STAT 542: Homework 3

Spring 2022, by Kamila Makhambetova (kamilam3)

Due: Thur, Feb 10, 11:59 PM CT

Contents

Instruction

Students are encouraged to work together on homework. However, sharing, copying, or providing any part of a homework solution or code is an infraction of the University's rules on Academic Integrity. Any violation will be punished as severely as possible. Final submissions must be uploaded to compass2g. No email or hardcopy will be accepted. For **late submission policy and grading rubrics**, please refer to the course website.

- What is expected for the submission to **Gradescope**
  - You are required to submit one rendered **PDF** file `HWx_yourNetID.pdf`. For example, `HW01_rqzhu.pdf`. Please note that this must be a `.pdf` file generated by a `.Rmd` file. `.html` format cannot be accepted.
  - Please follow the instructions on Gradescope to select corresponding PDF pages for each question.
- Please note that your homework file is a **PDF** report instead of a messy collection of R codes. This report should **include**:
  - Your Name and NetID. (Replace `Ruoqing Zhu(rqzhu)` by your name and NetID if you are using this template).
  - Make all of your `R` code chunks visible for grading.
  - Relevant outputs from your `R` code chunks that support your answers.
  - Provide clear answers or conclusions for each question. For example, you could start with `Answer: I fit SVM with the following choice of tuning parameters ...`
  - Many assignments require your own implementation of algorithms. **Basic comments are strongly encouraged** to explain the logic to our graders. However, line-by-line code comments are unnecessary.
- Requirements regarding the `.Rmd` file.

  - You do **NOT** need to submit `Rmd` files. However, your PDF file should be rendered directly from it.
  - Make sure that you **set random seeds** for simulation or randomized algorithms so that the results are reproducible. If a specific seed number is not provided in the homework, you can consider using your NetID.
  - For some questions, there will be restrictions on what packages/functions you can use. Please read the requirements carefully. As long as the question does not specify such restrictions, you can use anything.

About HW3

In the first question, we will use a simulation study to confirm the theoretical analysis we developed during the lecture. In the second question, we will practice several linear model selection techniques such as AIC, BIC, and best subset selection. However, some difficulties are at the data processing part, in which we use the Bitcoin data from Kaggle. This is essentially a time-series dataset, and we use the information in previous days to predict the price in a future day. Make sure that you process the data correctly to fit this task.

Question 1 [50 Points] A Simulation Study

Let's use a simulation study to confirm the bias-variance trade-off of linear regressions. Consider the following model.

$$Y = \sum_{j}^{p} 0.8^j \times X_j + \epsilon$$

All the covariates and the error term follow i.i.d. standard Gaussian distribution. The true model involves all the variables; however, larger indexes do not significantly contribute to the variation. Hence, there could be a benefit in using a smaller subset for prediction purposes. Let's confirm that with a simulation study.

- Generate 100 samples of covariates X with $p = 30$ by the following code.

```r
set.seed(542)
n = 100
p = 30
b = 0.8^(1:p)
X = matrix(rnorm(n*p), n, p)
Ytrue = X %*% b
```

- Then the study essentially **repeats** the following steps 100 times. Begin with another fixed random seed before your loop.
  - Using the fixed covariates X, generate 100 training outcomes Y_{train} and 100 testing outcomes Y_{test} independently.
  - Consider using only the first j variables to fit the linear regression (**NO intercept term**). Let j ranges from 1 to 30. Calculate and record the corresponding prediction error by comparing your prediction with the outcomes for testing data.

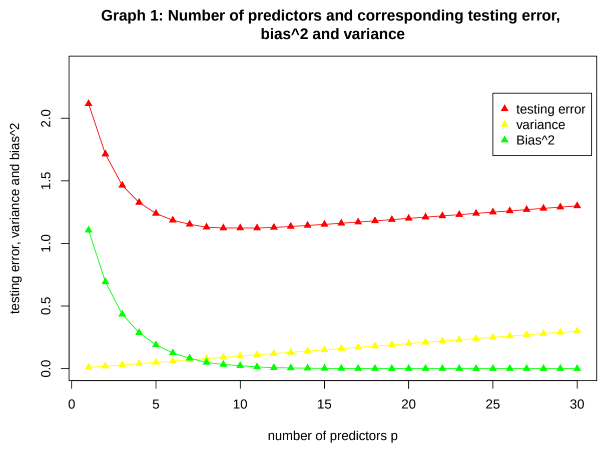
Without running the simulation, for each j value, we also have the theoretical decomposition of the testing error based on the lecture. Suppose you know the true model, covariates X and the distribution of random noise.

a) [15 pts] Please calculate the bias^2 , variance (of the prediction) and testing error for each j based on the theoretical formulas. Plot the 3 lines on the same figure, using the `number of variables` as the x-axis and `bias^2`, `variance`, `theoretical testing error` as the y-axis. Label each line.

