Homework 1

Aryan Mishra

2/17/2022

Loading the Relevant Libraries.

```
library(plyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-3

library(mgcv)

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.8-38. For overview type 'help("mgcv-package")'.

library(Metrics)
library(fastDummies)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

Setting the Seed and Working Directory.

```
set.seed(1)
setwd('/Users/aryan/Desktop/MSBA_Spring_22/Machine Learning /Homework_1')
```

Loading the Dataset.

```
data("dji30ret", package = "rugarch")
head(dji30ret)

##                      AA          AXP           BA          BAC
C
## 1987-03-16 -0.024466052 -0.006329135 -0.018158735 -0.007142888  0.00000000
0
## 1987-03-17  0.024466052  0.007905180  0.003919012  0.007142888  0.00000000
0
## 1987-03-18  0.000000000  0.012519725  0.005201572  0.017637141  0.03524593
9
## 1987-03-19  0.020927520  0.001554002 -0.005201572  0.006968669  0.02564243
1
## 1987-03-20 -0.002962965 -0.018809332  0.033336420  0.017212129 -0.00847462
7
## 1987-03-23  0.002962965 -0.004758137 -0.006325132 -0.003418807  0.01268516
0
##                     CAT        CVX          DD          DIS          GE
## 1987-03-16 -0.010810916 0.02030527  0.00000000 -0.021713760 -0.00806456
## 1987-03-17  0.002713706 0.03295191  0.03913300  0.035932009  0.02004075
## 1987-03-18 -0.013642776 0.02085080  0.02169282  0.000000000 -0.01197619
## 1987-03-19  0.019048195 0.00000000 -0.01950224 -0.009456335  0.00000000
## 1987-03-20  0.010723963 0.01886848  0.02378491  0.009456335  0.01988137
## 1987-03-23  0.015873349 0.01698883  0.02427560  0.020955365  0.02718614
##                      GM          HD         HPQ          IBM        INTC
## 1987-03-16 -0.011825128 -0.02105341 -0.01129956 -0.001693481  0.03077166
## 1987-03-17  0.008685408  0.00000000  0.04445176  0.016807118  0.05884050
## 1987-03-18 -0.003938564  0.08167803 -0.01754431 -0.005850414  0.01418463
## 1987-03-19  0.005509656  0.01941809  0.03264708  0.005016733  0.04138522
## 1987-03-20  0.004698521  0.03774033  0.01698555  0.011196468 -0.01360565
## 1987-03-23  0.013964540  0.03636764 -0.01058211  0.007395268  0.00000000
##                     JNJ          JPM         AIG           KO          MCD
## 1987-03-16 -0.010810916 -0.012205906 0.000000000 -0.020408872 -0.012396853
## 1987-03-17  0.018843088  0.012205906 0.005352377  0.007702220  0.018538121
## 1987-03-18 -0.008032172 -0.003472226 0.000000000 -0.002560821 -0.018538121
## 1987-03-19  0.000000000 -0.001740644 0.000000000  0.020305266  0.002076844
## 1987-03-20  0.013351333 -0.003490405 0.058738817  0.000000000  0.018500014
## 1987-03-23  0.018397365  0.010434877 0.018288124  0.012484557  0.032066876
##                     MMM          MRK        MSFT          PFE           PG
## 1987-03-16 -0.006079046  0.013216051  0.00000000 -0.005633818 -0.022663860
## 1987-03-17  0.021713760  0.034411998  0.04082199  0.005633818  0.011396135
## 1987-03-18 -0.003586376  0.000000000  0.03922071  0.011173301  0.011267725
## 1987-03-19  0.020154802 -0.006362694  0.03774033  0.005540180  0.011142177
## 1987-03-20  0.026637360  0.025211419 -0.03774033  0.005509656  0.002766253
## 1987-03-23  0.011363759 -0.002076844  0.00000000  0.005479466  0.010989122
##                       T          UTX           VZ          WMT           XO
M
## 1987-03-16 -0.002528446 -0.002509412 -0.004590674 -0.019868203 -0.00597016
7
## 1987-03-17  0.000000000  0.010000083  0.001532567  0.009983444  0.02756080
0
## 1987-03-18 -0.002534856 -0.007490672 -0.013878403 -0.006644543  0.01158314
1
## 1987-03-19  0.005063302  0.002503130  0.006191970  0.003327790 -0.00577479
9
## 1987-03-20  0.017522351  0.000000000  0.006153866  0.022989518  0.00960622
2
## 1987-03-23  0.002478316 -0.005012542  0.009160369  0.022472856  0.02827709
6
```

Question 1

```
print(dim(dji30ret))

## [1] 5521   30

print(summary(dji30ret))

