Machine Learning Homework 1

Jin Miao (JMiao18)

February 13, 2018

Task 1

Create a data-frame historical daily total returns from January 1st 2000 to December 31st 2016. Descriptive Statistics are shown as follows:

```
mlfin = read.csv(file = "M:/A Master of Science in Marketing Sciences/MS Machine Learning/b8142c
7257d20c44.csv", header = TRUE)

attach(mlfin)
mlfin$date = as.character(mlfin$date)
mlfin$date <- as.Date(mlfin$date, "%m/%d/%Y")
mlfin$RET = as.character(mlfin$RET)
mlfin$RET = as.numeric(mlfin$RET)

threshold1 = as.Date("12/31/2005","%m/%d/%Y")
threshold2 = as.Date("12/31/2010","%m/%d/%Y")

train1 = subset(mlfin, date <= threshold1)
train2 = subset(mlfin, date <= threshold2 & date > threshold1)
test = subset(mlfin, date > threshold2)

library(tidyr)
newtrain1 = subset(train1, select = -c(PERMNO))
train1wide = spread(newtrain1, TICKER, RET)
newtrain2 = subset(train2, select = -c(PERMNO))
train2wide = spread(newtrain2, TICKER, RET)
newtest = subset(test, select = -c(PERMNO))
testwide = spread(newtest, TICKER, RET)
rm(train1,train2,test,newtrain1,newtest,newtrain2)
```

```
summary(mlfin)
```

```
##      PERMNO           date                TICKER           RET          
##  Min.   :10107   Min.   :2000-01-03   AA     : 4277   Min.   :-0.3902440  
##  1st Qu.:12490   1st Qu.:2004-03-01   AXP    : 4277   1st Qu.:-0.0081240  
##  Median :21573   Median :2008-05-27   BA     : 4277   Median : 0.0003040  
##  Mean   :28838   Mean   :2008-06-08   C      : 4277   Mean   : 0.0003735  
##  3rd Qu.:43449   3rd Qu.:2012-09-12   CAT    : 4277   3rd Qu.: 0.0088180  
##  Max.   :70519   Max.   :2016-12-30   DD     : 4277   Max.   : 0.5782490  
##                                       (Other):87063   NA's   :2
```

In summary, for each company, we have 4277 daily observations, starting from 2000-01-03 and ending at 2016-12-30. The best return is 57.82% and the worst one is -39.02%. RET has two missing values.

Task 2:

The companies in Training Set 1 is (the first variable in output denotes the date)

```
##  [1] "date" "AA"   "AXP"  "BA"   "C"    "CAT"  "DD"   "DIS"  "GE"   "HD"
## [11] "HON"  "HPQ"  "HWP"  "IBM"  "INTC" "IP"   "JNJ"  "KO"   "MCD"  "MMM"
## [21] "MO"   "MRK"  "MSFT" "PG"   "SBC"  "T"    "UTX"  "WMT"  "XOM"
```

The companies in Training Set 2 is (the first variable in output denotes the date)

```
##  [1] "date" "AA"   "AXP"  "BA"   "C"    "CAT"  "DD"   "DIS"  "GE"   "HD"
## [11] "HON"  "HPQ"  "IBM"  "INTC" "IP"   "JNJ"  "KO"   "MCD"  "MMM"  "MO"
## [21] "MRK"  "MSFT" "PG"   "T"    "UTX"  "WMT"  "XOM"
```

The companies in Test Set is (the first variable in output denotes the date)

```
##  [1] "date" "AA"   "ARNC" "AXP"  "BA"   "C"    "CAT"  "DD"   "DIS"  "GE"
## [11] "HD"   "HON"  "HPQ"  "IBM"  "INTC" "IP"   "JNJ"  "KO"   "MCD"  "MMM"
## [21] "MO"   "MRK"  "MSFT" "PG"   "T"    "UTX"  "WMT"  "XOM"
```

By comparison, we can find that “HWP” “SBC” were excluded from Dow Jones in Training Set 2 (Jan 1st, 2006), and that “ARNC” are added into Test Set (Jan 1st, 2011). In order to make predictions, “HWP” “SBC” are excluded from Training Set 1 and “ARNC” is excluded from Test Set.

```
train1wide = subset(train1wide, select = -c(HWP,SBC))
testwide = subset(testwide, select = -c(ARNC))
```

(a) Perform PCA on the stock returns in the Training Set 1. Print the Principal Component loadings you calculated.

First, I need to deal wth missing values. There are 585 missing values for HP due to ticker change at May 2, 2002. I searched for the stock data with its former ticker “HWP” from Jan 1, 2000 to May 2, 2002.

After this major change, the number of missing values in Training Set 1 is

```
train1_data = train1wide[,2:length(train1wide[1,])]
sum(is.na(train1_data))
```