- $\text{Bias}^2 = \frac{1}{n}\|E(Y_{\text{pred}}) - Y_{\text{true}}\|^2$, where Y is an $n \times 1$ vector.
- $\text{Var} = \frac{1}{n}E\|Y_{\text{pred}} - E(Y_{\text{pred}})\|^2$.

```
library(matlib)

#error is N(0,1)
sigma=1
Var=rep(0,30)
Bias_sq=rep(0,30)
test_error=rep(0,30)

mew=X%*%b


for(j in 1:30){

  if(j==1){

    #Hat matrix X*( X^T X)^-1 *X^T
    X_tran_X=t(X[ ,1])%*%X[ ,1]
    X_tran_X_inv=1/X_tran_X
    X_mult_trans_X= X[ ,1]%*%t(X[ ,1:j])
    Hat_matrix=as.vector(X_tran_X_inv)*X_mult_trans_X

    #Bias^2 =(m-Hm)^T *(m-Hm)
    Bias_sq[j]=(t((mew-Hat_matrix%*%mew))%*%(mew-Hat_matrix%*%mew))/n

    #Var= p *sigma^2
    Var[j]=(j*sigma*sigma)/n

    #test_error=n*sigma^2+Bias^2 + Var
    test_error[j]= sigma*sigma+Bias_sq[j]+Var[j]

  }else{
    #Hat matrix X*( X^T X)^-1 *X^T
  Hat_matrix=X[ ,1:j]%*%inv(t(X[ ,1:j])%*%X[ ,1:j])%*%t(X[ ,1:j])

  #Bias^2 =(m-Hm)^T *(m-Hm)
  Bias_sq[j]=(t((mew-Hat_matrix%*%mew))%*%(mew-Hat_matrix%*%mew))/n

  #Var= p *sigma^2
  Var[j]=(j*sigma*sigma)/n

  #test_error=n*sigma^2+Bias^2 + Var
  test_error[j]= sigma*sigma+Bias_sq[j]+Var[j]

  }


}
```

```
#Draw graphs

x=1:30


plot(x,test_error, type = "l",xlim=range(c(1,30)), ylim=range(c(0,2.4)), col="red", xlab = "number of p
points(x,test_error, pch=17, col="red")


points(x,Var, pch=17, col="yellow")
lines(x,Var, pch=17, type = "l", col="yellow")

points(x,Bias_sq, pch=17, col="green")
lines(x,Bias_sq, type = "l", col="green")

title(main="Graph 1: Number of predictors and corresponding testing error, \nbias^2 and variance")
legend(25, 2.2,legend=c("testing error","variance", "Bias^2"), col=c("red","yellow", "green"),
                                   pch=c(17,17), ncol=1)
```



b) [5 pts] Report the theoretical testing error with $p = 30$, $\frac{1}{n}E\|Y_{\text{test}} - Y_{\text{pred}}\|^2$.

```
print(paste0("The theoretical testing error when p=30 is ",test_error[30]))
```

```
## [1] "The theoretical testing error when p=30 is 1.30000000000618"
```

Answer

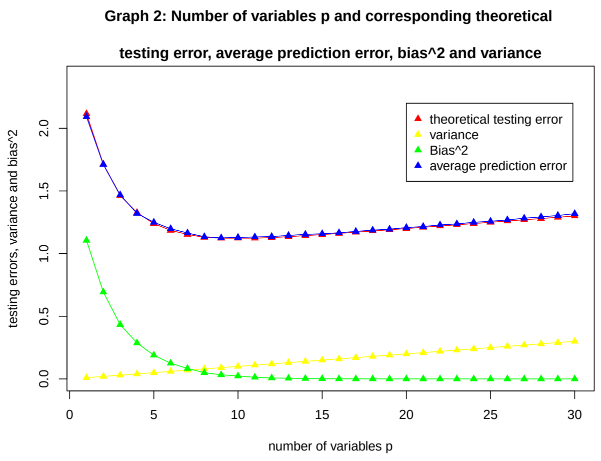
Theoretical testing error for p=30 is 1.30000000000618. It was calculated by using $error_{test} = \frac{n*\theta^2+Bias^2{}_{30}+Var_{30}}{n}$.

After finishing the simulation:

c) [20 pts] Perform the simulation. Report the averaged (empirical) prediction error with $p = 30$. Note that 100 times simulation approximates the E operation. Plot `pred err` in the **same figure** of question a. Label your line. Does your empirical testing error match our theoretical analysis? Comment on your findings.

```r
 set.seed(1998)

Y_train=rep(0,100)
Y_test=rep(0,100)
err_matrix=matrix(0,100, 30)

 #run simulation 100 times to produce 100 results
 for(i in 1:100){

  #produce Y_train and Y_test by adding standard normal distributed errors
  n = 100
  p = 30
  b = 0.8^(1:p)
  Y_train= X %*% b+rnorm(n = 100, mean = 0, sd = 1)
  Y_test= X %*% b+rnorm(n = 100, mean = 0, sd = 1)

  #Run for predictors p from 1 to 30
  for(j in 1:30){

          # construct the data
          traindata = data.frame("x" = X[, 1:j, drop = FALSE], "y" = Y_train)
          testdata = data.frame("x" = X[, 1:j, drop = FALSE], "y" = Y_test)

          # fit model and calculate testing error
          fit1 = lm(y~., data = traindata)
          Y_pred = predict(fit1, testdata)
          err_matrix[i, j] = mean( (Y_pred - testdata$y)^2 )

  }
 }

#calculate average predictor errors for p from 1 to 30
aver_pred_err=rep(0,30)

for(i in 1:30){
  aver_pred_err[i] = mean( err_matrix[ ,i])}

#Plot graphs
x=1:30

plot(x,test_error, type = "l",xlim=range(c(1,30)), ylim=range(c(0,2.4)),
     col="red", xlab = "number of variables p" ,
     ylab = "testing errors, variance and bias^2")
points(x,test_error, pch=17, col="red")

points(x,Var, pch=17, col="yellow")
lines(x,Var, pch=17, type = "l", col="yellow")

points(x,Bias_sq, pch=17, col="green")
lines(x,Bias_sq, type = "l", col="green")

points(x,aver_pred_err, pch=17, col="blue")
lines(x,aver_pred_err, type = "l", col="blue")
```

```r
title(main="Graph 2: Number of variables p and corresponding theoretical
      \ntesting error, average prediction error, bias^2 and variance")
legend(20, 2.2,legend=c("theoretical testing error","variance", "Bias^2",
                        "average prediction error"), col=c("red","yellow",
                                                           "green", "blue"),
                                   pch=c(17,17), ncol=1)
```



```r
print(paste0("The average prediction error when p=30 is ", aver_pred_err[30]))
```

```
## [1] "The average prediction error when p=30 is 1.31860452378273"
```

Comment

The average prediction error when p=30 is 1.31860452378273. It is very close to its theoretical testing error 1.30000000000618. Also, according to Graph 2 average prediction errors for p from 1 to 30 are approximately equal to their corresponding theoretical testing errors.

d) [10 pts] Evaluate the bias^2 for model $p = 5$ without theoretical formulas. You can still assume you know the true outcomes while using the average results to approximate the E operation. Compare the empirical value with the theoretical one.

```r
set.seed(1998)

Y_train=rep(0,100)
Y_test=rep(0,100)
Y_pred_matrix_p5=matrix(0,100, 100)
n_2=1000

k=1

#run simulation 100 times to produce 100 results
for(i in 1:100){

  #produce Y_train and Y_test by adding standard normal distributed errors
  n = 100
  p = 30
  b = 0.8^(1:p)
  Y_train= X %*% b+rnorm(n = 100, mean = 0, sd = 1)
  Y_test= X %*% b+rnorm(n = 100, mean = 0, sd = 1)

  # find 100 Y_pred vectors for p=5
  for(j in 1:5){

          # construct the data
          traindata = data.frame("x" = X[, 1:j, drop = FALSE], "y" = Y_train)
          testdata = data.frame("x" = X[, 1:j, drop = FALSE], "y" = Y_test)

          # fit model and calculate predicted Y
          fit1 = lm(y~., data = traindata)
          Y_pred = predict(fit1, testdata)

          #record 100 Y_pred for p=5
          if(j==5){
            Y_pred_matrix_p5[, k]=Y_pred
            k=k+1
          }

  }
 }

#Find average for  each simulated Y_pred vector, i.e. transform 100x100 Y_pred matrix
#into 100x1 aver_Y_pred vector
aver_Y_pred=rep(0,100)

for(j in 1:100){
  aver_Y_pred[j]=mean(Y_pred_matrix_p5[,j])
}

#Bias^2=||E(Y_pred)-Y_true||/n
bias_sq_p5=(t(aver_Y_pred-Ytrue)%*%(aver_Y_pred-Ytrue))/n
print(paste0("The empirical bias^2 for p=5 is ", bias_sq_p5))
```

```
## [1] "The empirical bias^2 for p=5 is 1.83255771183786"
```

```r
print(paste0("The theoretical bias^2 for p=5 is ", Bias_sq[5]))
```

```
## [1] "The theoretical bias^2 for p=5 is 0.189452475341948"
```

Comment

The emperical bias for calculated by stimulating 100 Y_{pred} vectors, which has size 100x1. Then I found the mean of each Y_{pred} vector and created average predicted vector, which has size 100x1. Then I calculated bias by taking norm of difference between average predicted vector and Y_true and divide it by number of observations n. The empirical $bias^2$ for p=5 = 1.83255771183786. The theoretical $bias^2$ for p=5 is 0.189452475341948. We can see that empirical $bias^2$ 10 times larger than theoretical. I think it is odd. I assume that they should be aprroximately equal to each other.

Question 2 [50 Points] Bitcoin price prediction

For this question, we will use the Bitcoin data provided on the course website. The data were posted originally on Kaggle (link). Make sure that you read relevant information from the Kaggle website. Our data is the `bitcoin_dataset.csv` file. You should use a training/testing split such that your training data is constructed using only information up to 12/31/2016, and your testing data is constructed using only information starting from 01/01/2017. The goal of our analysis is to predict the `btc_market_price`. Since this is longitudinal data, we will use the information from previous days to predict the market price at a future day. In particular, on each calendar day (say, day 1), we use the information from three days onward (days 1, 2, and 3) to predict the market price on the 7th day.

