Empirical Methods in Finance - Assignment 3

Nitish Ramkumar, Prasanth Kumar, Ian Laker

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.

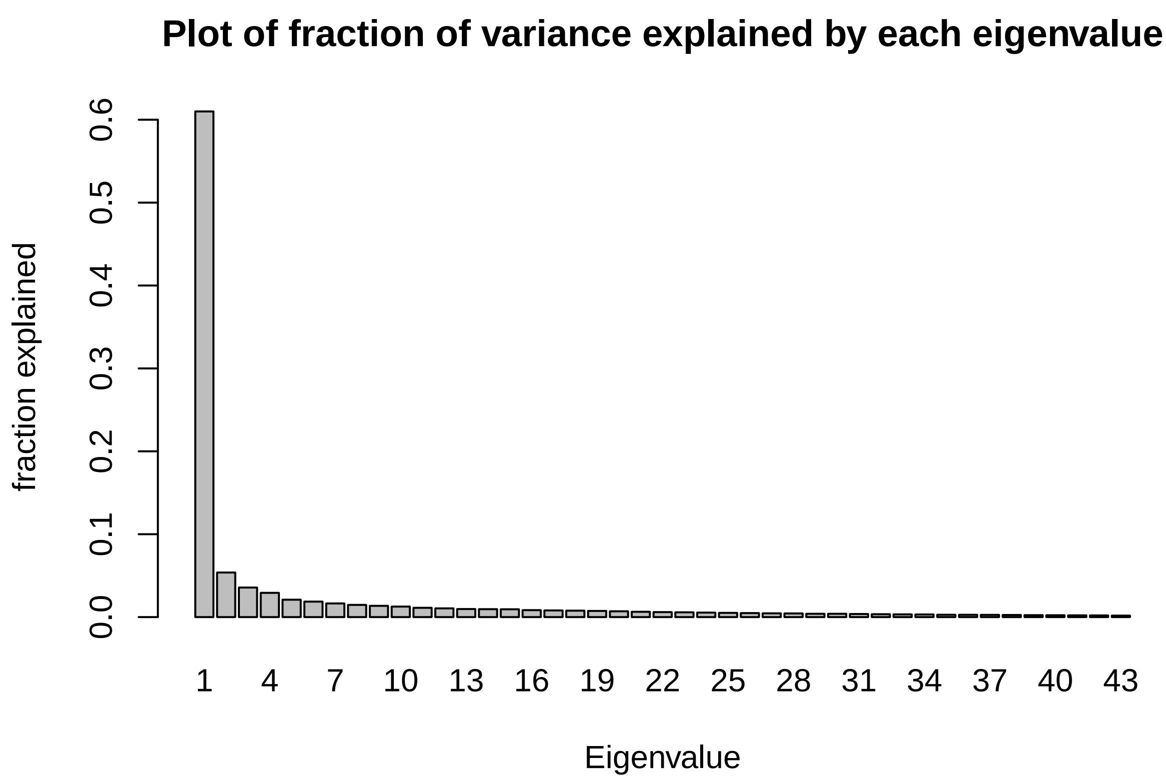
Principal Component Analysis

1

The eigenvalues of the variance-covariance matrix can be calculated using **eigen** R function

```
##  [1] 1061.782182   93.547500   62.062228   50.913744   36.527472
##  [6]   32.466362   28.690390   25.586583   23.620709   22.044200
## [11]   19.487297   18.334225   16.900536   16.642947   16.397853
## [16]   14.611519   13.972664   13.514532   12.933193   12.179409
## [21]   11.168473   10.452042   10.040775    9.384985    8.838352
## [26]    8.401733    7.902281    7.664925    6.961698    6.917135
## [31]    6.513456    6.059360    5.730276    5.475691    4.890513
## [36]    4.799644    4.637173    4.416646    4.031374    3.892260
## [41]    3.640982    3.481827    3.080961
```

The fraction explained by each eigen value can be seen in this graph.



2a

The largest 3 Principal Components explain **69.94%** of the total variance

2b

The Principial Components can be calculated using this formula

$y_{it} = e_i`r_t = \sum_{j=1}^{N} e_{ij} r_{jt}$

where y_{it} is the i^{th} principal component at time t.
e_{ij} is the weight of the j^{th} asset in the i^{th} eigenvector.
r_{jt} is the return of the j^{th} asset at time t.

