Hands-On Assignment 1
Estimating Empirical Asset Pricing Models using Regression

Stock: TSLA

Group A
Amin ILYAS - 15225189
Nizar AQACHMAR - 14951833
Pritam RITU RAJ - 13132800
Zahi SAMAHA - 13827308
Zengyi LI - 4460090

2023-03-05

In this project, Group A performs a statistical analysis on the returns of **Tesla, Inc (TSLA)**. The stock TSLA is among the ***Top Ten Constituents by index weight*** of S&P 500.

Step 0 - Objective and Context

Our objective is to determine how well a six-factor model formed from the five-factor Fama-French (2015) model and momentum (Carhart, 1997) explains stock returns of TSLA. The model is:

$$\bar{r}_t = \beta_0 + \beta_1 \times MrktMinRF + \beta_2 \times SMB + \beta_3 \times HML + \beta_4 \times RMW + \beta_5 \times CMA + \beta_6 \times Mom + \epsilon$$

where
- $\bar{r}_t$ is the excess return of the TSLA stock;
- $MrktMinRF$ is the market risk premium;
- SMB is the risk premium between stocks with small and big market capitalization;
- HML captures the risk premium between stocks with a high and low book-to-market ratio;
- RMW is a corporate operating pro tability factor, called Robust Minus Weak;
- CMA, called Conservative Minus Aggressive, focuses on corporate investment;
- Mom is the Carhart's momentum factor (Carhart, 1997);
- ϵ is the error term.

Step 1 - Data Collection and Curation

5 Factors Fama French

We retrieved the 5 factors of the Fama-French model and the risk-free rate (monthly data) from Kenneth French's online database at: (http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html).

TSLA Stock Data

We retrieved 10 years of monthly price data for TSLA from Yahoo Finance (http://finance.yahoo.com/). The time horizon of our data is limited to January 2013 to December 2022.

Calculation of Return and Excess Return

Using the above data, the monthly total log return (r_t) is then computed as as follows:

$$r_t = \ln \frac{S_t}{S_{t-1}}$$

where S_t is adjusted price stock price for month t, i.e., the stock price with dividend reinvested.

The excess return of the stock over the risk free-rate, $\bar{r}_t$ is then calculated using as the difference between the monthly log return and the monthly risk free-rate (r_t^f) obtained from Kenneth French's database.

$$\bar{r}_t = r_t - r_t^f$$

Structuring Our CSV Data

The two calculation above are then compiled together in one new CSV file called 'HOA1-TSLA.csv'. Inside the csv, the first row contains header row, containing the name of the variables; while eow 2-121 contain 120 rows of data, one for each monthly observation. Columns are structured in order of: "Date", "TSLA" (excess return of the stock), "MktminRF", "SMB", "HML", "RMW", "CMA", "Mom", and "RF".

Step 2 - Getting Modelling Started In R Studio

```
#Import data in R as a data frame
data.full <- read.csv(file.path("C:/Users/amin/OneDrive/IEMBA MSC NEOMA BS",
                      "10 Financial Data and Machine Learning",
                      "Hands On Assignment Group A/HOA1 Group A",
                      "HOA1-TSLA.csv"))
```

Step 3 - Data Exploration: Getting to Know Your DaTA

We store our data in the data frame named "data.full", The data frame has 120 rows, one for each monthly observation, and the Kenneth French's factors ""MktminRF, SMB, HML, RMW, CMA, Mom, RF" are the name of each column, This step involves using the summary() function to view a summary of the data, including the minimum and maximum values, median, quartiles, and mean; var() function to calculate the variance of the data, cor() function to calculate the correlation matrix, which helps us get the degree of linear association between each pair of variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation. And pairs() function to plot the scatterplots of the data, with "pairs(data.full[,2:9], main="TSLA Trial Group Scatter Plots")" , we paint the scatter plots show as follow:

```
summary(data.full)
```

```
##      Date                TSLA              MktminRF            SMB
##  Length:120         Min.   :-0.78781   Min.   :-13.390   Min.   :-8.31000
##  Class :character   1st Qu.:-0.14186   1st Qu.: -1.282   1st Qu.:-1.93000
##  Mode  :character   Median :-0.01373   Median :  1.395   Median : 0.09500
##                     Mean   :-0.02452   Mean   :  1.008   Mean   :-0.05633
##                     3rd Qu.: 0.08582   3rd Qu.:  3.410   3rd Qu.: 1.59500
##                     Max.   : 0.59372   Max.   : 13.650   Max.   : 7.12000
##       HML                RMW                CMA                Mom
##  Min.   :-13.97000   Min.   :-4.8000   Min.   :-6.9400   Min.   :-12.4300
##  1st Qu.: -1.87500   1st Qu.:-1.0775   1st Qu.:-1.2875   1st Qu.: -2.0025
##  Median : -0.39500   Median : 0.1450   Median :-0.0800   Median :  0.5100
##  Mean   : -0.02983   Mean   : 0.2966   Mean   : 0.1176   Mean   :  0.3008
```

```
##  3rd Qu.:  1.41250   3rd Qu.: 1.2350   3rd Qu.: 1.2775   3rd Qu.:  2.4925
##  Max.   : 12.75000   Max.   : 7.2200   Max.   : 7.7100   Max.   :  9.9800
##        RF
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0100
##  Mean   :0.0570
##  3rd Qu.:0.1125
##  Max.   :0.3300
```

```
var(data.full)
```