##        AA                 AXP                  BA          
##  Min.   :-0.2745595   Min.   :-0.3034304   Min.   :-0.1938568  
##  1st Qu.:-0.0114593   1st Qu.:-0.0109291   1st Qu.:-0.0098007  
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000  
##  Mean   : 0.0001608   Mean   : 0.0001687   Mean   : 0.0003058  
##  3rd Qu.: 0.0116377   3rd Qu.: 0.0114812   3rd Qu.: 0.0105709  
##  Max.   : 0.2087337   Max.   : 0.1712035   Max.   : 0.1439727  
##       BAC                  C                   CAT          
##  Min.   :-0.3420588   Min.   :-0.3056056   Min.   :-0.244156  
##  1st Qu.:-0.0093365   1st Qu.:-0.0111602   1st Qu.:-0.010575  
##  Median : 0.0000000   Median : 0.0000000   Median : 0.000000  
##  Mean   : 0.0001149   Mean   : 0.0000796   Mean   : 0.000378  
##  3rd Qu.: 0.0100293   3rd Qu.: 0.0116803   3rd Qu.: 0.011141  
##  Max.   : 0.2698774   Max.   : 0.4572902   Max.   : 0.137371  
##       CVX                 DD                  DIS          
##  Min.   :-0.1812526   Min.   :-0.2018984   Min.   :-0.3426451  
##  1st Qu.:-0.0082829   1st Qu.:-0.0093255   1st Qu.:-0.0102932  
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000  
##  Mean   : 0.0004538   Mean   : 0.0001774   Mean   : 0.0002886  
##  3rd Qu.: 0.0095239   3rd Qu.: 0.0095836   3rd Qu.: 0.0105821  
##  Max.   : 0.1894765   Max.   : 0.1086964   Max.   : 0.1756133  
##        GE                   GM                   HD
##  Min.   :-0.1947441   Min.   :-0.3727220   Min.   :-0.3386365
##  1st Qu.:-0.0083683   1st Qu.:-0.0119502   1st Qu.:-0.0115889
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0002751   Mean   :-0.0002715   Mean   : 0.0006922
##  3rd Qu.: 0.0093459   3rd Qu.: 0.0115692   3rd Qu.: 0.0127656
##  Max.   : 0.1275967   Max.   : 0.3009365   Max.   : 0.1315251
##       HPQ                  IBM                 INTC
##  Min.   :-0.2263815   Min.   :-0.268161   Min.   :-0.2488610
##  1st Qu.:-0.0125577   1st Qu.:-0.009243   1st Qu.:-0.0139915
##  Median : 0.0000000   Median : 0.000000   Median : 0.0000000
##  Mean   : 0.0003792   Mean   : 0.000249   Mean   : 0.0005553
##  3rd Qu.: 0.0135366   3rd Qu.: 0.009469   3rd Qu.: 0.0158734
##  Max.   : 0.1591410   Max.   : 0.123635   Max.   : 0.2265276
##       JNJ                  JPM                  AIG
##  Min.   :-0.2043813   Min.   :-0.3234769   Min.   :-0.9362581
##  1st Qu.:-0.0078036   1st Qu.:-0.0111299   1st Qu.:-0.0089217
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0004993   Mean   : 0.0002586   Mean   :-0.0002978
##  3rd Qu.: 0.0084695   3rd Qu.: 0.0111094   3rd Qu.: 0.0094814
##  Max.   : 0.1153126   Max.   : 0.2239172   Max.   : 0.3585320
##        KO                  MCD                  MMM
##  Min.   :-0.2828628   Min.   :-0.1827990   Min.   :-0.2257926
##  1st Qu.:-0.0080020   1st Qu.:-0.0093024   1st Qu.:-0.0074074
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0004333   Mean   : 0.0004514   Mean   : 0.0003322
##  3rd Qu.: 0.0089027   3rd Qu.: 0.0099834   3rd Qu.: 0.0082499
##  Max.   : 0.1791113   Max.   : 0.1030806   Max.   : 0.1049975
##       MRK                  MSFT                PFE
##  Min.   :-0.3119154   Min.   :-0.379490   Min.   :-0.1892420
##  1st Qu.:-0.0090772   1st Qu.:-0.010643   1st Qu.:-0.0096386
##  Median : 0.0000000   Median : 0.000000   Median : 0.0000000
##  Mean   : 0.0003447   Mean   : 0.000787   Mean   : 0.0003885
##  3rd Qu.: 0.0100307   3rd Qu.: 0.012848   3rd Qu.: 0.0107528
##  Max.   : 0.1224923   Max.   : 0.178465   Max.   : 0.0989399
##        PG                   T                   UTX
##  Min.   :-0.3598917   Min.   :-0.1352280   Min.   :-0.3029024
##  1st Qu.:-0.0074074   1st Qu.:-0.0087377   1st Qu.:-0.0083770
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0004917   Mean   : 0.0003364   Mean   : 0.0004517
##  3rd Qu.: 0.0085107   3rd Qu.: 0.0096619   3rd Qu.: 0.0098064
##  Max.   : 0.1980376   Max.   : 0.1505242   Max.   : 0.1278841
##        VZ                  WMT                  XOM
##  Min.   :-0.1931448   Min.   :-0.1249721   Min.   :-0.2676996
##  1st Qu.:-0.0087802   1st Qu.:-0.0100137   1st Qu.:-0.0078818
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0002848   Mean   : 0.0004985   Mean   : 0.0004964
##  3rd Qu.: 0.0089366   3rd Qu.: 0.0104713   3rd Qu.: 0.0090419
##  Max.   : 0.1365130   Max.   : 0.1146918   Max.   : 0.1653925
```

Question 2A

```
aa <- as.data.frame(dji30ret[['AA']], row.names = row.names(dji30ret))
colnames(aa)[1] <- "Target"
head(aa)