Hence you need to first reconstruct the data properly to fit this purpose. This is mainly to put the outcome (of day 7) and the covariates (of the previous days) into the same row. Note that you may face missing data, categorical predictors, outliers, scaling issues, computational issues, and maybe others for this question. Use your best judgment to deal with them. There is no general "best answer". Hence the grading will be based on whether you provided reasoning for your decision and whether you carried out the analysis correctly.

a. [25 Points] Data Construction. Data pre-processing is usually the most time-consuming and difficult part of an analysis. We will use this example as a practice. Construct your data appropriately such that further analysis can be performed. Make sure that you consider the following:

   - The data is appropriate for our analysis goal: each row contains the outcome on the seventh day and the covariates based on the first three days. The covariates are not limited to the price.
   - Missing data is addressed (you can remove variables, remove observations, impute values or propose your own method)
   - You may process the covariates and/or outcome by considering centering, scaling, transformation, removing outliers, etc. However, these are your choice.

For each of the above tasks, make sure that you **clearly document your choice**. In the end, provide a summary table/figure of your data. You can consider using boxplots, quantiles, histograms, or any method that is easy for readers to understand. You are required to pick at least one method to present.

```r
library(tidyverse)
library(data.table)

#read csv file
bitcoin = read.csv(file = "bitcoin.csv")

#sort by date
bitcoin<-bitcoin%>%arrange(Date)

#create dataset with 2 columns date and market price
bitcoin2<-bitcoin%>%dplyr::select(Date, btc_market_price)

i=7
j=1
l=1

#combine 7th days market price with previous 1st days covariates
bitcoin_day1 <- data.frame(matrix(ncol = 26, nrow = 0))

while(i<=nrow(bitcoin)){
  bitcoin_day1[l,]= bind_cols(bitcoin2[i,],bitcoin[j,])
  i=i+1
  j=j+1
  l=l+1
}

#combine 7th days market price, previous 1st days covariates
#with previous 2nd days covariates
bitcoin_day2<- data.frame(matrix(ncol = 50, nrow = 0))

i=1

while(i<=nrow(bitcoin_day1)){
  bitcoin_day2[i,]= bind_cols(bitcoin_day1[i,],bitcoin[i+1,])
  i=i+1
}

#combine 7th days market price, previous 1st days covariates,
#previous 2nd days covariates with previous 3rd days covariates
bitcoin_final<- data.frame(matrix(ncol = 74, nrow = 0))

i=1

while(i<=nrow(bitcoin_day2)){
  bitcoin_final[i,]= bind_cols(bitcoin_day2[i,],bitcoin[i+2,])
  i=i+1
}

#name cplumns of final table
names(bitcoin_final)[1:2]<-c("Date_day7","btc_market_price_day7")

names(bitcoin_final)[3:26]<-c("Date_day1","btc_market_price_day1","btc_total_bitcoins_day1", "btc_marke

names(bitcoin_final)[27:50]<-c("Date_day2","btc_market_price_day2","btc_total_bitcoins_day2", "btc_mark

names(bitcoin_final)[51:74]<-c("Date_day3","btc_market_price_day3","btc_total_bitcoins_day3", "btc_mark

# drop NA values from final table
bitcoin_final2<-na.omit(bitcoin_final)

# construct train and test tables from final table by filtering date
train<-bitcoin_final2 %>%
    filter(Date_day7<="2016-12-31 00:00:00")%>%
                   arrange(Date_day7)

  test<-bitcoin_final2 %>%
    filter(Date_day7>="2017-01-01 00:00:00")%>%
                   arrange(Date_day7)
#remove date columns
train<-train%>%dplyr::select(-c(Date_day7, Date_day1, Date_day2, Date_day3))
test<-test%>%dplyr::select(-c(Date_day7, Date_day1, Date_day2, Date_day3))
```

Comment

In this part I create new table by combining market price of 7th day with previous 3 days covariates. Then I dropped all rows with "NA" entries. Finally, I generated training and testing data, such that my training data is constructed using only information up to 12/31/2016, and my testing data is constructed using only information starting from 01/01/2017

```r
library(MASS)


my_model<-lm(btc_market_price_day7~ ., train)

#Compute the studentized delted residuals
stud.del.res <- studres(my_model)

#Calculate the Bonferroni critical value for outliers:
alpha = 0.05
t <- qt(1-alpha/(2*length(stud.del.res)),
                 length(stud.del.res)-9-1)

sprintf("The critical value is t= %s", t)
```

```
## [1] "The critical value is t= 4.26866357350233"
```