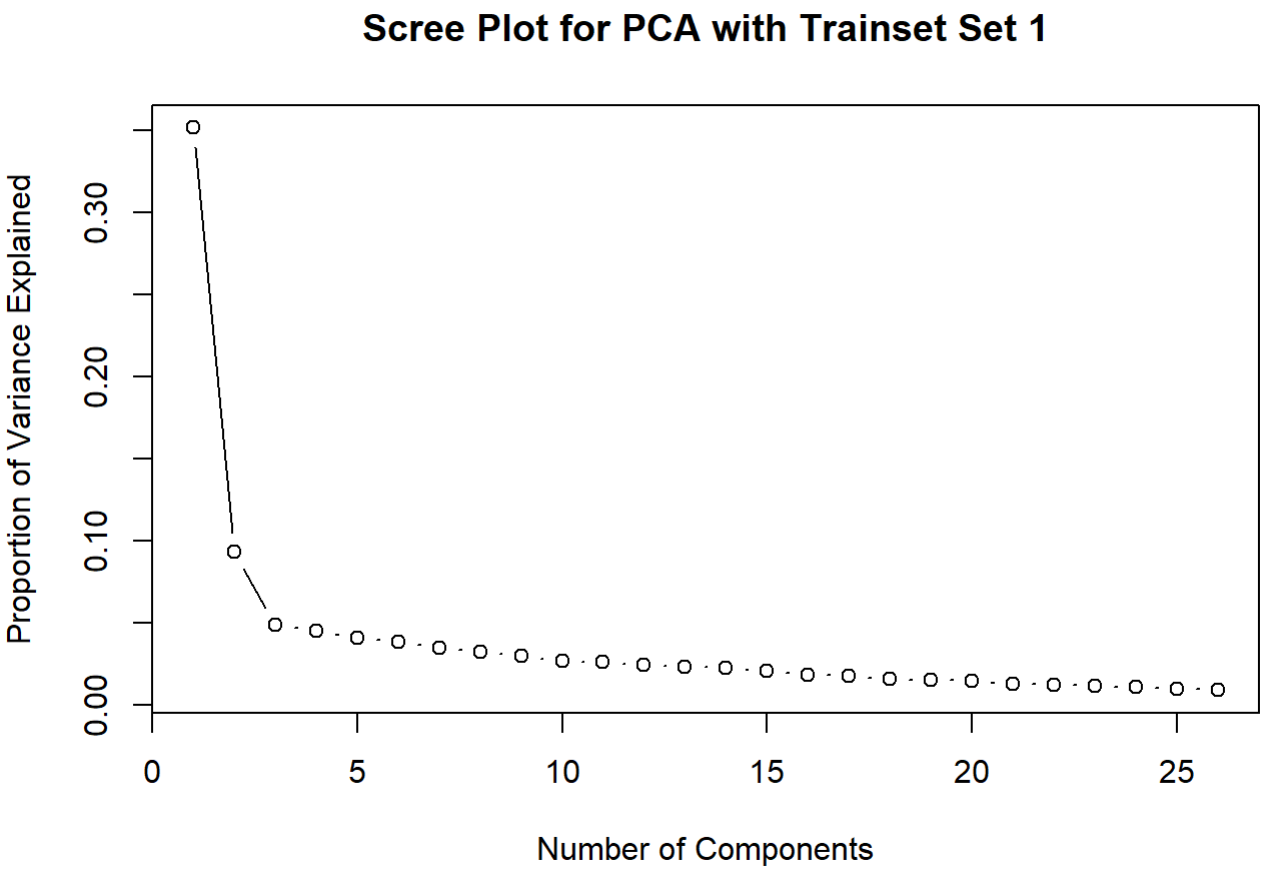
```
## [1] 7
```

I choose to replace these missing values with the mean return. At this time, missing data take up less than 0.5% of the total raining Set 1, the possible bias incurred by this practice is negligible.

```
NA2mean <- function(x) replace(x, is.na(x), mean(x, na.rm = TRUE))
train1_data = replace(train1_data, TRUE, lapply(train1_data, NA2mean))
```

Then I use “prcomp” function in R to conduct Principal Component Analysis. Given that all variables are stock returns, which are comparable to each other, I do not use “scale” option to standardize the data. The Scree Plot can be shown as follows:

```
tr1pca = prcomp(train1_data, scale = FALSE)
tr1pcaVar = tr1pca$sdev^2
tr1pve = tr1pcaVar/sum(tr1pcaVar)
plot(tr1pve, type = "b", main = "Scree Plot for PCA with Trainset Set 1", ylab = "Proportion of
 Variance Explained", xlab = "Number of Components")
```



Based on the explained variance, I choose the first three principal components, whose loading matrix is shown as follows:

```
tr1rot = tr1pca$rotation
tra1loading = tr1rot[,1:3]
round(tra1loading,4)
```

```
##         PC1     PC2     PC3
## AA   0.2368  0.1874 -0.0468
## AXP  0.2416  0.0490  0.0911
## BA   0.1803  0.1189 -0.1319
## C    0.2401  0.0195  0.1298
## CAT  0.2018  0.1562 -0.0906
## DD   0.1824  0.1939 -0.0644
## DIS  0.2148 -0.0276  0.0591
## GE   0.2334  0.0552  0.0040
## HD   0.2431  0.1386  0.2561
## HON  0.2623  0.1169 -0.1999
## HPQ  0.2787 -0.4285 -0.3094
## IBM  0.1965 -0.1924 -0.0303
## INTC 0.3225 -0.5415 -0.1012
## IP   0.1919  0.2000 -0.0013
## JNJ  0.0800  0.1212 -0.0222
## KO   0.0818  0.1354 -0.0309
## MCD  0.1109  0.1261 -0.0235
## MMM  0.1529  0.1492 -0.0340
## MO   0.0678  0.1521 -0.0419
## MRK  0.1057  0.1449  0.0019
## MSFT 0.2231 -0.2712 -0.0132
## PG   0.0772  0.1801 -0.0484
## T    0.1753 -0.1575  0.8195
## UTX  0.2104  0.1476 -0.1688
## WMT  0.1689  0.1228  0.1349
## XOM  0.1095  0.1115  0.0354
```

(b) Then use the estimated Principal Components loadings and apply them to Training Set 2 to create daily data for all the Principal Components for the dates in Training Set 2.