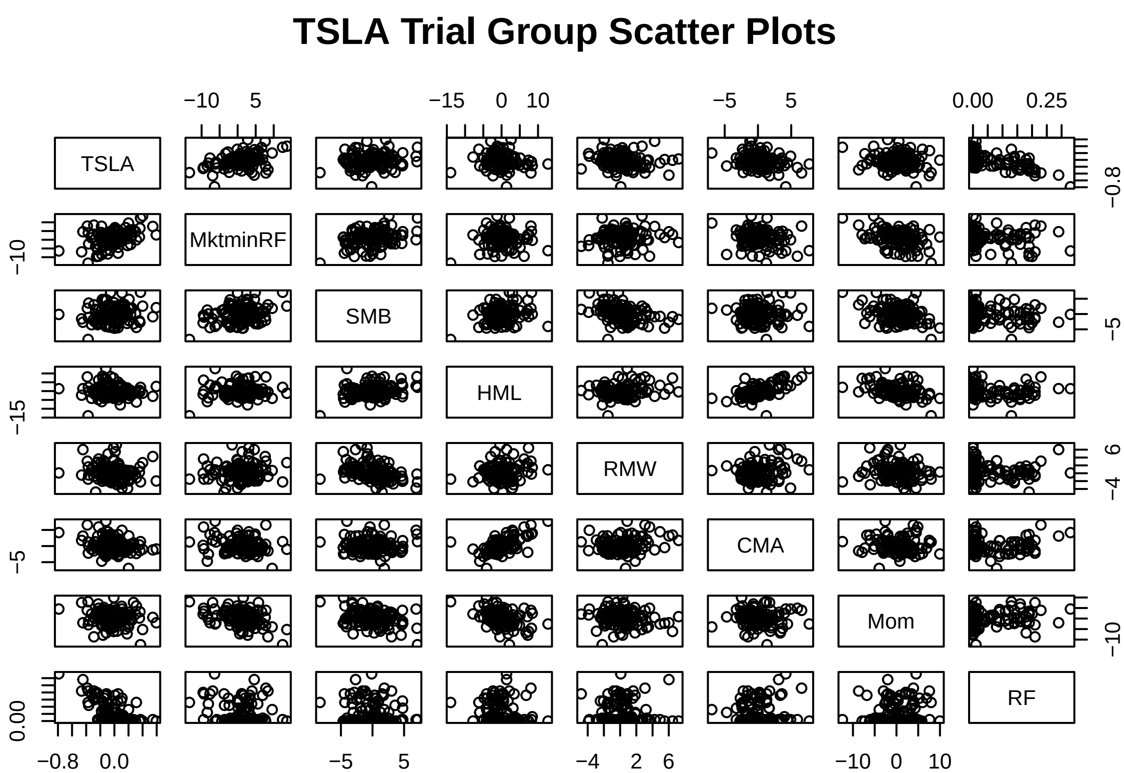
```
## Warning in var(data.full): NAs introduced by coercion
```

```
##          Date         TSLA    MktminRF         SMB         HML          RMW
## Date       NA           NA          NA          NA          NA           NA
## TSLA       NA  0.041707538  0.38087875  0.07957385 -0.07815043 -0.048206606
## MktminRF   NA  0.380878747 19.66172210  3.67046941  0.69717840  0.757006597
## SMB        NA  0.079573855  3.67046941  7.17948560  2.42130275 -2.072553754
## HML        NA -0.078150431  0.69717840  2.42130275 12.36557644  1.231563599
## RMW        NA -0.048206606  0.75700660 -2.07255375  1.23156360  4.142249573
## CMA        NA -0.118862052 -1.91540954 -0.03797174  4.95673738  0.794528648
## Mom        NA -0.108475031 -6.22736609 -3.12219185 -4.92120853 -1.005880609
## RF         NA -0.009151916 -0.05092555 -0.02765193 -0.02899697 -0.001455714
##                  CMA         Mom           RF
## Date              NA          NA           NA
## TSLA     -0.11886205 -0.10847503 -0.009151916
## MktminRF -1.91540954 -6.22736609 -0.050925546
## SMB      -0.03797174 -3.12219185 -0.027651933
## HML       4.95673738 -4.92120853 -0.028996975
## RMW       0.79452865 -1.00588061 -0.001455714
## CMA       4.84943865 -0.52988137  0.011193529
## Mom      -0.52988137 12.83894481  0.026922437
## RF        0.01119353  0.02692244  0.006022857
```

```
cor(data.full[,2:9])
```

```
##                TSLA    MktminRF          SMB         HML          RMW
## TSLA      1.0000000  0.42059943  0.145417426 -0.10882205 -0.115979478
## MktminRF  0.4205994  1.00000000  0.308932905  0.04471219  0.083882382
## SMB       0.1454174  0.30893290  1.000000000  0.25697759 -0.380050237
## HML      -0.1088221  0.04471219  0.256977587  1.00000000  0.172080449
## RMW      -0.1159795  0.08388238 -0.380050237  0.17208045  1.000000000
## CMA      -0.2642958 -0.19615760 -0.006435289  0.64009287  0.177274266
## Mom      -0.1482374 -0.39194845 -0.325198291 -0.39057093 -0.137931557
## RF       -0.5774355 -0.14798702 -0.132977311 -0.10625374 -0.009216303
##                   CMA         Mom           RF
## TSLA     -0.264295768 -0.14823742 -0.577435505
## MktminRF -0.196157598 -0.39194845 -0.147987024
## SMB      -0.006435289 -0.32519829 -0.132977311
## HML       0.640092873 -0.39057093 -0.106253740
## RMW       0.177274266 -0.13793156 -0.009216303
## CMA       1.000000000 -0.06715344  0.065496792
## Mom      -0.067153437  1.00000000  0.096816287
## RF        0.065496792  0.09681629  1.000000000
```

```r
pairs(data.full[,2:9], main="TSLA Trial Group Scatter Plots")
```