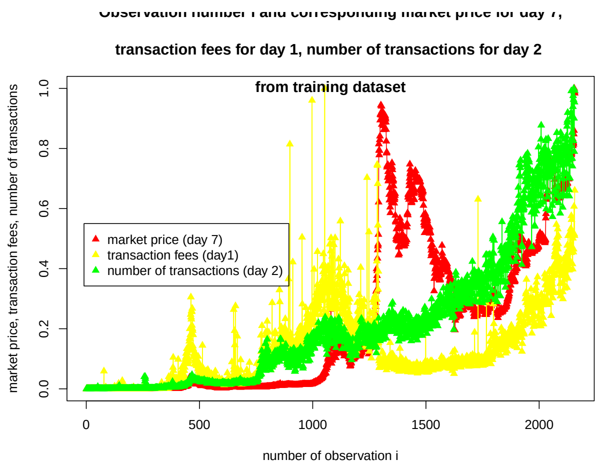
```r
paste0("Any studentized residuals > ",t," ?  ",
       any(abs(stud.del.res) > t))
```

```
## [1] "Any studentized residuals > 4.26866357350233 ?  TRUE"
```

```r
#find indexes of outliers observations
index_stud_del_res=which(abs(stud.del.res)>t)

#Compute leverages and find extremely high leverages
diagonal <- lm.influence(my_model)$hat

m=2*mean(diagonal)
paste0("Any h_ii  > ",2*mean(diagonal)," ?  ",
       any(diagonal > 2*mean(diagonal)))
```

```
## [1] "Any h_ii  > 0.055532870559412 ?  TRUE"
```

```r
#find indexes of outliers observations
index_leverage=which(diagonal > 2*mean(diagonal))

#drop these outliers observations from training data
index_outliers=unique(c(index_leverage, index_stud_del_res))
train_new<-train[-c(index_outliers),]

#Do min_max_standardization on all columns of training and testing datasets

min_max_standard <- function(x) {
    (x - min(x)) / (max(x) - min(x))
}

for(i in 1:ncol(train_new)){
  train_new[, i]<-min_max_standard(train_new[, i])
}

for(i in 1:ncol(test)){
  test[, i]<-min_max_standard(test[, i])
}

#Draw graphs

x=1:nrow(train_new)

plot(x,train_new$btc_market_price_day7, type = "l", ylim=range(c(0,1)),
     col="red", xlab = "number of observation i",
     ylab = "market price, transaction fees, number of transactions")
points(x,train_new$btc_market_price_day7, pch=17, col="red")

points(x,train_new$btc_transaction_fees_day1, pch=17, col="yellow")
lines(x,train_new$btc_transaction_fees_day1, pch=17, type = "l", col="yellow")

points(x,train_new$btc_n_transactions_day2, pch=17, col="green")
lines(x,train_new$btc_n_transactions_day2, type = "l", col="green")

title(main="Observation number i and corresponding market price for day 7,
      \ntransaction fees for day 1, number of transactions for day 2
      \nfrom training dataset", line=-1)
legend(-10, 0.55,legend=c("market price (day 7) ","transaction fees (day1)",
                          "number of transactions (day 2)"), col=c("red","yellow", "green"),
                                   pch=c(17,17), ncol=1)
```



Comment

In this part I fist found outliers in train dataset by using Studentized deleted residuals and leverages methods. Studentized deleted residuals help to detect outlying Y_i observations. Leverages help to detect outlying X_i points. By studentized deleted residual I found the critical t. By comparing absolute value of Studentized deleted residuals with the critical t, the outliers were determined. By rule of thumb any leverage $h_{ii} > 2*\bar{h}$ is flagged as having "high" leverage. By comparing value of h_{ii} with $2\bar{h}$, the outliers were determined. I decided to drop all outliers from train dataset, as outliers can affect the fitted regression prediction and can cause regression models with poor predictive properties. Then I decided to standardized all my variables in testing and training datasets, as their ranges differ a lot from each other. So I used min_max normalization, so each variable in range [0;1]. You can see it from above graphs.

b. [20 Points] Model Selection Criterion. Use AIC and BIC criteria to select the best model and report the result from each of them. Use the forward selection for AIC and backward selection for BIC. Report the following mean squared error from **both training and testing data**.