##                   Target
## 1987-03-16 -0.024466052
## 1987-03-17  0.024466052
## 1987-03-18  0.000000000
## 1987-03-19  0.020927520
## 1987-03-20 -0.002962965
## 1987-03-23  0.002962965
```

Question 2B

```
aa <- aa %>% bind_cols(data.frame(t(ldply(1:5, lag, x= (dji30ret$AA))))) #Add
ing lagged returns of AA using some R magic
aa <- na.omit(aa) #Removing NAs
head(aa)

##                   Target           X1           X2           X3            X
4
## 1987-03-23  0.002962965 -0.002962965  0.020927520  0.000000000  0.02446605
2
## 1987-03-24  0.017595762  0.002962965 -0.002962965  0.020927520  0.00000000
0
## 1987-03-25  0.005797118  0.017595762  0.002962965 -0.002962965  0.02092752
0
## 1987-03-26 -0.032308243  0.005797118  0.017595762  0.002962965 -0.00296296
5
## 1987-03-27  0.000000000 -0.032308243  0.005797118  0.017595762  0.00296296
5
## 1987-03-30 -0.024170361  0.000000000 -0.032308243  0.005797118  0.01759576
2
##                       X5
## 1987-03-23 -0.024466052
## 1987-03-24  0.024466052
## 1987-03-25  0.000000000
## 1987-03-26  0.020927520
## 1987-03-27 -0.002962965
## 1987-03-30  0.002962965
```

Question 2B (continued)

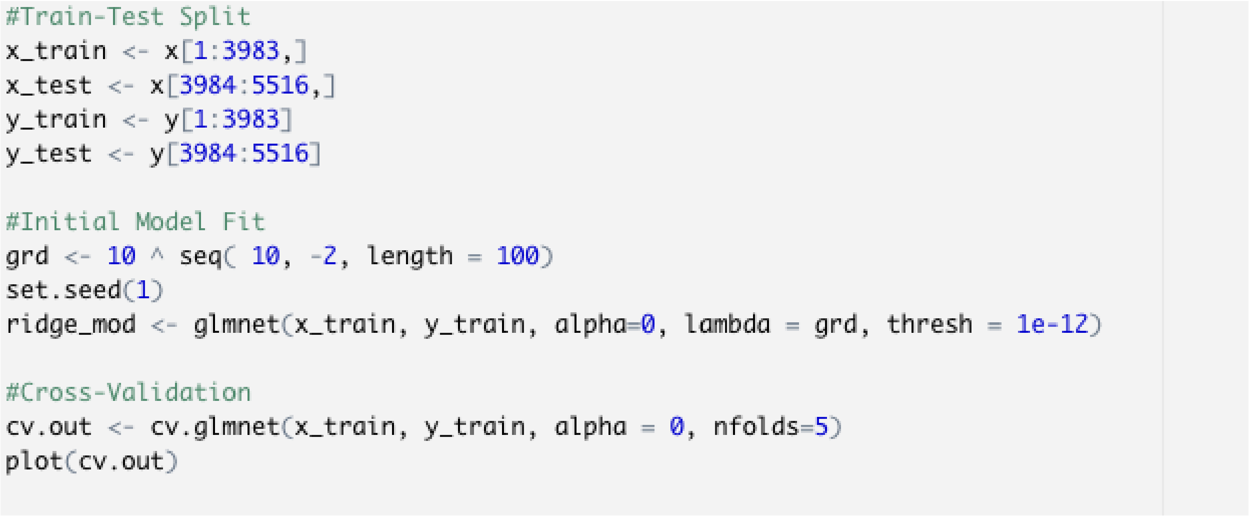
```
print(summary(aa))

##       Target                 X1                  X2
##   Min.   :-0.2745595   Min.   :-0.274560   Min.   :-0.2745595
##   1st Qu.:-0.0114625   1st Qu.:-0.011463   1st Qu.:-0.0114461
##   Median : 0.0000000   Median : 0.000000   Median : 0.0000000
##   Mean   : 0.0001576   Mean   : 0.000149   Mean   : 0.0001552
##   3rd Qu.: 0.0116305   3rd Qu.: 0.011618   3rd Qu.: 0.0116305
##  Max.   : 0.2087337   Max.   : 0.208734   Max.   : 0.2087337
##       X3                   X4                   X5
##  Min.   :-0.2745595   Min.   :-0.2745595   Min.   :-0.2745595
##  1st Qu.:-0.0114343   1st Qu.:-0.0114295   1st Qu.:-0.0114343
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0001697   Mean   : 0.0001807   Mean   : 0.0001733
##  3rd Qu.: 0.0116305   3rd Qu.: 0.0116378   3rd Qu.: 0.0116305
##  Max.   : 0.2087337   Max.   : 0.2087337   Max.   : 0.2087337
```

Question 2C

```
x <- model.matrix(Target~.,aa)[,-1]
y <- aa$Target
print(which(rownames(aa) %in% c("1987-03-23", "2002-12-31"))) #Finding out th
e indices for the relevant dates.