First, I check the missing values.

```
sum(is.na(train2wide))
```

```
## [1] 0
```

Then, I predict the daily return using the loading matrix from PCA with Training Set 1. The first 20 days in the Training Set 2 are shown as follows:

```
train2_data = train2wide[,2:length(train2wide[1,])]
train2_daily_predict = as.matrix(train2_data) %*% tra1loading
round(head(train2_daily_predict,20),4)
```

```
##        PC1     PC2     PC3
## 1   0.0575  0.0052  0.0056
## 2   0.0095 -0.0140 -0.0156
## 3   0.0025 -0.0133  0.0032
## 4   0.0385 -0.0043 -0.0001
## 5   0.0203  0.0065  0.0011
## 6  -0.0015  0.0001  0.0043
## 7   0.0150 -0.0140  0.0141
## 8  -0.0353 -0.0122  0.0026
## 9   0.0022 -0.0099 -0.0120
## 10 -0.0228  0.0057 -0.0051
## 11 -0.0454  0.0566  0.0138
## 12  0.0168  0.0023 -0.0058
## 13 -0.1039 -0.0102  0.0031
## 14 -0.0055  0.0241 -0.0062
## 15  0.0206  0.0073  0.0077
## 16  0.0024 -0.0142  0.0097
## 17  0.0513  0.0319 -0.0009
## 18  0.0354  0.0064 -0.0012
## 19 -0.0008 -0.0123  0.0103
## 20 -0.0048  0.0108 -0.0009
```

(b)Then create a data-frame where the Y variable is the first stock's return at time t + 1 and the X variables are all the lagged Principal Components from time t to time t □ - 30.

```
lth = length(train2_data[,1])
full = c()
k = 2
for (i in 31:(lth - 1))
  {
    df = c()
    dat = train2wide[i,1]
    value = train2wide[i + 1,k]
    firm = names(train2wide)[k]
    df = cbind(df, dat, value, firm)
    for (j in (i - 1):(i - 30))
    {
      df = cbind(df, train2_daily_predict[j,1], train2_daily_predict[j,2], train2_daily_predict[
j,3])
    }
    full = rbind(full,df)
  }
```

(c) Repeat this for all the stocks and stack these data-frames vertically (across stocks) to produce one such big data frame.

```
lth = length(train2_data[,1])
fultrain = c()
for (k in 2:length(names(train2wide)))
{
  full = c()
  for (i in 31:(lth - 1))
  {
    df = c()
    dat = train2wide[i,1]
    value = train2wide[i + 1,k]
    firm = names(train2wide)[k]
    df = cbind(df, dat, value, firm)
    for (j in (i - 1):(i - 30))
    {
      df = cbind(df, train2_daily_predict[j,1], train2_daily_predict[j,2], train2_daily_predict[
j,3])
    }
    full = rbind(full,df)
  }
  fultrain = rbind(fultrain, full)
}
```

Add dummy variables describing the different stocks.

```
fultrain = as.data.frame(fultrain)
fultrain[,1] = as.Date(as.numeric(fultrain[,1]), origin = "1970-01-01")
fultrain[,2] = as.double(as.character(fultrain[,2]))
fultrain[,3] = as.character(fultrain[,3])

for (a in 4:93)
{
  fultrain[,a] = as.double(as.character(fultrain[,a]))
}

library(dummies)
dful = dummy.data.frame(fultrain)
```