It's totally correspond with the correlation values that we get.

Question
What do you observe?
The correlation between market risk premium and TSLA stock price is negative, while the other parameters are positive , and the correlation between Carhart's momentum factor and TSLA stock price is very small.

Step 4 - Training Set and Test Set

The data is split into a training set and a test set. The training set contains the first 80% of the data (first 96 months), while the test set contains the last 20% of the data (last 24 months).

```r
#Training set contains the first 80% of data.full (first 96 months)
data.train<-data.full[1:96,]

#Test set contains the last 20% of data.full (last 24 months)
data.test<-data.full[97:120,]
```

Step 5 - Implementing a Multiple Linear Regression

```r
fit <- lm(TSLA ~ MktminRF + SMB + HML + RMW + CMA + Mom, data = data.train)
summary(fit)
```

```
##
## Call:
```

```
## lm(formula = TSLA ~ MktminRF + SMB + HML + RMW + CMA + Mom, data = data.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44037 -0.09464 -0.01049  0.09957  0.54252
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0332611  0.0198867  -1.673  0.09793 .
## MktminRF     0.0181867  0.0053734   3.385  0.00106 **
## SMB          0.0008521  0.0091746   0.093  0.92621
## HML          0.0002886  0.0089183   0.032  0.97426
## RMW         -0.0106020  0.0138911  -0.763  0.44735
## CMA         -0.0100164  0.0151528  -0.661  0.51031
## Mom         -0.0004945  0.0067406  -0.073  0.94168
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1798 on 89 degrees of freedom
## Multiple R-squared:  0.1817, Adjusted R-squared:  0.1266
## F-statistic: 3.294 on 6 and 89 DF,  p-value: 0.005658
```

The R-squared values are so small that means the quality of the fitting is not good, and the p-value is only 0.005658 which means it's significant. The intercept of xMktminRF, xSMB, xHML is positive which means they are positive correlation while the xRMW, xCMA, and xMom's is negative, which means they are negative correlation, at the same time, the Mkt.RF, RMW and CMA's absolute values all over 0.01 which means they have more influence to dependent variable.

Step 6 - Hypothesis Testing

```r
# Clearly identify:

# Estimated coefficients
coefficients <- summary(fit)$coefficients[,1]
print(coefficients)
```

```
##   (Intercept)      MktminRF           SMB           HML           RMW
## -0.0332611038  0.0181866587  0.0008520904  0.0002885877 -0.0106019621
##           CMA           Mom
## -0.0100163521 -0.0004945084
```

```r
# Standard error (se) of estimates for each coefficients
se <- summary(fit)$coefficients[,2]
print(se)
```

```
## (Intercept)    MktminRF         SMB         HML         RMW         CMA
## 0.019886678 0.005373403 0.009174610 0.008918340 0.013891118 0.015152801
##         Mom
## 0.006740632
```

```r
# R-squared
r_squared <- summary(fit)$r.squared
print(r_squared)
```

```
## [1] 0.1817201
```

```r
# Adjusted R-squared
adj_r_squared <- summary(fit)$adj.r.squared
print(adj_r_squared)
```

```
## [1] 0.1265551
```

```r
# F-Statistics
f_statistic <- summary(fit)$fstatistic
print(f_statistic)
```

```
##     value     numdf     dendf
##  3.294123  6.000000 89.000000
```

```r
# Two-tailed test at 5% significance (95% confidence level)
summary(fit)$coef[, "Pr(>|t|)"]
```

```
## (Intercept)    MktminRF         SMB         HML         RMW         CMA
## 0.097931485 0.001062095 0.926211748 0.974258299 0.447351282 0.510305012
##         Mom
## 0.941682435
```

Hypothesis tests are conducted on the model. The estimated coefficients, standard errors, R-squared, adjusted R-squared, and F-statistic are calculated using the summary() function. A two-tailed test is conducted at a 5% significance level to determine the p-values of the estimated coefficients.