## [1]    1 3983

print(which(rownames(aa) %in% c("2003-01-01", "2009-02-03")))

## [1] 5516
```

As we can see, the range for the training data is from index 1 to index 3983, while the range for the testing data is from index 3984 to index 5516.

Question 2C (continued)

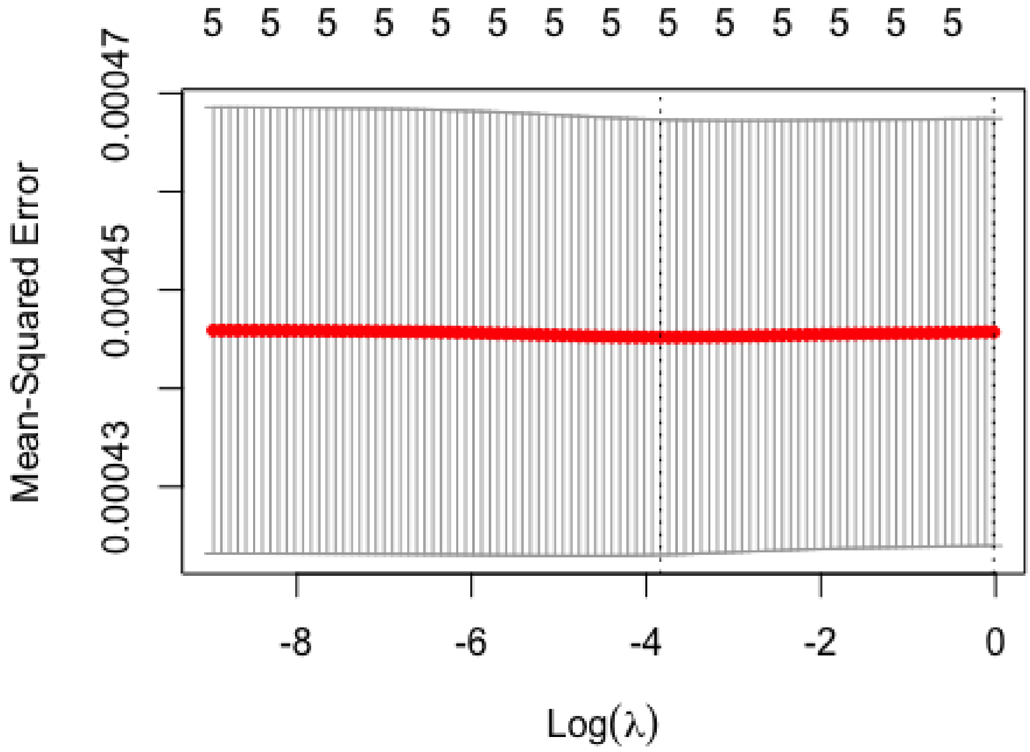


```
#Train-Test Split
x_train <- x[1:3983,]
x_test <- x[3984:5516,]
y_train <- y[1:3983]
y_test <- y[3984:5516]

#Initial Model Fit
grd <- 10 ^ seq( 10, -2, length = 100)
set.seed(1)
ridge_mod <- glmnet(x_train, y_train, alpha=0, lambda = grd, thresh = 1e-12)

#Cross-Validation
cv.out <- cv.glmnet(x_train, y_train, alpha = 0, nfolds=5)
plot(cv.out)
```

[1]

[1] For some reason, R didn't compile this block of code, so I had to take a screenshot.

Question 2C (continued)



Question 2C (continued)

```
bestlam <- cv.out$lambda.min
print(bestlam)

## [1] 0.02156627
```

The optimal lambda, in this case, is 0.02156627. Therefore, we will use this value when fitting our model.

Question 2C (continued)

```
#Fitting Model using Optimal Lambda
set.seed(1)
ridge_mod <- glmnet(x_train, y_train, alpha=0, lambda = bestlam)
print(summary(ridge_mod))

##           Length Class     Mode
## a0        1      -none-    numeric
## beta      5      dgCMatrix S4
## df        1      -none-    numeric
## dim       2      -none-    numeric
## lambda    1      -none-    numeric
## dev.ratio 1       -none-     numeric
## nulldev   1       -none-     numeric
## npasses   1       -none-     numeric
## jerr      1       -none-     numeric
## offset    1       -none-     logical
## call      5       -none-     call
## nobs      1       -none-     numeric
```

Question 2D

```
#Predicting Model
set.seed(1)
ridge_pred <- predict(ridge_mod, s = 0, newx = x_test)
print(mae(y_test, ridge_pred))