- The mean squared error: $n^{-1}\sum_i (Y_i - \widehat{Y}_i)^2$
- Since these quantities can be affected by scaling and transformation, make sure that you **state any modifications applied to the outcome variable**. Compare the training data errors and testing data errors. Which model works better? Provide a summary of your results.

```
library(lars)

fit2<-lm(btc_market_price_day7~ ., train_new)

#forward selection for AIC

step(lm(btc_market_price_day7 ~1, data=train_new),
     scope=list(upper=fit2, lower=~1), direction="forward", k=2, trace=0)
```

```
##
## Call:
## lm(formula = btc_market_price_day7 ~ btc_market_price_day3 +
##     btc_estimated_transaction_volume_usd_day2 + btc_market_cap_day1 +
##     btc_transaction_fees_day1 + btc_trade_volume_day1 + btc_hash_rate_day1 +
##     btc_cost_per_transaction_day3 + btc_estimated_transaction_volume_usd_day3 +
##     btc_n_transactions_day1 + btc_n_transactions_total_day1 +
##     btc_blocks_size_day3 + btc_total_bitcoins_day3 + btc_n_transactions_per_block_day2 +
##     btc_n_transactions_excluding_chains_longer_than_100_day1 +
##     btc_n_orphaned_blocks_day1 + btc_difficulty_day1 + btc_estimated_transaction_volume_usd_day1 +
##     btc_n_transactions_per_block_day3 + btc_hash_rate_day3 +
##     btc_n_transactions_total_day3 + btc_n_transactions_excluding_popular_day3 +
##     btc_n_unique_addresses_day1 + btc_avg_block_size_day1 + btc_output_volume_day2 +
##     btc_hash_rate_day2 + btc_trade_volume_day3, data = train_new)
##
## Coefficients:
##                                               (Intercept)
##                                                  0.002105
##                                     btc_market_price_day3
##                                                  0.994629
##                 btc_estimated_transaction_volume_usd_day2
##                                                  0.031342
##                                       btc_market_cap_day1
##                                                 -0.119565
##                                 btc_transaction_fees_day1
##                                                  0.009343
##                                     btc_trade_volume_day1
##                                                  0.021706
##                                        btc_hash_rate_day1
##                                                  0.106102
##                             btc_cost_per_transaction_day3
##                                                    0.051194
##                  btc_estimated_transaction_volume_usd_day3
##                                                    0.018570
##                                   btc_n_transactions_day1
##                                                   -0.106287
##                             btc_n_transactions_total_day1
##                                                   24.856705
##                                      btc_blocks_size_day3
##                                                   -1.119782
##                                   btc_total_bitcoins_day3
##                                                   -0.033784
##                         btc_n_transactions_per_block_day2
##                                                    0.050847
## btc_n_transactions_excluding_chains_longer_than_100_day1
##                                                    0.069269
##                                btc_n_orphaned_blocks_day1
##                                                   -0.008892
##                                     btc_difficulty_day1
##                                                   -0.191963
##                  btc_estimated_transaction_volume_usd_day1
##                                                    0.020748
##                         btc_n_transactions_per_block_day3
##                                                    0.036049
##                                      btc_hash_rate_day3
##                                                    0.059847
##                             btc_n_transactions_total_day3
##                                                  -23.736061
##                  btc_n_transactions_excluding_popular_day3
##                                                    0.049158
##                              btc_n_unique_addresses_day1
##                                                   -0.035347
##                                   btc_avg_block_size_day1
##                                                    0.033936
##                                    btc_output_volume_day2
##                                                    0.013757
##                                      btc_hash_rate_day2
##                                                    0.043071
##                                     btc_trade_volume_day3
##                                                    0.010063
```

```r
#new fit based on results of forward selection method with AIC
#Training data fit
fit_AIC_train<-lm(btc_market_price_day7 ~ btc_market_price_day3 +
    btc_estimated_transaction_volume_usd_day2 + btc_market_cap_day1 +
    btc_transaction_fees_day1 + btc_trade_volume_day1 + btc_hash_rate_day1 +
    btc_cost_per_transaction_day3 + btc_estimated_transaction_volume_usd_day3 +
    btc_n_transactions_day1 + btc_n_transactions_total_day1 +
    btc_blocks_size_day3 + btc_total_bitcoins_day3 +
      btc_n_transactions_per_block_day2 +
    btc_n_transactions_excluding_chains_longer_than_100_day1 +
    btc_n_orphaned_blocks_day1 + btc_difficulty_day1 +
      btc_estimated_transaction_volume_usd_day1 +
      btc_n_transactions_per_block_day3 + btc_hash_rate_day3 +
      btc_n_transactions_total_day3 + btc_n_transactions_excluding_popular_day3 + btc_n_unique_addresse
```

```
AIC_y_pred_train <- predict(fit_AIC_train, data=train_new)
AIC_train_MSE <- mean((train_new$btc_market_price_day7 - AIC_y_pred_train)^2)

#new fit based on results of forward selection method with AIC
