market.Prices.of.risk.fit.1<-lm(I(Alpha-Mean.FedFunds)~.-1,data=Stock.portfolio.betas.1)
plot(Market.Prices.of.risk.fit.1$residuals,type="h",xaxt="n",xlab="Stock")
abline(h=0)
axis(1, at=1:16, labels=SP500.IH.names)
```



Based on the residual diagram of the fitted model using only the first two pricinple components, we can see that all the stocks with negative residuals are the same stocks below the security market line from the EF and CAPM models. Those are the overvalued stocks. The two PC APT model residual diagram also revealed that health sector stock returns outperformed the model prediction and industrial sector stocks generated worse values than the model prediction, which is consistant with the results of XLV and XLI from the EF and CAPM models.

Based on the previous analysis, only the first two PCs can be related to broader market factors. Adding the 3rd and 4th components may make the APT model worse. Because those additional components do not have any connection with the broader markets and could just be idiosyncratic variance of the first two Principal components.