Mean sample returns for these 3 factor portfolios are

```
##       PCA1       PCA2       PCA3
## -3.7787500  0.2156532 -0.5140327
```

sample standard deviation for these 3 factor portfolios are

```
##      PCA1      PCA2      PCA3
## 32.584999  9.671996  7.877958
```

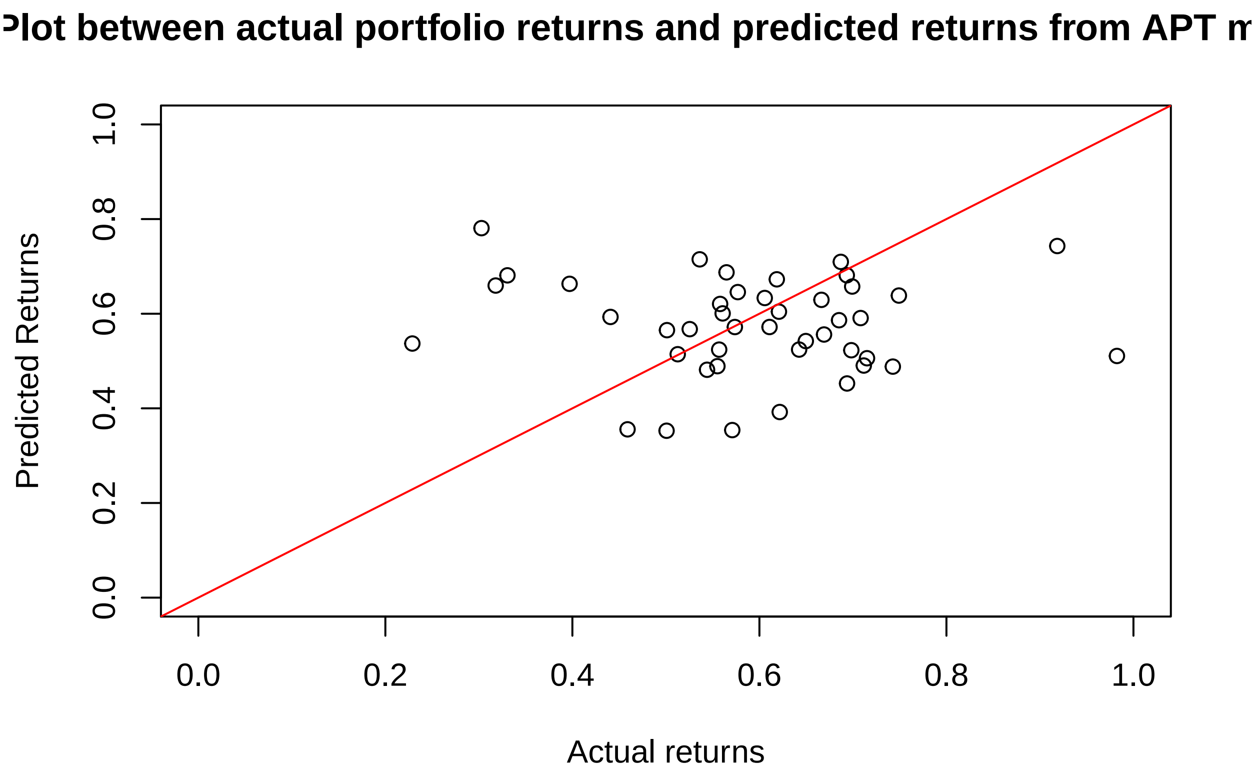
Correlation for these 3 factor portfolios are

```
##               PCA1         PCA2          PCA3
## PCA1  1.000000e+00 2.412379e-16 -1.471233e-16
## PCA2  2.412379e-16 1.000000e+00  1.937757e-16
## PCA3 -1.471233e-16 1.937757e-16  1.000000e+00
```

2c

The loadings for each industry in the case will be equal to the weights of the industry in each of the eigen vectors. The formula $\sqrt{\lambda_i} e_i$ (λ_i is the eigen value and e_i is the eigen vector weights) holds good only if the data is standardized (divided by standard deviation).

For calculating this predicted value, the PCA value is used along with the loading factor (weights in this case). This is done for every industry.



Though some points were close to the 45 degree line, most of the points were dispersed away from the 45 degree line.