What is the dimensionality of your data-frame? Provide a printout of its ‘summary()’.

```
dim(dful)
```

```
## [1] 31928   118
```

```
summary(dful)
```

```
##       dat                 value                firmAA
##  Min.   :1970-01-02   Min.   :-0.3902440   Min.   :0.00000
##  1st Qu.:1970-11-04   1st Qu.:-0.0084735   1st Qu.:0.00000
##  Median :1971-09-07   Median : 0.0004735   Median :0.00000
##  Mean   :1971-09-07   Mean   : 0.0003789   Mean   :0.03846
##  3rd Qu.:1972-07-10   3rd Qu.: 0.0093142   3rd Qu.:0.00000
##  Max.   :1973-05-13   Max.   : 0.5782490   Max.   :1.00000
##     firmAXP             firmBA             firmC             firmCAT
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03846   Mean   :0.03846   Mean   :0.03846   Mean   :0.03846
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##      firmDD            firmDIS             firmGE             firmHD
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03846   Mean   :0.03846   Mean   :0.03846   Mean   :0.03846
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##     firmHON            firmHPQ            firmIBM            firmINTC
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03846   Mean   :0.03846   Mean   :0.03846   Mean   :0.03846
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##      firmIP            firmJNJ             firmKO            firmMCD
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03846   Mean   :0.03846   Mean   :0.03846   Mean   :0.03846
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##     firmMMM             firmMO            firmMRK            firmMSFT
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03846   Mean   :0.03846   Mean   :0.03846   Mean   :0.03846
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##      firmPG             firmT             firmUTX            firmWMT
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03846   Mean   :0.03846   Mean   :0.03846   Mean   :0.03846
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##     firmXOM               V4                   V5
##  Min.   :0.00000   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:0.00000   1st Qu.:-0.027646   1st Qu.:-0.0116842
##  Median :0.00000   Median : 0.004572   Median : 0.0009967
```

```
##  Mean   :0.03846   Mean   : 0.001766   Mean   : 0.0004290
##  3rd Qu.:0.00000   3rd Qu.: 0.033727   3rd Qu.: 0.0129009
##  Max.   :1.00000   Max.   : 0.577298   Max.   : 0.1192071
##        V6                   V7                   V8
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0116842
##  Median :-0.0004858   Median : 0.004572   Median : 0.0009967
##  Mean   :-0.0001685   Mean   : 0.001757   Mean   : 0.0004327
##  3rd Qu.: 0.0076489   3rd Qu.: 0.033727   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##        V9                  V10                  V11
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0116842
##  Median :-0.0004858   Median : 0.004572   Median : 0.0009967
##  Mean   :-0.0001657   Mean   : 0.001769   Mean   : 0.0004327
##  3rd Qu.: 0.0076489   3rd Qu.: 0.033727   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V12                  V13                  V14
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0116842
##  Median :-0.0004615   Median : 0.004649   Median : 0.0010116
##  Mean   :-0.0001553   Mean   : 0.001777   Mean   : 0.0004421
##  3rd Qu.: 0.0076489   3rd Qu.: 0.033727   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V15                  V16                  V17
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0116842
##  Median :-0.0004858   Median : 0.004683   Median : 0.0009967
##  Mean   :-0.0001625   Mean   : 0.001815   Mean   : 0.0004366
##  3rd Qu.: 0.0076145   3rd Qu.: 0.033905   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V18                  V19                  V20
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0117239
##  Median :-0.0004858   Median : 0.004649   Median : 0.0009678
##  Mean   :-0.0001668   Mean   : 0.001797   Mean   : 0.0004055
##  3rd Qu.: 0.0076145   3rd Qu.: 0.033905   3rd Qu.: 0.0128873
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V21                  V22                  V23
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0117239
##  Median :-0.0005018   Median : 0.004572   Median : 0.0009678
##  Mean   :-0.0001719   Mean   : 0.001779   Mean   : 0.0004116
##  3rd Qu.: 0.0076145   3rd Qu.: 0.033905   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V24                  V25                  V26
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027646   1st Qu.:-0.0117239
##  Median :-0.0004858   Median : 0.004572   Median : 0.0009678
##  Mean   :-0.0001626   Mean   : 0.001770   Mean   : 0.0004097
##  3rd Qu.: 0.0076145   3rd Qu.: 0.033905   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V27                  V28                  V29
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
```

```
##  1st Qu.:-0.0086415   1st Qu.:-0.027756   1st Qu.:-0.0117239
##  Median :-0.0004615   Median : 0.004399   Median : 0.0008854
##  Mean   :-0.0001523   Mean   : 0.001727   Mean   : 0.0004044
##  3rd Qu.: 0.0076489   3rd Qu.: 0.033905   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V30                  V31                 V32
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027756   1st Qu.:-0.0117239
##  Median :-0.0004615   Median : 0.004399   Median : 0.0008854
##  Mean   :-0.0001441   Mean   : 0.001733   Mean   : 0.0004047
##  3rd Qu.: 0.0076768   3rd Qu.: 0.034057   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V33                  V34                 V35
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0117239
##  Median :-0.0004246   Median : 0.004399   Median : 0.0009678
##  Mean   :-0.0001334   Mean   : 0.001752   Mean   : 0.0004145
##  3rd Qu.: 0.0076768   3rd Qu.: 0.034057   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V36                  V37                 V38
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0117637
##  Median :-0.0004246   Median : 0.004193   Median : 0.0008854
##  Mean   :-0.0001320   Mean   : 0.001741   Mean   : 0.0004037
##  3rd Qu.: 0.0076768   3rd Qu.: 0.034057   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V39                  V40                 V41
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0117637
##  Median :-0.0004246   Median : 0.004399   Median : 0.0008854
##  Mean   :-0.0001306   Mean   : 0.001773   Mean   : 0.0003876
##  3rd Qu.: 0.0076768   3rd Qu.: 0.034227   3rd Qu.: 0.0128873
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V42                  V43                 V44
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0117637
##  Median :-0.0004615   Median : 0.004399   Median : 0.0008854
##  Mean   :-0.0001378   Mean   : 0.001789   Mean   : 0.0004085
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034304   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V45                  V46                 V47
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0118465
##  Median :-0.0004858   Median : 0.004399   Median : 0.0007977
##  Mean   :-0.0001382   Mean   : 0.001790   Mean   : 0.0003947
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034304   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V48                  V49                 V50
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0117637
##  Median :-0.0004858   Median : 0.004399   Median : 0.0008854
##  Mean   :-0.0001405   Mean   : 0.001798   Mean   : 0.0004140
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034304   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
```

```
##       V51                 V52                V53          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086002   1st Qu.:-0.027646   1st Qu.:-0.0117637  
##  Median :-0.0004858   Median : 0.004193   Median : 0.0008854  
##  Mean   :-0.0001396   Mean   : 0.001785   Mean   : 0.0004218  
##  3rd Qu.: 0.0077147   3rd Qu.: 0.034304   3rd Qu.: 0.0129009  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V54                 V55                V56          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086002   1st Qu.:-0.027757   1st Qu.:-0.0117637  
##  Median :-0.0005018   Median : 0.004193   Median : 0.0007977  
##  Mean   :-0.0001601   Mean   : 0.001708   Mean   : 0.0004126  
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034304   3rd Qu.: 0.0129009  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V57                 V58                V59          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086002   1st Qu.:-0.027757   1st Qu.:-0.0117637  
##  Median :-0.0004858   Median : 0.004193   Median : 0.0007977  
##  Mean   :-0.0001554   Mean   : 0.001710   Mean   : 0.0004069  
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034304   3rd Qu.: 0.0129009  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V60                 V61                V62          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086002   1st Qu.:-0.028165   1st Qu.:-0.0117637  
##  Median :-0.0004858   Median : 0.004038   Median : 0.0007977  
##  Mean   :-0.0001579   Mean   : 0.001615   Mean   : 0.0004485  
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034227   3rd Qu.: 0.0129356  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V63                 V64                V65          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086002   1st Qu.:-0.028165   1st Qu.:-0.0117637  
##  Median :-0.0004858   Median : 0.003966   Median : 0.0007977  
##  Mean   :-0.0001588   Mean   : 0.001501   Mean   : 0.0004254  
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034057   3rd Qu.: 0.0129009  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V66                 V67                V68          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086002   1st Qu.:-0.028165   1st Qu.:-0.0117637  
##  Median :-0.0005018   Median : 0.003966   Median : 0.0007977  
##  Mean   :-0.0001651   Mean   : 0.001517   Mean   : 0.0004169  
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034057   3rd Qu.: 0.0129009  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V69                 V70                V71          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086415   1st Qu.:-0.028824   1st Qu.:-0.0118465  
##  Median :-0.0005108   Median : 0.003966   Median : 0.0007618  
##  Mean   :-0.0001840   Mean   : 0.001492   Mean   : 0.0003990  
##  3rd Qu.: 0.0076145   3rd Qu.: 0.034057   3rd Qu.: 0.0129009  
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071  
##       V72                 V73                V74          
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194  
##  1st Qu.:-0.0086415   1st Qu.:-0.028165   1st Qu.:-0.0118465  
##  Median :-0.0005018   Median : 0.004038   Median : 0.0007618  
##  Mean   :-0.0001812   Mean   : 0.001535   Mean   : 0.0003975
```