#Training data fit and prediction of Y using testing data
fit_AIC_test<-lm(btc_market_price_day7 ~ btc_market_price_day3 +
    btc_estimated_transaction_volume_usd_day2 + btc_market_cap_day1 +
    btc_transaction_fees_day1 + btc_trade_volume_day1 + btc_hash_rate_day1 +
    btc_cost_per_transaction_day3 + btc_estimated_transaction_volume_usd_day3 +
    btc_n_transactions_day1 + btc_n_transactions_total_day1 +
    btc_blocks_size_day3 + btc_total_bitcoins_day3 +
      btc_n_transactions_per_block_day2 +
    btc_n_transactions_excluding_chains_longer_than_100_day1 +
    btc_n_orphaned_blocks_day1 + btc_difficulty_day1 +
      btc_estimated_transaction_volume_usd_day1
    + btc_n_transactions_per_block_day3 + btc_hash_rate_day3
    + btc_n_transactions_total_day3 + btc_n_transactions_excluding_popular_day3 + btc_n_unique_addresse

AIC_y_pred_test <- predict(fit_AIC_test, data=test)
AIC_test_MSE <- mean((test$btc_market_price_day7 - AIC_y_pred_test)^2)

print(paste0("MSE_train for forward selection method with AIC =  ", AIC_train_MSE ))
```

```
## [1] "MSE_train for forward selection method with AIC =  0.000429012858765177"
```

```
print(paste0("MSE_test for forward selection method with AIC =  ", AIC_test_MSE ))
```

```
## [1] "MSE_test for forward selection method with AIC =  0.126109005682927"
```

```
library(lars)

#backward selection for BIC
step(fit2, direction="backward", k=log(nrow(train_new)), trace=0)
```

```
##
## Call:
## lm(formula = btc_market_price_day7 ~ btc_market_price_day1 +
##     btc_trade_volume_day1 + btc_blocks_size_day1 + btc_n_orphaned_blocks_day1 +
##     btc_hash_rate_day1 + btc_difficulty_day1 + btc_n_transactions_day1 +
##     btc_n_transactions_total_day1 + btc_n_transactions_excluding_chains_longer_than_100_day1 +
##     btc_n_transactions_per_block_day2 + btc_estimated_transaction_volume_usd_day2 +
##     btc_market_price_day3 + btc_total_bitcoins_day3 + btc_cost_per_transaction_day3 +
##     btc_estimated_transaction_volume_usd_day3, data = train_new)
##
## Coefficients:
##                                             (Intercept)
##                                                0.001621
##                                   btc_market_price_day1
##                                               -0.100082
##                                   btc_trade_volume_day1
##                                                0.030569
##                                     btc_blocks_size_day1
##                                                -1.056005
##                               btc_n_orphaned_blocks_day1
##                                                -0.009676
##                                       btc_hash_rate_day1
##                                                 0.072183
##                                      btc_difficulty_day1
##                                                -0.073891
##                                  btc_n_transactions_day1
##                                                -0.090609
##                            btc_n_transactions_total_day1
##                                                 1.123604
## btc_n_transactions_excluding_chains_longer_than_100_day1
##                                                 0.082383
##                        btc_n_transactions_per_block_day2
##                                                 0.046827
##                btc_estimated_transaction_volume_usd_day2
##                                                 0.029951
##                                    btc_market_price_day3
##                                                 0.975535
##                                  btc_total_bitcoins_day3
##                                                -0.032183
##                           btc_cost_per_transaction_day3
##                                                 0.062366
##                btc_estimated_transaction_volume_usd_day3
##                                                 0.031911
```

```
#new fit based on results of backward selection method with BIC
#Training data fit
fit_BIC_train<-lm(btc_market_price_day7 ~ btc_market_price_day1 +
    btc_trade_volume_day1 + btc_blocks_size_day1 + btc_n_orphaned_blocks_day1 +
    btc_hash_rate_day1 + btc_difficulty_day1 + btc_n_transactions_day1 +
    btc_n_transactions_total_day1 +
      btc_n_transactions_excluding_chains_longer_than_100_day1 +
    btc_n_transactions_per_block_day2 +
      btc_estimated_transaction_volume_usd_day2 +
    btc_market_price_day3 + btc_total_bitcoins_day3 +
      btc_cost_per_transaction_day3 +
    btc_estimated_transaction_volume_usd_day3, data=train_new)

BIC_y_pred_train <- predict(fit_BIC_train, data=train_new)
BIC_train_MSE <- (sum((train_new$btc_market_price_day7 - BIC_y_pred_train)^2))/nrow(train_new)

#new fit based on results of backward selection method with BIC
#Testing data fit
fit_BIC_test<-lm(btc_market_price_day7 ~ btc_market_price_day1 +
    btc_trade_volume_day1 + btc_blocks_size_day1 + btc_n_orphaned_blocks_day1 +
    btc_hash_rate_day1 + btc_difficulty_day1 + btc_n_transactions_day1 +
    btc_n_transactions_total_day1 + btc_n_transactions_excluding_chains_longer_than_100_day1 +
    btc_n_transactions_per_block_day2 + btc_estimated_transaction_volume_usd_day2 +
    btc_market_price_day3 + btc_total_bitcoins_day3 + btc_cost_per_transaction_day3 +
    btc_estimated_transaction_volume_usd_day3, data=train_new)

BIC_y_pred_test <- predict(fit_BIC_test, data=test)
```

```
BIC_test_MSE <- (sum((test$btc_market_price_day7 - BIC_y_pred_test)^2))/nrow(test)

print(paste0("MSE_train for backward selection method with BIC =  ", BIC_train_MSE ))
```

```
## [1] "MSE_train for backward selection method with BIC =  0.000436177076578446"
```

```
print(paste0("MSE_test for backward selection method with BIC =  ", BIC_test_MSE ))
```

```
## [1] "MSE_test for backward selection method with BIC =  0.653851700524627"
```