The loading was also retrieved out of regression and compared with the loading out of eigen vectors and were found to be same (testing in the code appended at the end)

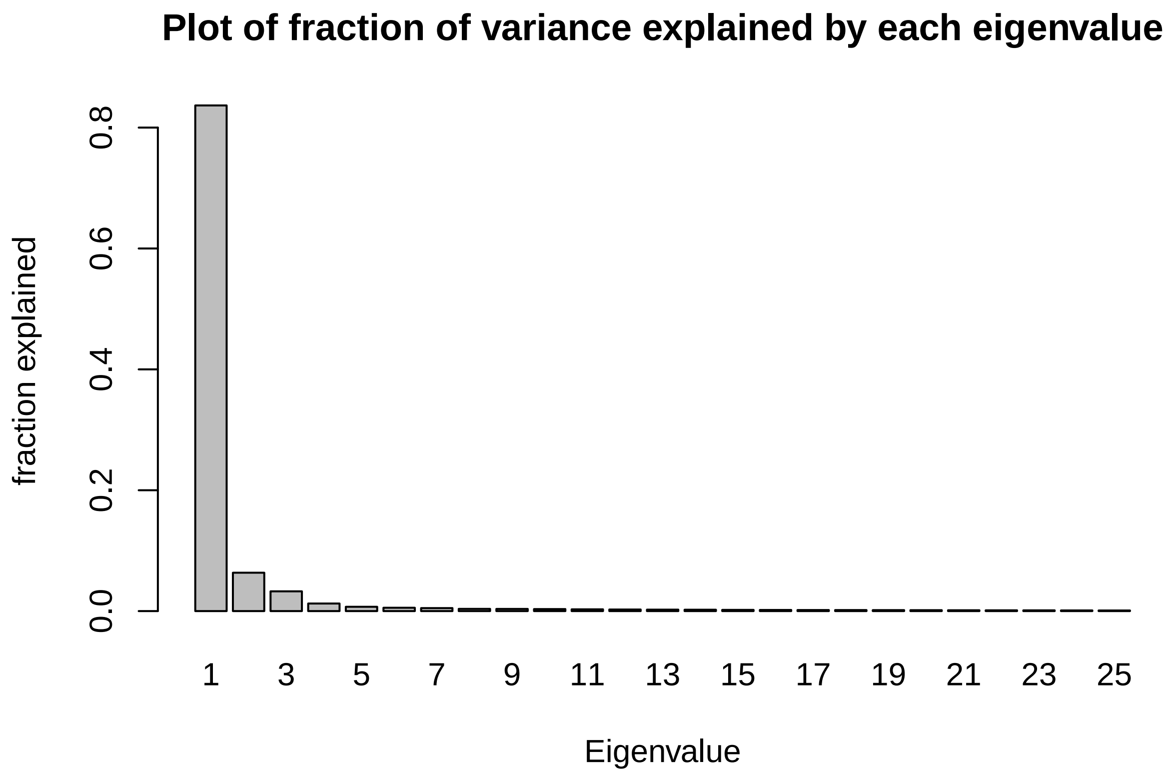
2d

Cross sectional R^2 = **-0.5738903**. R^2 is negative because we don't consider an α (intercept) in this case. Due to this reason, this formula is not valid calculation method for R^2 in this case.

Generally we expect α to be equal to 0, as the factors are traded. But due to the negative R^2, in this we can say that α is not 0.

3a

The plot of variance explained by every eigen value for the 25 F-F portfolios is as below



3b

The first 5 PCA components explains as much as 95% of the data (information got out of the cumulative proportion in the summary view of princomp.

Arbitrage Pricing in Factor Models

1a

The exposures of portfolio A involves 0.5 unit exposure to factor 1 and 0.75 unit exposure to factor 2. So to get the arbitrage opportunity,

We should go long the portfolio and short the assets which replicates the portfolio exposure (0.5 unit of factor 1 and 0.75 unit of factor 2).

By doing this we will get 1% extra return (profit) out of the portfolio.

1b

δ should be equal to -1% to avoid arbitrage opportunities.