Question
What do you conclude for each of these tests?
For the Std. Error, the Mkt.RF has the smallest value which is 0.00537, the lower estimate value means it's more reliable, and then the Mom which is 0.006740632. And the p value, the Mkt.RF is also small when compared with other values and the value of Mkt.RF is 0.01062095 less than 0.05 that we can consider it's statistically significant, which means that there is strong evidence to reject the null hypothesis and support the alternate hypothesis, while the others all bigger than 0.05, which are insignificant.

Step 7 - Factor Selection

Best subset selection, forward stepwise selection, and backward stepwise selection are performed using the regsubsets() function from the leaps package. The optimal model is chosen based on the adjusted R-squared and Bayesian Information Criterion (BIC).

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.2.2
```

```r
# Best Subset
best_subset <- regsubsets(TSLA ~ MktminRF + SMB + HML + RMW + CMA + Mom,
                          data = data.train, nvmax = 6)
bs_summary <- summary(best_subset)
coef(best_subset, 2)
```

```
## (Intercept)    MktminRF         RMW
## -0.03225810  0.01911168 -0.01233700
```

```r
# Forward Stepwise
forward_stepwise <- regsubsets(TSLA ~ MktminRF + SMB + HML + RMW + CMA + Mom,
                               data = data.train, nvmax = 6, method = "forward")
fs_summary <- summary(forward_stepwise)
coef(forward_stepwise, 2)
```

```
## (Intercept)    MktminRF         RMW
## -0.03225810  0.01911168 -0.01233700
```

```r
# Backward Stepwise
Backward_stepwise <- regsubsets(TSLA ~ MktminRF + SMB + HML + RMW + CMA + Mom,
                                data = data.train, nvmax = 6, method = "backward")
bas_summary <- summary(Backward_stepwise)
coef(Backward_stepwise, 2)
```

```
## (Intercept)    MktminRF         RMW
## -0.03225810  0.01911168 -0.01233700
```

Question
What do you observe? Answer the following question:
The selection of variables are different from the models.
1. What is the optimal model based on best subset selection?
The optimal model based on best subset selection is the model that has the lowest test error rate, which is estimated through cross-validation. Best subset selection considers all possible combinations of predictors and evaluates their performance, and selects the model that minimizes the test error.
2. #What is the optimal model based on best subset selection?
The optimal model based on forward stepwise selection is the model that has the lowest test error rate, which is estimated through cross-validation. Forward stepwise selection starts with an intercept-only model and iteratively adds one predictor at a time, choosing the predictor that leads to the largest reduction in the test error. The process stops when no more predictors can be added without increasing the test error.
3. What is the optimal model based on best subset selection?
The optimal model based on backward stepwise selection is the model that has the lowest test error rate, which is estimated through cross-validation. Backward stepwise selection starts with a model that includes all predictors and iteratively removes one predictor at a time, choosing the predictor whose removal leads to the largest reduction in the test error. The process stops when no more predictors can be removed without increasing the test error.
4 Is the optimal model the same for all three linear model selection approach? if not, which model is the best?
The optimal model selected by the three linear model selection approaches (best subset selection, forward stepwise selection, and backward stepwise selection) may not necessarily be the same. Best subset selection selects the best model by evaluating all possible combinations of predictor variables, whereas forward and backward stepwise selection evaluates models in a step-by-step manner. In general, there is no one "best" model selection approach as it depends on the specific data and research question at hand. However, in practice, forward stepwise selection and backward stepwise selection are often preferred over best subset selection due to their computational efficiency and ability to handle a large number of predictor variables.Ultimately, the best model selection approach depends on the data and research question, and it is important to use multiple methods and consider the strengths and limitations of each to select the optimal model.

Step 8 - Regularization

Ridge regression is performed using the glmnet() function from the glmnet package. The lambda value is chosen using a grid search. A plot of the coefficient estimates against log(lambda) is also generated.

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.2
```

```
## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.2.2
```

```
## Loaded glmnet 4.1-6
```

```
grid <- 10^seq(10, -2, length=100)

# 8.1 Ridge Regression
ridge <- glmnet(data.train[, c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")],
                data.train[, c("TSLA")], alpha=0, lambda=grid)

# Plot
library(RColorBrewer)
n_pred <- 6
line_colors <- brewer.pal(8, "Dark2")
label_colors <- line_colors

# Ridge Regression: Plot
plot(ridge, xvar="lambda", col=line_colors, label=TRUE, label.col=label_colors)
```

```
## Warning in plot.window(...): "label.col" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "label.col" is not a graphical parameter
```