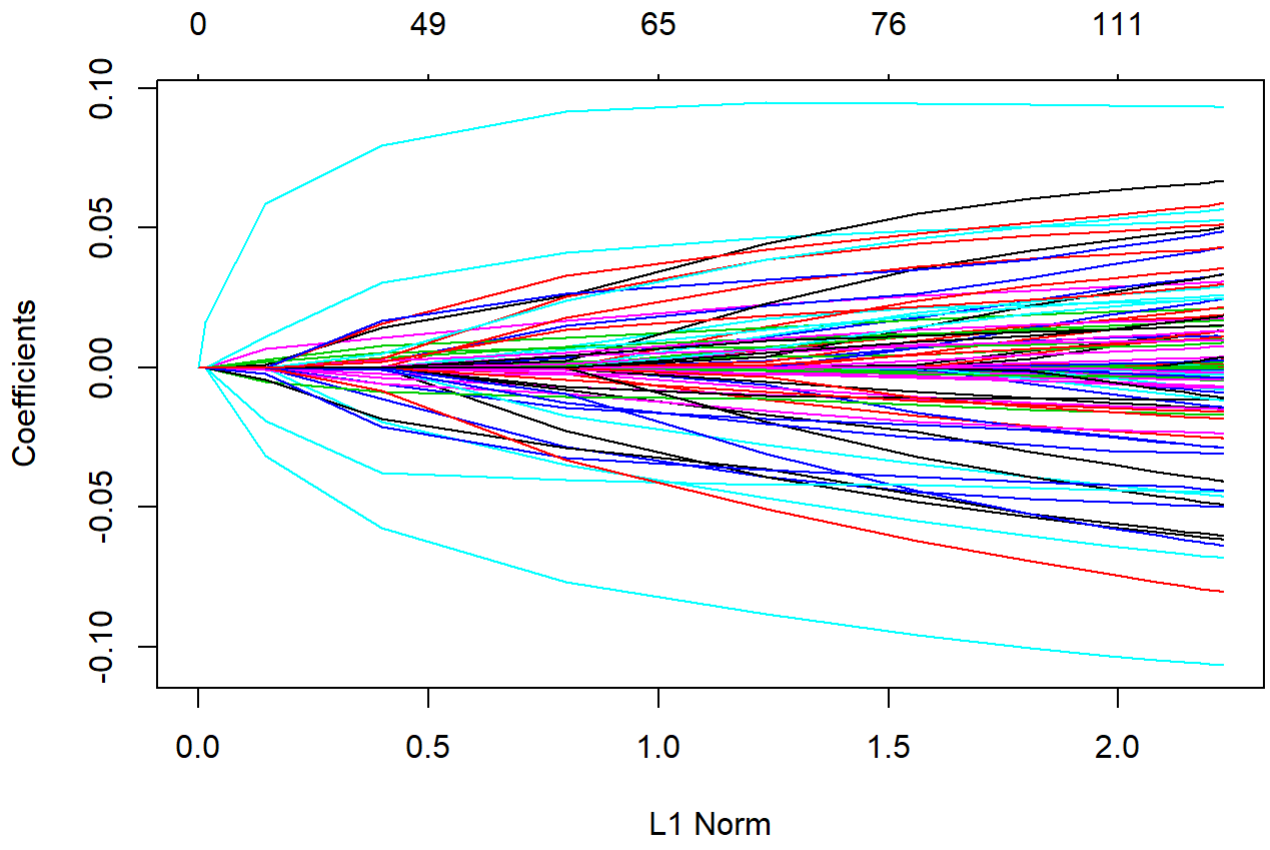
```
##  3rd Qu.: 0.0076145   3rd Qu.: 0.034057   3rd Qu.: 0.0129009
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V75                 V76                 V77
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.028165   1st Qu.:-0.0118465
##  Median :-0.0005018   Median : 0.003966   Median : 0.0007438
##  Mean   :-0.0001712   Mean   : 0.001477   Mean   : 0.0003734
##  3rd Qu.: 0.0076590   3rd Qu.: 0.033905   3rd Qu.: 0.0128873
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V78                 V79                 V80
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086415   1st Qu.:-0.027757   1st Qu.:-0.0117637
##  Median :-0.0004858   Median : 0.004038   Median : 0.0007618
##  Mean   :-0.0001647   Mean   : 0.001537   Mean   : 0.0004014
##  3rd Qu.: 0.0076590   3rd Qu.: 0.033905   3rd Qu.: 0.0128873
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V81                 V82                 V83
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027757   1st Qu.:-0.0117239
##  Median :-0.0004337   Median : 0.004193   Median : 0.0007618
##  Mean   :-0.0001533   Mean   : 0.001572   Mean   : 0.0004097
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034057   3rd Qu.: 0.0128873
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V84                 V85                 V86
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0085663   1st Qu.:-0.027757   1st Qu.:-0.0117637
##  Median :-0.0003962   Median : 0.004038   Median : 0.0007618
##  Mean   :-0.0001449   Mean   : 0.001563   Mean   : 0.0004005
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034057   3rd Qu.: 0.0128873
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V87                 V88                 V89
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0085663   1st Qu.:-0.027757   1st Qu.:-0.0118465
##  Median :-0.0003914   Median : 0.004038   Median : 0.0007438
##  Mean   :-0.0001363   Mean   : 0.001508   Mean   : 0.0003634
##  3rd Qu.: 0.0076590   3rd Qu.: 0.033905   3rd Qu.: 0.0127038
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V90                 V91                 V92
##  Min.   :-0.0736256   Min.   :-0.430921   Min.   :-0.1444194
##  1st Qu.:-0.0086002   1st Qu.:-0.027757   1st Qu.:-0.0118465
##  Median :-0.0003914   Median : 0.004193   Median : 0.0007438
##  Mean   :-0.0001461   Mean   : 0.001568   Mean   : 0.0003586
##  3rd Qu.: 0.0076590   3rd Qu.: 0.034057   3rd Qu.: 0.0127038
##  Max.   : 0.1267421   Max.   : 0.577298   Max.   : 0.1192071
##       V93
##  Min.   :-0.0736256
##  1st Qu.:-0.0085663
##  Median :-0.0003793
##  Mean   :-0.0001295
##  3rd Qu.: 0.0076590
##  Max.   : 0.1267421
```