## [1] 0.01772329
```

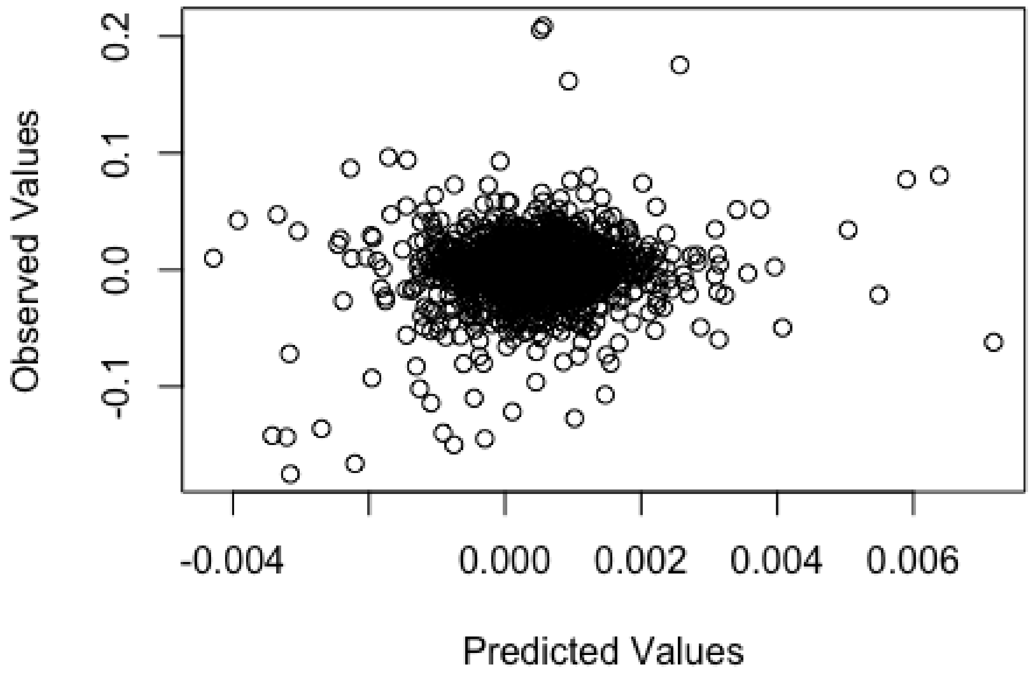
As seen above, the MAE is 0.01772329.

Question 2D (continued)

```
plot(ridge_pred, y_test,xlab = "Predicted Values", ylab = "Observed Values")
```

[2]

[2] For some reason, R didn't compile this block of code, so I had to take a screenshot.



Conclusions: Even though it might seem like we have a very low mean absolute error, in the context of predicting daily stock returns, the model performs poorly. Essentially, an MAE of 0.0177 implies that our average ABSOLUTE error is 1.77%, and in a data set with a mean daily return of 0.0001576 (0.01576%), the error is quite substantial. In the world of finance, this could lead to huge losses, especially also in the presence of outliers (which are indeed present in our data set). This model is basically an autoregressive model, specifically an AR (5) model, which is rarely utilized in technical analysis to forecast future security prices. This is because we are implicitly assuming that the future will resemble the past, which can prove inaccurate under certain market conditions, such as financial crises, or the surge of 'meme-stocks'.

Furthermore, note that since the number of predictors is a lot lower than the number of observations, and that the relationship between the past 5 lagged returns and today's returns are almost always non-linear, a ridge regression will not perform well, which might explain why our model is performing poorly.

Question 3A

```
df_main <- data.frame(matrix(ncol = 7, nrow = 0))
for (i in colnames(dji30ret)){
  stock_df <- as.data.frame(dji30ret[[i]], row.names = row.names(dji30ret))
  stock_df <- stock_df %>% bind_cols(data.frame(t(ldply(1:5, lag, x= (dji30re
t[,i])))))
```

```
  stock_df <- na.omit(stock_df)
  stock_df[7] <- i
  df_main <- rbind(df_main, stock_df)
  }

colnames(df_main) = c('Daily_Returns', paste0("Lagged_returns_", 1:5), 'Ticke
r_')

for (stock in 2:length(colnames(dji30ret))){
  df_main[stock+6] = dummy_cols(df_main$Ticker, remove_first_dummy = TRUE)[st
ock]
  colnames(df_main)[stock+6] = c(paste0(colnames(dji30ret)[stock],'_dummy'))}
head(df_main)
```

```
##            Daily_Returns Lagged_returns_1 Lagged_returns_2 Lagged_returns_
3
## 1987-03-23   0.002962965     -0.002962965      0.020927520      0.00000000
0
## 1987-03-24   0.017595762      0.002962965     -0.002962965      0.02092752
0
## 1987-03-25   0.005797118      0.017595762      0.002962965     -0.00296296
5
## 1987-03-26  -0.032308243      0.005797118      0.017595762      0.00296296
5
## 1987-03-27   0.000000000     -0.032308243      0.005797118      0.01759576
2
## 1987-03-30  -0.024170361      0.000000000     -0.032308243      0.00579711
8
##            Lagged_returns_4 Lagged_returns_5 Ticker_ AXP_dummy BA_dummy
## 1987-03-23      0.024466052     -0.024466052      AA         0        0
## 1987-03-24      0.000000000      0.024466052      AA         0        0
## 1987-03-25      0.020927520      0.000000000      AA         0        0
## 1987-03-26     -0.002962965      0.020927520      AA         0        0
## 1987-03-27      0.002962965     -0.002962965      AA         0        0
## 1987-03-30      0.017595762      0.002962965      AA         0        0
##            BAC_dummy C_dummy CAT_dummy CVX_dummy DD_dummy DIS_dummy GE_dum
my
## 1987-03-23         0       0         0         0        0         0
0
## 1987-03-24         0       0         0         0        0         0
0
## 1987-03-25         0       0         0         0        0         0
0
## 1987-03-26         0       0         0         0        0         0
0
## 1987-03-27         0       0         0         0        0         0
0
## 1987-03-30         0       0         0         0        0         0
0
##            GM_dummy HD_dummy HPQ_dummy IBM_dummy INTC_dummy JNJ_dummy JPM_
```