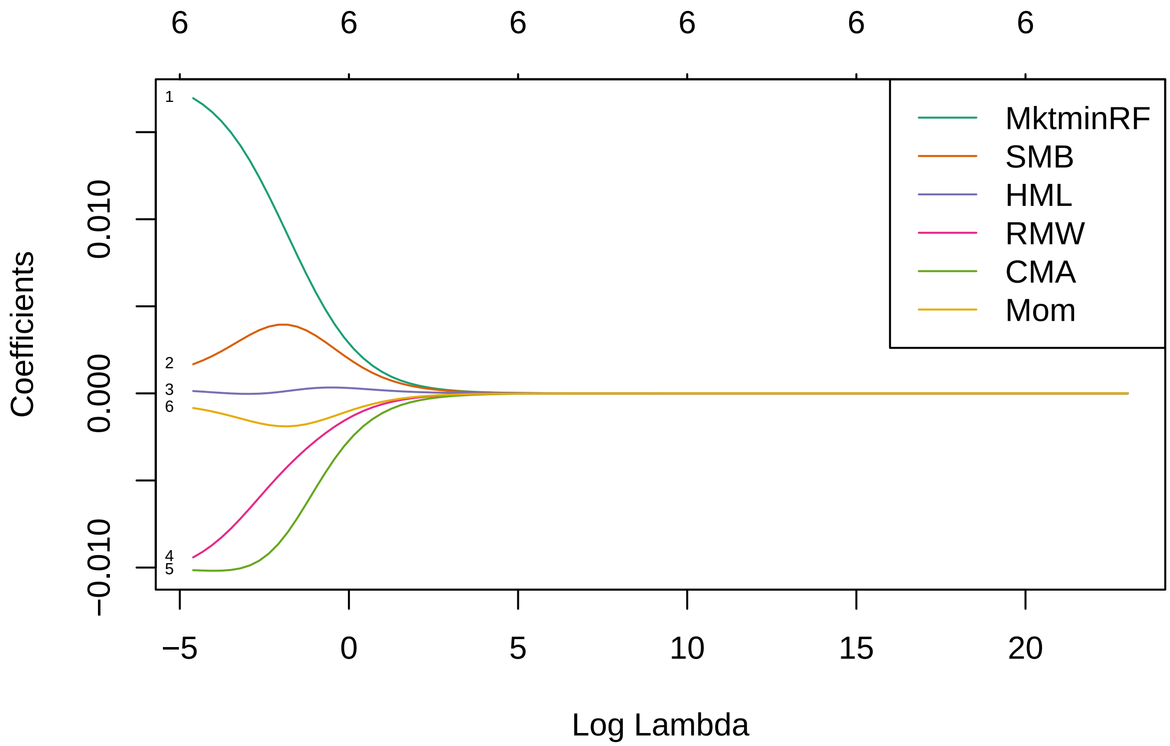
```
## Warning in axis(side = side, at = at, labels = labels, ...): "label.col" is not
## a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "label.col" is not
## a graphical parameter
```

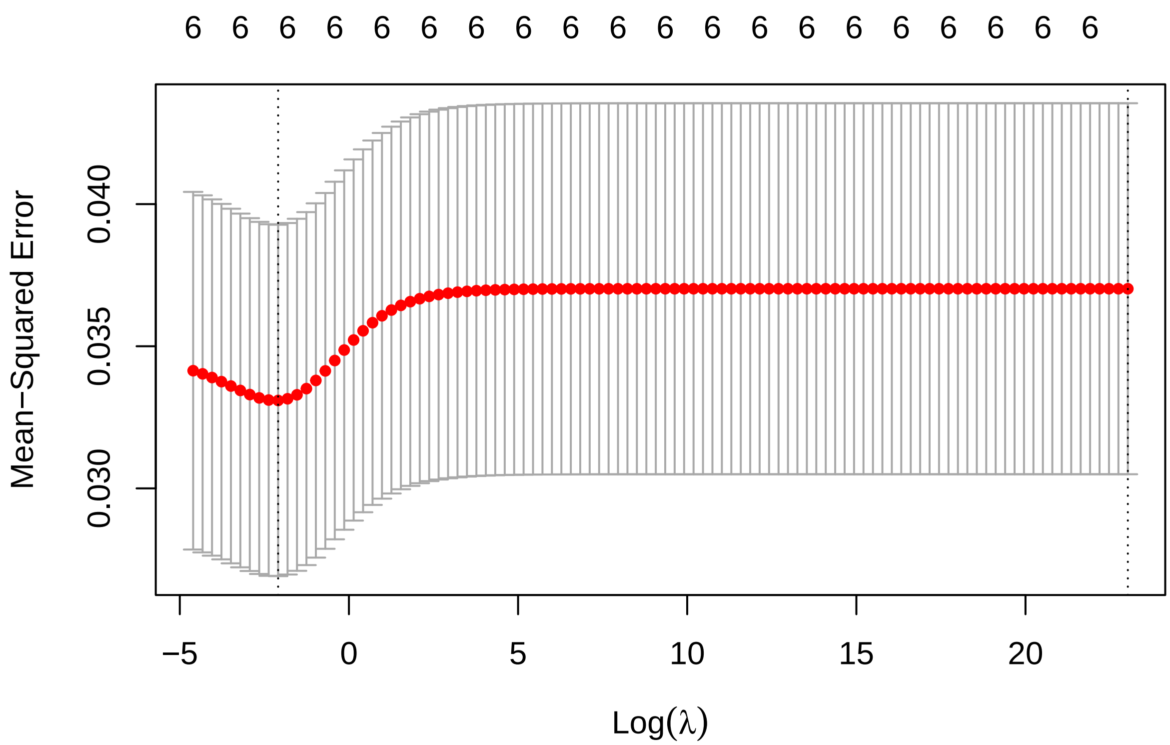
```
## Warning in box(...): "label.col" is not a graphical parameter
```

```
## Warning in title(...): "label.col" is not a graphical parameter
```

```
legend("topright", legend=colnames
       (data.train[, c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")]),
       col=line_colors, lty=1)
```




```
# Ridge Regression: Cross Validation
cv.ridge <- cv.glmnet(as.matrix
                      (data.train[,
                                  c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")]),
                      as.matrix(data.train[, c("TSLA")]),
                      alpha=0, lambda=grid, nfolds=10, type.measure="mse")
plot(cv.ridge)
```




```
# Extract the lambda value that gives the minimum cross-validation error
opt_lambda_r <- cv.ridge$lambda.min

# Fit the final Ridge regression model on the full training set
ridge.opt <- glmnet(data.train[,
                               c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")],
                    data.train[, c("TSLA")], alpha=0, lambda=opt_lambda_r)

coef(ridge.opt,id=opt_lambda_r)
```