Thus, there are 31928 Columns and 118 rows.

Task 3

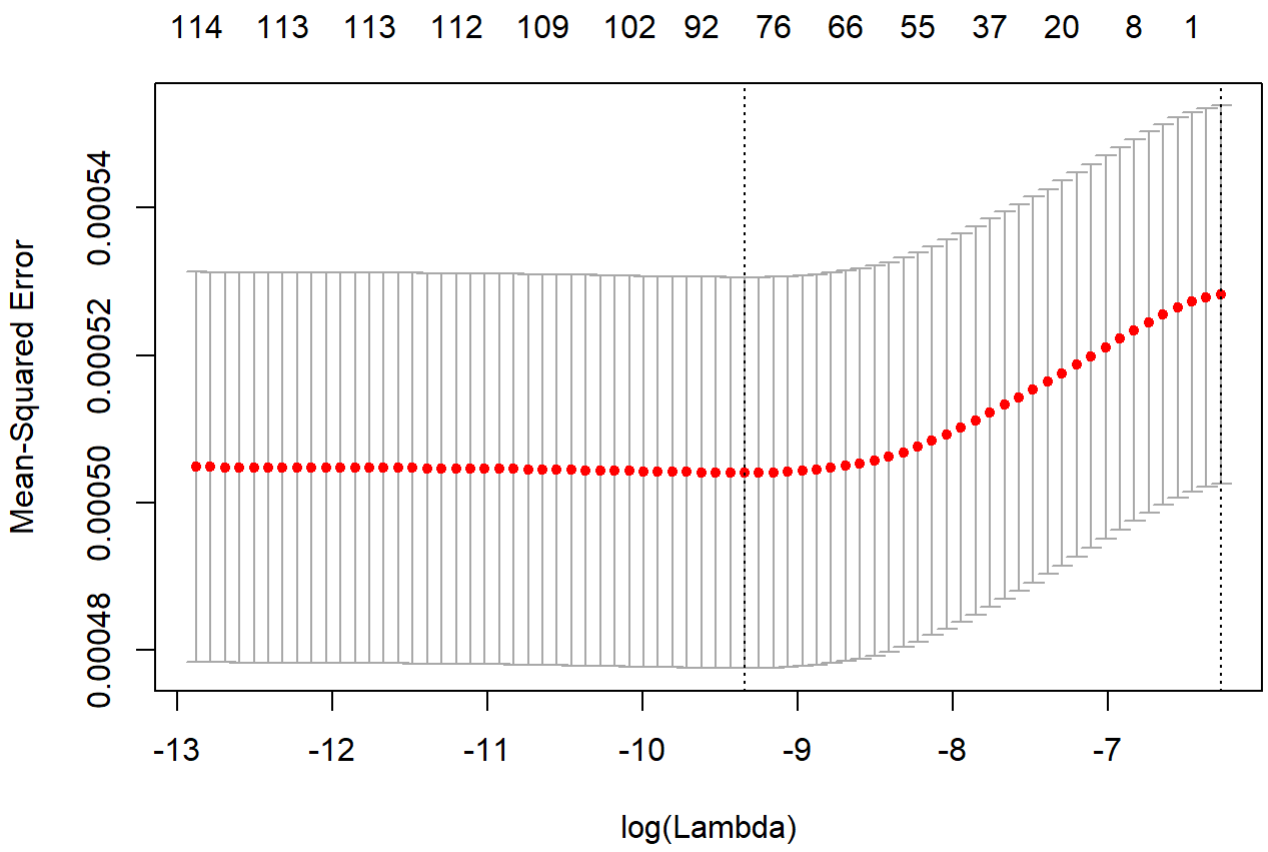
(a)Fit a Lasso model to predict the t + 1 return using the Principal Components from t to t - 30 as explanatory variables. In your data-frame above, for each each row the “Y” should be the return of a stock at t+1 and the “X”s should be all the principal components from t to t - □30 plus the stock dummy variable.

```
x <- model.matrix(value~.,dful)[,-c(1,2)]
y <- dful$value
library(ISLR)
library(glmnet)
grd <- 0.01^seq( 10, -2, length = 100)
set.seed(1)
train <- sample( 1:nrow(x),nrow(x)/2)
test <- -train
y.test <- y[test]

lasso.mod <- glmnet( x[train, ], y[train], alpha = 1, lambda = grd)
plot(lasso.mod)
```