```
dummy
## 1987-03-23        0        0         0         0          0         0
0
## 1987-03-24        0        0         0         0          0         0
0
## 1987-03-25        0        0         0         0          0         0
0
## 1987-03-26        0        0         0         0          0         0
0
## 1987-03-27        0        0         0         0          0         0
0
## 1987-03-30        0        0         0         0          0         0
0
##            AIG_dummy KO_dummy MCD_dummy MMM_dummy MRK_dummy MSFT_dummy
## 1987-03-23         0        0         0         0         0          0
## 1987-03-24         0        0         0         0         0          0
## 1987-03-25         0        0         0         0         0          0
## 1987-03-26         0        0         0         0         0          0
## 1987-03-27         0        0         0         0         0          0
## 1987-03-30         0        0         0         0         0          0
##            PFE_dummy PG_dummy T_dummy UTX_dummy VZ_dummy WMT_dummy XOM_dum
my
## 1987-03-23         0        0       0         0        0         0
0
## 1987-03-24         0        0       0         0        0         0
0
## 1987-03-25         0        0       0         0        0         0
0
## 1987-03-26         0        0       0         0        0         0
0
## 1987-03-27         0        0       0         0        0         0
0
## 1987-03-30         0        0       0         0        0         0
0
```

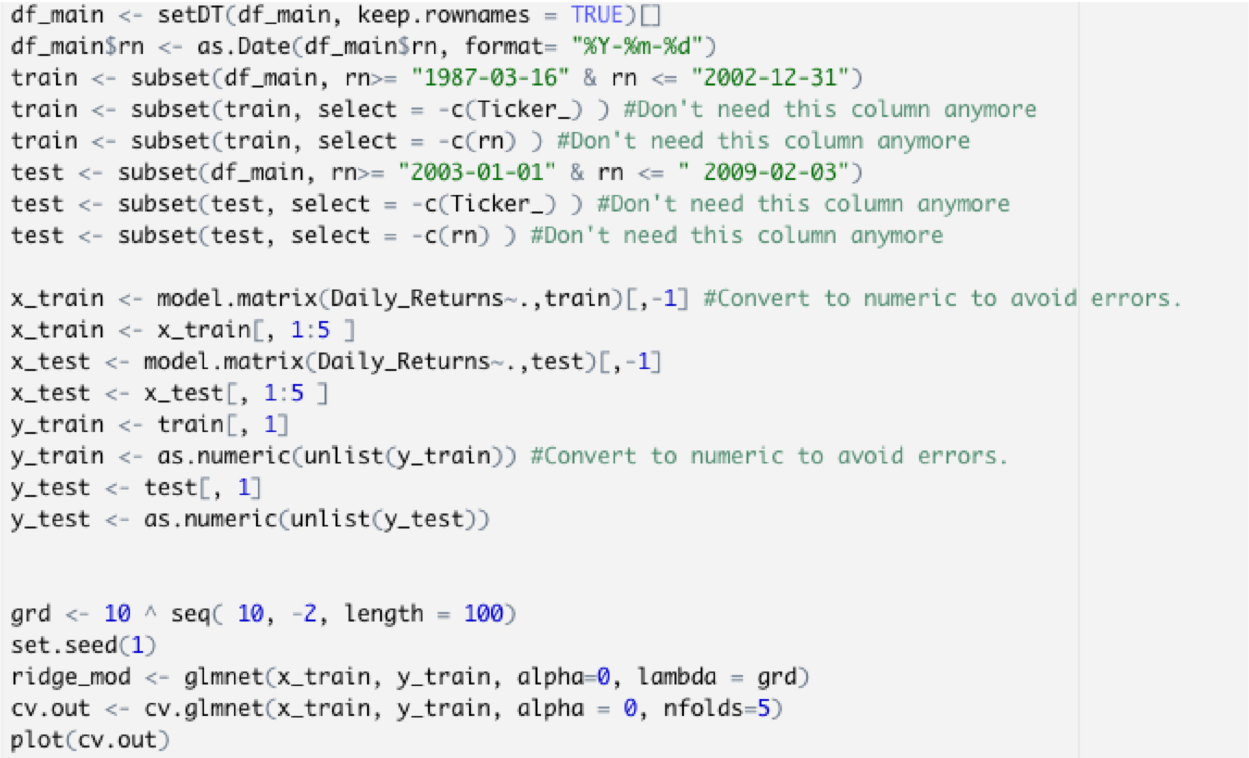
```
print(summary(df_main))
```