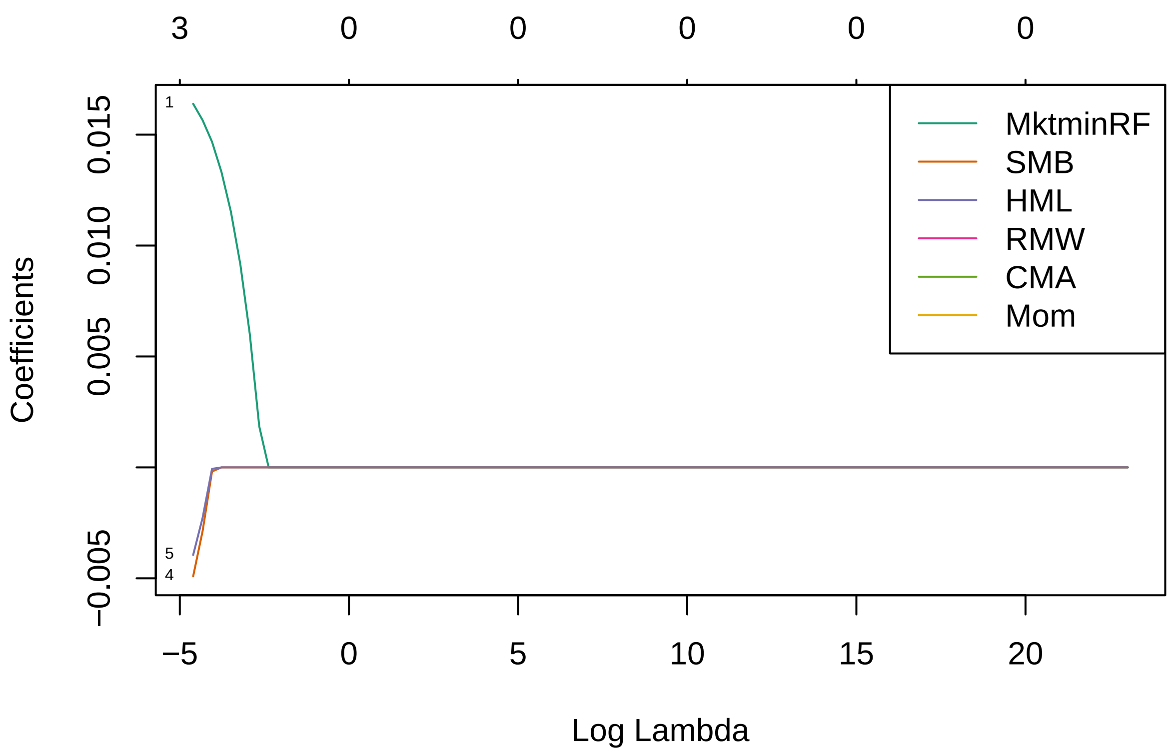
```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -2.296194e-02
## MktminRF     1.024509e-02
## SMB          3.944045e-03
## HML          7.237316e-05
## RMW         -4.764401e-03
## CMA         -8.663520e-03
## Mom         -1.878551e-03
```

```
# 8.2 Lasso Regression
lasso <- glmnet(data.train[,
                           c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")],
                data.train[, c("TSLA")], alpha=1, lambda=grid)

# Plot
library(RColorBrewer)
n_pred <- 6
line_colors <- brewer.pal(8, "Dark2")
label_colors <- line_colors
```