(b) Use 5-fold cross validation to do feature selection. Create a plot of the Lasso parameter vs. the MSE.

```
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1, nfolds = 5)
plot(cv.out)
```



Report your optimal Lasso parameter. Fit the model using the optimal Lasso lambda parameter calculated above to the whole training data and report your results.

```
bestlam = cv.out$lambda.min
sprintf("bestlam is %.10f", bestlam)
```

```
## [1] "bestlam is 0.0000875170"
```

```
trlasso.pred <- predict(cv.out, s = bestlam, newx <- x)
sprintf("MSE is %.10f", mean((trlasso.pred - y)^2))
```

```
## [1] "MSE is 0.0004982277"
```

(c)Are there any issues with using cross validation in a time series setting?

Answer: In a time series setting, it is possible that there are significant differences between observations at different periods of time. For example, due to changes in market regulations, the outside environment and company structure during certain periods of Training Set 2 were hugely different from other periods. It would be biased to use cross validation in this setting.

Task 4

(a) Use the fitted model to predict returns in the Test Set.

First, I checked the missing values in Test Set and replace them with 0.

```
testwide[is.na(testwide)] = 0
```

Then, following the same procedure as Task 2, I calculated the principal components in the Test Set.

```
test_data = testwide[,2:length(testwide[1,])]
tr1pca2 = prcomp(train2_data, scale = FALSE)
tr1rot2 = tr1pca2$rotation
tra1loading = tr1rot2[,1:3]
test_daily_predict = as.matrix(test_data) %*% tra1loading
```

Then I construct the stacked dataset similar to Task 2 including the three principal components within the 30-day window (without dummies for firms).

```
lth = length(test_data[,1])
ful = c()
for (k in 2:length(names(testwide)))
{
  full = c()
  for (i in 31:(lth - 1))
  {
    df = c()
    dat = testwide[i,1]
    value = testwide[i + 1,k]
    firm = names(testwide)[k]
    df = cbind(df, dat, value, firm)
    for (j in (i - 1):(i - 30))
    {
      df = cbind(df, test_daily_predict[j,1], test_daily_predict[j,2], test_daily_predict[j,3])
    }
    full = rbind(full,df)
  }
  ful = rbind(ful, full)
}

fultest = as.data.frame(ful)
fultest[,1] = as.character(fultest[,1])
fultest[,1] = as.numeric(fultest[,1])
fultest[,1] = as.Date(fultest[,1], origin = "1970-01-01")
fultest[,2] = as.double(as.character(fultest[,2]))
fultest[,3] = as.character(fultest[,3])

for (a in 4:93)
{
  fultest[,a] = as.double(as.character(fultest[,a]))
}
```