```
##  Daily_Returns        Lagged_returns_1     Lagged_returns_2
##  Min.   :-0.9362581   Min.   :-0.9362581   Min.   :-0.9362581
##  1st Qu.:-0.0095356   1st Qu.:-0.0095331   1st Qu.:-0.0095304
##  Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
##  Mean   : 0.0003194   Mean   : 0.0003207   Mean   : 0.0003238
##  3rd Qu.: 0.0103353   3rd Qu.: 0.0103340   3rd Qu.: 0.0103346
##  Max.   : 0.4572902   Max.   : 0.4572902   Max.   : 0.4572902
##  Lagged_returns_3     Lagged_returns_4     Lagged_returns_5
##  Min.   :-0.9362581   Min.   :-0.9362581   Min.   :-0.936258
##  1st Qu.:-0.0095253   1st Qu.:-0.0095199   1st Qu.:-0.009524
##  Median : 0.0000000   Median : 0.0000000   Median : 0.000000
##  Mean   : 0.0003293   Mean   : 0.0003395   Mean   : 0.000332
##  3rd Qu.: 0.0103360   3rd Qu.: 0.0103414   3rd Qu.: 0.010336
##  Max.   : 0.4572902   Max.   : 0.4572902   Max.   : 0.457290
##    Ticker_            AXP_dummy          BA_dummy          BAC_dummy
##  Length:165480      Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  Class :character   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Mode  :character   Median :0.00000   Median :0.00000   Median :0.00000
##                     Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##                     3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##                     Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##     C_dummy          CAT_dummy         CVX_dummy          DD_dummy
##  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   :0.03333   Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##  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
##    DIS_dummy          GE_dummy          GM_dummy          HD_dummy
##  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   :0.03333   Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##  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
##    HPQ_dummy         IBM_dummy         INTC_dummy        JNJ_dummy
##  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   :0.03333   Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##  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
##    JPM_dummy         AIG_dummy          KO_dummy         MCD_dummy
##  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   :0.03333   Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##  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
##    MMM_dummy         MRK_dummy         MSFT_dummy        PFE_dummy
##  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   :0.03333   Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##  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
##     PG_dummy          T_dummy          UTX_dummy          VZ_dummy
##  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   :0.03333   Mean   :0.03333   Mean   :0.03333   Mean   :0.03333
##  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
##    WMT_dummy          XOM_dummy
##  Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000
##  Mean   :0.03333   Mean   :0.03333
##  3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000
```

Question 3A (continued)

```
print(dim(df_main))

## [1] 165480     36
```

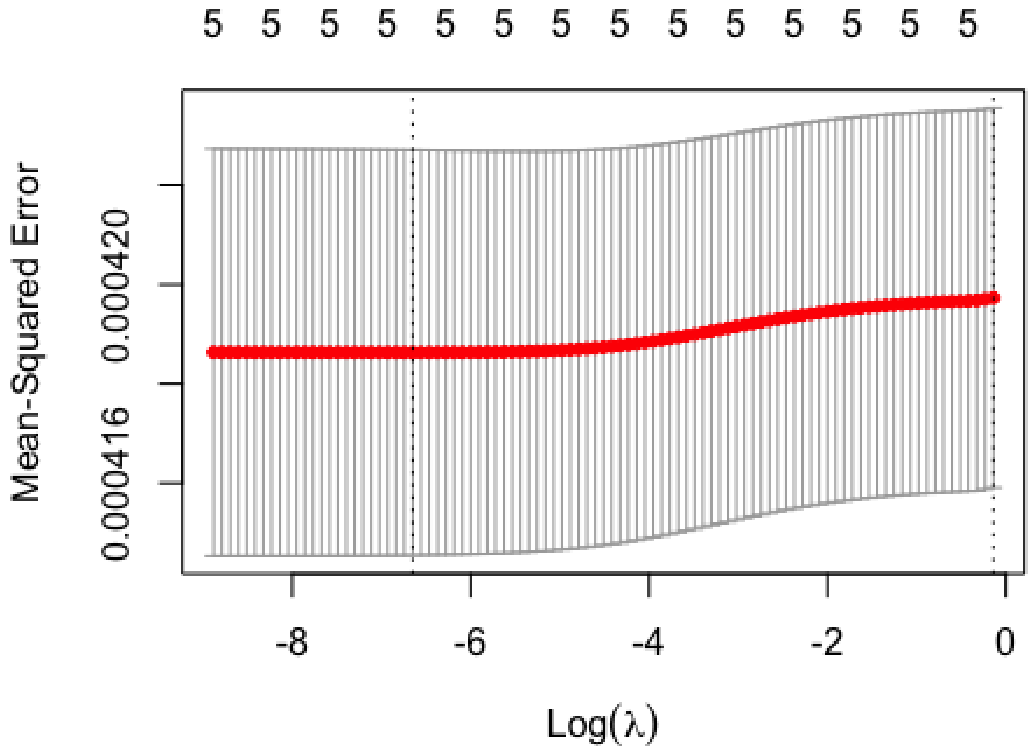
Question 3B



```
df_main <- setDT(df_main, keep.rownames = TRUE)□
df_main$rn <- as.Date(df_main$rn, format= "%Y-%m-%d")
train <- subset(df_main, rn>= "1987-03-16" & rn <= "2002-12-31")
train <- subset(train, select = -c(Ticker_) ) #Don't need this column anymore
train <- subset(train, select = -c(rn) ) #Don't need this column anymore
test <- subset(df_main, rn>= "2003-01-01" & rn <= " 2009-02-03")
test <- subset(test, select = -c(Ticker_) ) #Don't need this column anymore
test <- subset(test, select = -c(rn) ) #Don't need this column anymore

x_train <- model.matrix(Daily_Returns~.,train)[,-1] #Convert to numeric to avoid errors.
x_train <- x_train[, 1:5 ]
x_test <- model.matrix(Daily_Returns~.,test)[,-1]
x_test <- x_test[, 1:5 ]
y_train <- train[, 1]
y_train <- as.numeric(unlist(y_train)) #Convert to numeric to avoid errors.
y_test <- test[, 1]
y_test <- as.numeric(unlist(y_test))


grd <- 10 ^ seq( 10, -2, length = 100)
set.seed(1)
ridge_mod <- glmnet(x_train, y_train, alpha=0, lambda = grd)
cv.out <- cv.glmnet(x_train, y_train, alpha = 0, nfolds=5)
plot(cv.out)
```

[3]

[3] For some reason, R didn't compile this block of code, so I had to take a screenshot.



Question 3B (continued)