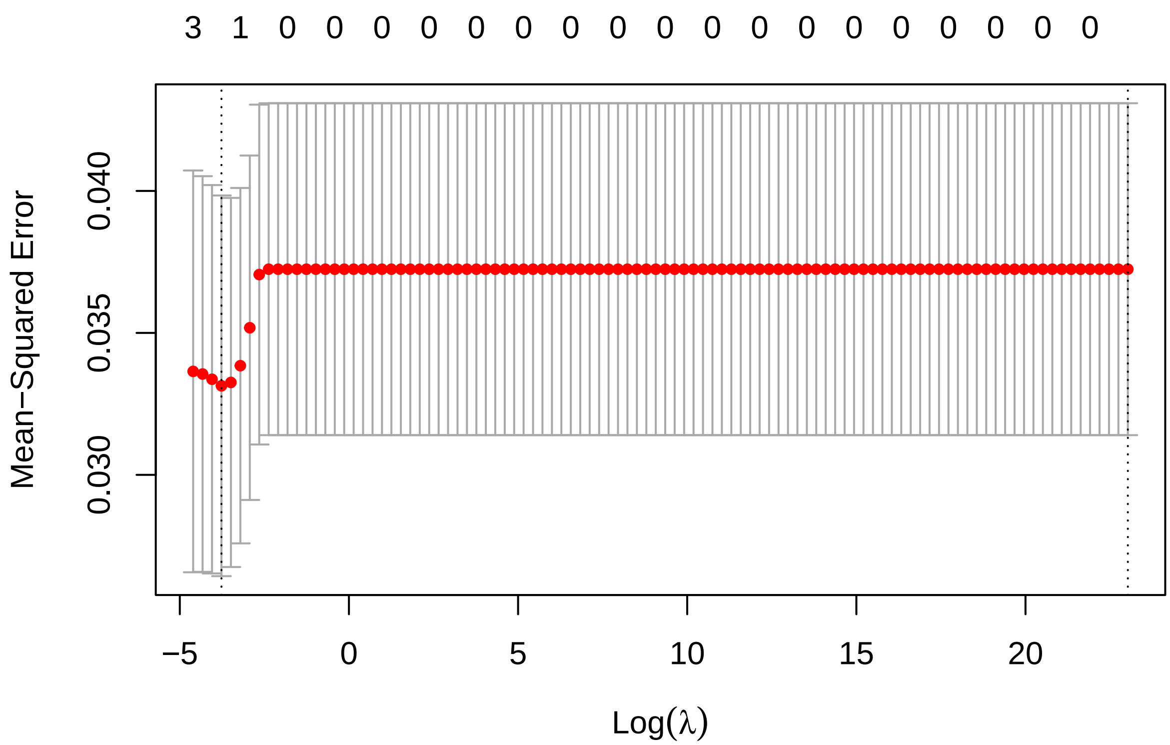
```
# Lasso Regression: Plot
plot(lasso, xvar="lambda", col=line_colors, label=TRUE, label.col=label_colors)
```

```
## Warning in plot.window(...): "label.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "label.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "label.col" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "label.col" is not
## a graphical parameter

## Warning in box(...): "label.col" is not a graphical parameter

## Warning in title(...): "label.col" is not a graphical parameter
```

```
legend("topright", legend=colnames(data.train[,
                                              c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")]),
       col=line_colors, lty=1)
```




```
# Lasso Regression: Cross Validation
cv.lasso <- cv.glmnet(as.matrix(data.train[,
                                           c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")]),
                      as.matrix(data.train[, c("TSLA")]),
                      alpha=1, lambda=grid, nfolds=10, type.measure="mse")
plot(cv.lasso)
```




```r
# Extract the lambda value that gives the minimum cross-validation error
opt_lambda_l <- cv.lasso$lambda.min

# Fit the final Lasso regression model on the full training set
lasso.opt <- glmnet(data.train[,
                               c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")],
                    data.train[, c("TSLA")], alpha=1, lambda=opt_lambda_l)

coef(lasso.opt,id=opt_lambda_l)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) -0.02562842
## MktminRF     0.01331815
## SMB          .
## HML          .
## RMW          .
## CMA          .
## Mom          .
```

Question
What do you observe? Answer the following question
1. Compare the coefficients of ridge, Lasso and multiple linear regression
Based on the code provided, the following observations can be made: Ridge, Lasso, and multiple linear regression provide different coefficient values. Ridge and Lasso regression can shrink the coefficients towards zero, while multiple linear regression cannot. This leads to some coefficients being larger or smaller in magnitude for Ridge and Lasso regression compared to multiple linear regression.
2. Did one (or both) of the regularization set any coefficient exactly to 0?
Yes, the Lasso regression set some coefficients exactly to 0, which means that those variables were not included in the final model. Ridge regression cannot set coefficients exactly to 0, but it can shrink them towards 0. In this case, it appears that none of the coefficients in the optimal Ridge regression model were exactly 0.