Expected value of ${R^e}_{A,t} = 0.5E(R_{f1,t}) + 0.75E(R_{f2,t})$

$= (0.5)(6\%) + (0.75)(-2\%) = \mathbf{1.5\%}$

R Code

```
#Data Retrieval
suppressMessages(library(xts))
portfolio.data <- read.csv("48_Industry_Portfolios.csv",header=TRUE,sep = ","
                           ,stringsAsFactors = FALSE,skip = 11,nrows = 1086)
portfolio.data$X <- as.yearmon(as.character(portfolio.data$X),format="%Y%m")
portfolio.data <- xts(portfolio.data[,-1],order.by = portfolio.data$X)
factors.data <- read.csv("F-F_Research_Data_Factors.csv",header=TRUE,sep = ","
                         ,stringsAsFactors = FALSE,skip = 3,nrows=1086)
factors.data$X <- as.yearmon(as.character(factors.data$X),format="%Y%m")
factors.data <- xts(factors.data[,-1],order.by = factors.data$X)

#Date and 99 constraints
portfolio.data <- portfolio.data[index(portfolio.data)>="1960-01-01" &
                                   index(portfolio.data)<="2015-12-31",]
portfolio.invalidColumns <- apply(portfolio.data,2
                                  ,function(x){sum(x %in% -99.99) > 0})
portfolio.data <- portfolio.data[,!portfolio.invalidColumns]
factors.data <- factors.data[index(factors.data)>="1960-01-01"
                             & index(factors.data)<="2015-12-31",]

#Calculate Excess Returns
portfolio.excessReturns <- apply(portfolio.data,2,function(x){t(x - factors.data$RF)})

#1A
eigen.info <- eigen(cov(portfolio.excessReturns))
eigen.info$values

#1B
eigen.fractionExplained <- sapply(eigen.info$values,
                                  function(x){x/sum(eigen.info$values)})
barplot(eigen.fractionExplained,names.arg = 1:length(eigen.fractionExplained)
        ,main = "Plot of fraction of variance explained by each eigenvalue",
        xlab = "Eigenvalue",ylab="fraction explained")

#2A
#Take top 3 eigen vectors
eigen.significantPcVector <- eigen.info$vectors[,c(1,2,3)]
eigen.significantfraction <- eigen.fractionExplained[c(1,2,3)]
sum(eigen.significantfraction)

#2B

#multiply weights by the corresponding industry returns
pcas <- apply(eigen.significantPcVector,2,function(eigenVec)
  {apply(portfolio.excessReturns,1,function(returnsTime){returnsTime%*%eigenVec})})
colnames(pcas) <- c("PCA1","PCA2","PCA3")

pcas.mean <- apply(pcas,2,mean)
pcas.sd <- apply(pcas,2,sd)
pcas.cor <- cor(pcas)

#2C
```

```r
#Test to check if regression coefficients match the eigen loadings
##(in this case it is the weights itself)
summary(LmOutput) <- lm(portfolio.excessReturns[,2] ~ pcas[,1] + pcas[,2] + pcas[,3])
EigenOutput <- eigen.significantPcVector[1,]
##LmOutput$coefficients[-1] == EigenOutput

#Get the actual average of all industries
portfolio.actual.avg <- apply(portfolio.excessReturns,2,mean)
#use that to calculate predicted value
portfolio.predicted <- portfolio.actual.avg%*%eigen.significantPcVector%*%
  t(eigen.significantPcVector)
#take average across time
portfolio.predicted.avg <- apply(portfolio.predicted,2,mean)
#plot actual vs predicted
plot(portfolio.actual.avg,portfolio.predicted.avg,xlab="Actual returns",
     ylab="Predicted Returns",xlim = c(0,1),ylim=c(0,1),
     main="Plot between actual portfolio returns and predicted returns from APT model")
abline(0,1,col="red")

#2D
#RCross section calculation
numerator <- var(portfolio.actual.avg - portfolio.predicted.avg)
Rcrosssection <- 1 - (numerator/var(portfolio.actual.avg))

#3A
#25 FF portfolios
portfolio.25.data <- read.csv("25_Portfolios_5x5.csv",header=TRUE,sep = ","
                              ,stringsAsFactors = FALSE,skip = 19,nrows = 1086)
portfolio.25.data$X <- as.yearmon(as.character(portfolio.25.data$X),format="%Y%m")
portfolio.25.data <- xts(portfolio.25.data[,-1],order.by = portfolio.25.data$X)

#Date and 99 constraints
portfolio.25.data <- portfolio.25.data[index(portfolio.25.data)>="1960-01-01"
                                       & index(portfolio.25.data)<="2015-12-31",]
portfolio.25.invalidColumns <- apply(portfolio.25.data,2,
                                     function(x){sum(x %in% -99.99) > 0})
portfolio.25.data <- portfolio.25.data[,!portfolio.25.invalidColumns]
portfolio.25.excessReturns <- apply(portfolio.25.data,2
                                    ,function(x){t(x - factors.data$RF)})

eigen.25.Info <- eigen(cov(portfolio.25.excessReturns))
eigen.25.fractionExplained <- sapply(eigen.25.Info$values
                                     ,function(x){x/sum(eigen.25.Info$values)})
barplot(eigen.25.fractionExplained,names.arg = 1:length(eigen.25.fractionExplained)
        ,main = "Plot of fraction of variance explained by each eigenvalue"
        ,xlab = "Eigenvalue",ylab="fraction explained")


#3B
eigen.25.pcaInfo <- princomp(portfolio.25.excessReturns)
sum <- summary(eigen.25.pcaInfo)
```