```
bestlam <- cv.out$lambda.min
print(bestlam)

## [1] 0.001291421
```

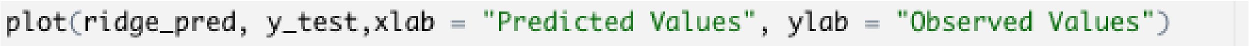
Question 3B (continued)

```
grd <- 10 ^ seq( 10, -2, length = 100)
set.seed(1)
ridge_mod <- glmnet(x_train, y_train, alpha=0, lambda = bestlam)
print(summary(ridge_mod))

##           Length Class     Mode
## a0        1      -none-    numeric
## beta      5      dgCMatrix S4
## df        1      -none-    numeric
## dim       2      -none-    numeric
## lambda    1      -none-    numeric
## dev.ratio 1      -none-    numeric
## nulldev   1      -none-    numeric
## npasses   1      -none-    numeric
## jerr      1      -none-    numeric
## offset    1      -none-    logical
## call       5      -none-     call
## nobs       1      -none-     numeric
```

Question 3C

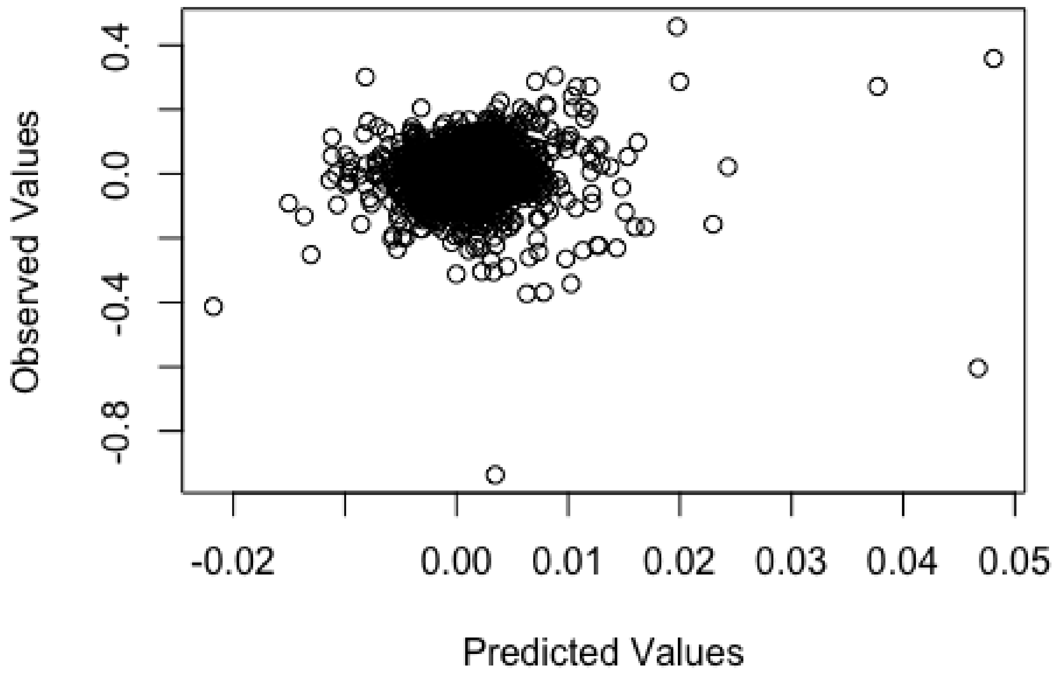
```
set.seed(1)
ridge_pred <- predict(ridge_mod, s = 0, newx = x_test)
print(mae(y_test, ridge_pred))

## [1] 0.0127123
```

Question 3C (continued)



```
plot(ridge_pred, y_test,xlab = "Predicted Values", ylab = "Observed Values")
```

[4]



Conclusions: Compared to our previous model, we notice that this model performs better in terms of MAE, with a lower MAE of 0.0127. One reasonable explanation for a lower MAE could be that we have more observations. Also note that the mean daily average return for

[4] For some reason, R didn't compile this block of code, so I had to take a screenshot.

this data set is 0.0003194, which is 0.03194%, and since our average ABSOLUTE error is 1.27%, this model is still a very poor model (although still better than the previous model). This is because the number of observations is a lot higher than the number of predictors and the relationship between the past 5 lagged returns and daily returns is again almost never linear, so a ridge regression is not a viable choice at the end of the day.

Question 3D

Yes. We can include the dummy variables in our data set as possible features that could help us predict the daily returns. This will increase the number of predictors and add complexity to our model, which theoretically should reduce bias and improve the error. On a more intuitive level, the stock ticker (stock name) should contain at some level of information about the company, which can lead to better fit of the model. Moreover, we could also include interactions between dummy variables to capture industry-wide trends (e.g. interact all technology stocks), which may add robustness to the model.