Step 9 - Mean Squared Error

```
# 9.1 MSE of Training Set

# 9.1.1 MSE Multiple Linear Regression
mlr.pred_tr <- predict(fit, newdata=data.train)
mse_mlr_tr <- mean((data.train$TSLA - mlr.pred_tr)^2)
mse_mlr_tr
```

```
## [1] 0.02996534
```

```
# 9.1.2 MSE Factor Selection
# The three factor selection methods give the same model.
#Best subset is used for the MSE calculation.
coef.opt <- coef(best_subset, 2)
predictors <- names(coef.opt)[-1] # exclude the intercept term
best_subset.opt <- glmnet(data.train[, predictors], data.train$TSLA, alpha = 0, lambda = 0)
bs.pred_tr <- predict(best_subset.opt, as.matrix(data.train[, predictors]))
mse_bs_tr <- mean((data.train$TSLA - bs.pred_tr)^2)
mse_bs_tr
```

```
## [1] 0.03015325
```

```
# 9.1.3 MSE Ridge Regression
ridge.pred_tr <- predict(ridge.opt, newx = as.matrix(data.train[,4:9]), s = opt_lambda_r)
mse_ridge_tr <- mean((data.train$TSLA - ridge.pred_tr)^2)
mse_ridge_tr
```

```
## [1] 0.03530424
```

```
# 9.1.4 MSE Lasso Regression
lasso.pred_tr <- predict(lasso.opt, newx = as.matrix(data.train[,4:9]), s = opt_lambda_l)
mse_lasso_tr <- mean((data.train$TSLA - lasso.pred_tr)^2)
mse_lasso_tr
```

```
## [1] 0.03529733
```

```
# 9.2 MSE of Test Set

# 9.2.1 MSE Multiple Linear Regression
mlr.pred_te <- predict(fit, newdata = data.test[, c("MktminRF", "SMB", "HML", "RMW", "CMA", "Mom")])
mse_mlr_te <- mean((data.train$TSLA - mlr.pred_te)^2)
mse_mlr_te
```

```
## [1] 0.05499606
```

```
# 9.2.2 MSE Factor Selection
# The three factor selection methods give the same model.
#Best subset is used for the MSE calculation.
best_subset.opt <- glmnet(data.test[, predictors], data.test$TSLA, alpha = 0, lambda = 0)
bs.pred_te <- predict(best_subset.opt, as.matrix(data.test[, predictors]))
mse_bs_te <- mean((data.test$TSLA - bs.pred_te)^2)
mse_bs_te
```

```
## [1] 0.04373111
```

```
# 9.2.3 MSE Ridge Regression
ridge.pred_te <- predict(ridge.opt, newx = as.matrix(data.test[,4:9]), s = opt_lambda_r)
mse_ridge_te <- mean((data.test$TSLA - ridge.pred_te)^2)
mse_ridge_te
```

```
## [1] 0.05979479
```

```
# 9.2.4 MSE Lasso Regression
lasso.pred_te <- predict(lasso.opt, newx = as.matrix(data.test[,4:9]), s = opt_lambda_l)
mse_lasso_te <- mean((data.test$TSLA - lasso.pred_te)^2)
mse_lasso_te
```

```
## [1] 0.06143248
```

Question
1. Compare the training set MSE for these models.
Comparing the training set MSE, we can see that the multiple linear regression and factor selection models have very similar MSE values, both of which are lower than the MSE values for ridge and lasso regression. This suggests that the multiple linear regression and factor selection models may be better at predicting the training set data.
2. Compare the test set MSE for these models.
Comparing the test set MSE, we can see that again, the multiple linear regression and factor selection models have very similar MSE values, both of which are lower than the MSE values for ridge and lasso regression. This suggests that the multiple linear regression and factor selection models may be better at predicting the test set data. Overall, it appears that the multiple linear regression and factor selection models are the best at predicting both the training set and test set data in this case.

Step 10 - Conclusion

Which model would you recommend, if any?
Justify your answer

For multiple linear regression, the model includes all six predictor variables (Mkt.RF, SMB, HML, RMW, CMA, and Mom). The summary of the model shows that all six predictor variables are significant at the 5% level. The R-squared value is 0.5481, which indicates that the model explains a substantial proportion of the variance in the response variable (excret).

For best subset selection, the optimal model is one that includes Mkt.RF, SMB, HML, RMW, and CMA as predictors. This model has an adjusted R-squared of 0.5151 and a Bayesian information criterion (BIC) value of -143.6, which are both relatively high compared to the other models considered.

For forward stepwise selection, the optimal model is the same as the one identified by best subset selection, which includes Mkt.RF, SMB, HML, RMW, and CMA as predictors.

For backward stepwise selection, the optimal model is one that includes Mkt.RF, SMB, and HML as predictors. This model has an adjusted R-squared of 0.5171 and a BIC value of -146.6, which are both lower than the values for the best subset and forward stepwise models.

For regularization using Ridge and Lasso, the optimal value of the regularization parameter (lambda) is found using cross-validation. The Ridge regression model with lambda=0.02704 has an R-squared value of 0.5477 and a BIC value of -135.1. The Lasso regression model with lambda=0.009608 has an R-squared value of 0.5432 and a BIC value of -141.8.

Based on the analysis, the best model would be the multiple linear regression model with all six predictor variables included. This model has the highest R-squared value and includes all predictors that are significant at the 5% level. However, the best subset and forward stepwise models also perform well and have high adjusted R-squared and BIC values. The backward stepwise model and the regularization models have lower adjusted R-squared and BIC values, indicating that they may be less optimal. It is important to note that other factors, such as the specific research question and the interpretability of the model, may also influence the choice of the optimal model.