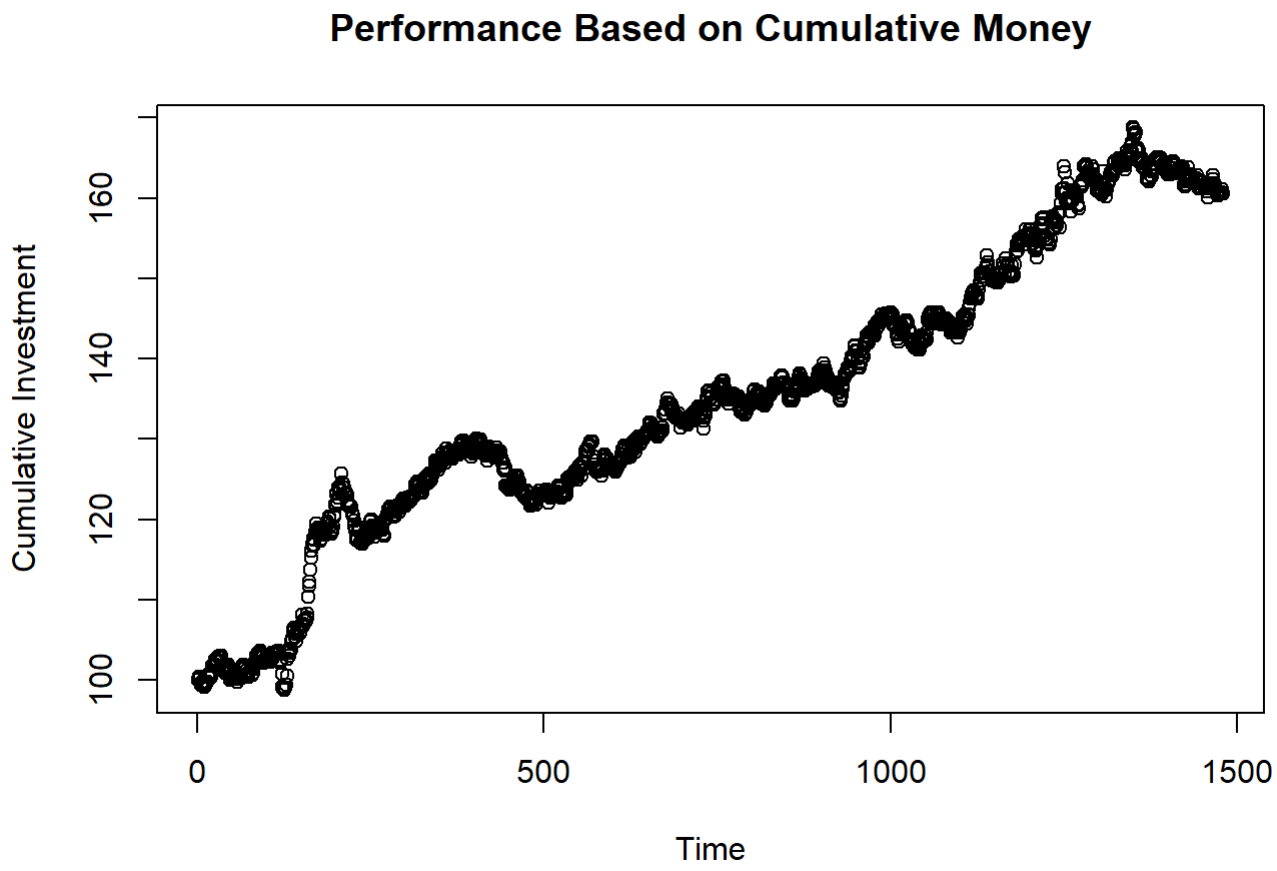
In the Training Set 2, for each company, there are 1259 observations. In the Test Set, for each company, there are 1510 observations. Construct the Long/Short Portfolio for every day in the Test Set

```
xtr <- fultrain[,-c(1,2,3)]
ytr <- fultrain$value
xte <- fultest[,-c(1,2,3)]
yte <- fultest$value
yte = matrix(yte, 1479, 26)

return_predict = c()
for (m in 0:25)
{
  lasso.mod <- glmnet( as.matrix(xtr)[(1 + 1228 * m) : (1228 * (m + 1)), ], ytr[(1 + 1228 * m):(
1228 * (m + 1))], alpha = 1, lambda = bestlam)
  pre = predict(lasso.mod, s = bestlam, newx <- as.matrix(xte)[(1 + 1479 * m):(1479 * (m + 1)),
])
  return_predict = cbind(return_predict, pre)
}

lr = c()
sr = c()

for (j in 1:1479)
{
  indexl = order(return_predict[j,],decreasing = TRUE)[1:5]
  indexs = order(return_predict[j,],decreasing = FALSE)[1:5]
  long_return = yte[j,indexl]
  lr = rbind(lr, long_return)
  short_return = yte[j,indexs]
  sr = rbind(sr, short_return)
}
head(cbind(lr,sr))
```

```
##                 [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## long_return  0.000000 -0.001707  0.003323 -0.007279  0.003439 -0.002037
## long_return  0.008163 -0.003980  0.007032  0.002270 -0.003311  0.003731
## long_return -0.006073 -0.013699  0.002480  0.024187 -0.003717 -0.007553
## long_return -0.044806 -0.014905 -0.009040 -0.010135 -0.010327 -0.052378
## long_return -0.010493 -0.021614 -0.006737 -0.009869  0.002353 -0.019508
## long_return  0.000266  0.003683 -0.001870 -0.032576  0.007547 -0.002128
##                 [,7]      [,8]      [,9]     [,10]
## long_return  0.014505  0.010920 -0.001663  0.005340
## long_return  0.004248  0.008452  0.002797  0.003666
## long_return  0.001974  0.002426  0.002456  0.000000
## long_return -0.017369 -0.036369 -0.042824 -0.028582
## long_return  0.000000 -0.006046 -0.019192 -0.009455
## long_return  0.002762  0.006769 -0.002548  0.006691
```

Evaluate the performance of this strategy.

```
rot = 0
rot[] = NA
rot[1] = 100
for (mj in 1:1479){
  rot[mj + 1] = rot[mj] + sum(rot[mj]/10 * ( -lr[mj,] + sr[mj,]))
}
plot(rot,main = "Performance Based on Cumulative Money", ylab = "Cumulative Investment", xlab =
"Time")
```




```
sprintf("The average annual return rate is %.4f", (rot[1480]/100)^(1/6)-1)
```

```
## [1] "The average annual return rate is 0.0821"
```