Comment

I implemented forward selection with AIC and backward selection with BIC on linear model with all variables from training dataset. All my variable are min-max normalized.

From forward selection with AIC I got best model with 26 covariates (btc_market_price_day3, btc_estimated_transaction_volume_usd_day2, btc_market_cap_day1, btc_transaction_fees_day1 + btc_trade_volume_day1, btc_hash_rate_day1, btc_cost_per_transaction_day3, btc_estimated_transaction_volume_us btc_n_transactions_day1, btc_n_transactions_total_day1, btc_blocks_size_day3, btc_total_bitcoins_day3, btc_n_transactions_per_block_day2, btc_n_transactions_excluding_chains_longer_than_100_day1, btc_n_orphaned_blocks_day1, btc_difficulty_day1, btc_estimated_transaction_volume_usd_day1, btc_n_transactions_per_block_day3, btc_hash_rate_day3, btc_n_transactions_total_day3, btc_n_transactions_exclud btc_n_unique_addresses_day1 + btc_avg_block_size_day1, btc_output_volume_day2, btc_hash_rate_day2, btc_trade_volume_day3).

From backward selection with BIC I got best model with 15 covariates (btc_market_price_day1, btc_trade_volume_day1, btc_blocks_size_day1, btc_n_orphaned_blocks_day1, btc_hash_rate_day1, btc_difficulty_day1, btc_n_transactions_day1, btc_n_transactions_total_day1, btc_n_transactions_excluding_chains_l btc_n_transactions_per_block_day2, btc_estimated_transaction_volume_usd_day2, btc_market_price_day3, btc_total_bitcoins_day3, btc_cost_per_transaction_day3, btc_estimated_transaction_volume_usd_day3).

For forward selection method with AIC:

MSE_train = 0.000429012858765177. MSE_test = 0.126109005682927.

For backward selection method with BIC:

MSE_train = 0.000436177076578446. MSE_test = 0.653851700524627.

Based on results of testing MSEs, the model that we got from forward selection method with AIC will perform better as it has MSE_test = 0.126109005682927, which is smaller that MSE_test (0.653851700524627) of backward selection method with BIC. So it is better to build our model with 26 covariates.

c. [10 Points] Best Subset Selection. Fit the best subset selection to the dataset and report the best model of each model size (up to 7 variables, excluding the intercept) and their prediction errors. Make sure that you simplify your output to only present the essential information. If the algorithm cannot handle this many variables, then consider using just day 1 and 2 information. You can use `leaps` package for subset selection.

```
#install.packages("leaps")
library(leaps)

subset_sel <- regsubsets( btc_market_price_day7 ~ btc_market_price_day1+btc_total_bitcoins_day1+btc_mar
                          + btc_blocks_size_day1 + btc_avg_block_size_day1 + btc_n_orphaned_blocks_day1
```

```
btc_n_transactions_excluding_chains_longer_than_100_day1 +
btc_output_volume_day1+btc_estimated_transaction_volume_day1 + btc_estimated_transaction_volume_usd_day
  btc_market_cap_day2+btc_trade_volume_day2 + btc_blocks_size_day2 + btc_avg_block_size_day2 + btc_n_or
btc_n_transactions_excluding_chains_longer_than_100_day2 +
btc_output_volume_day2+btc_estimated_transaction_volume_day2 + btc_estimated_transaction_volume_usd_day
```

```
## Reordering variables and trying again:
```

```
all_output <- summary(subset_sel)
tail(with(all_output, round(cbind(which, rsq, adjr2, cp, bic), 3)),10)
```

```
##   (Intercept) btc_market_price_day1 btc_total_bitcoins_day1 btc_market_cap_day1
## 8           1                     1                       0                   0
## 8           1                     1                       1                   0
## 8           1                     1                       0                   0
## 9           1                     1                       0                   0
## 9           1                     1                       1                   0
## 9           1                     1                       0                   0
## 9           1                     1                       1                   0
## 9           1                     1                       0                   0
## 9           1                     1                       1                   0
## 9           1                     1                       0                   0
##   btc_trade_volume_day1 btc_blocks_size_day1 btc_avg_block_size_day1
## 8                     1                    1                       0
## 8                     1                    1                       0
## 8                     1                    1                       0
## 9                     1                    0                       0
## 9                     1                    0                       0
## 9                     1                    1                       0
## 9                     1                    1                       0
## 9                     1                    0                       0
## 9                     1                    0                       0
## 9                     1                    1                       0
##   btc_n_orphaned_blocks_day1 btc_n_transactions_per_block_day1
## 8                          0                                 0
## 8                          0                                 0
## 8                          0                                 0
## 9                          0                                 0
## 9                          0                                 0
## 9                          0                                 0
## 9                          0                                 0
## 9                          0                                 0
## 9                          0                                 0
## 9                          0                                 0
##   btc_median_confirmation_time_day1 btc_hash_rate_day1 btc_difficulty_day1
## 8                                 0                  0                   0
## 8                                 0                  0                   0
## 8                                 0                  0                   0
## 9                                 0                  1                   0
## 9                                 0                  1                   0
## 9                                 0                  1                   0
## 9                                 0                  1                   0
## 9                               0               1                  0
## 9                               0               1                  0
## 9                               0               1                  0
##   btc_miners_revenue_day1 btc_transaction_fees_day1
## 8                       0                         0
## 8                       0                         0
## 8                       0                         0
## 9                       0                         0
## 9                       0                         0
## 9                       0                         0
## 9                       0                         0
## 9                       0                         0
## 9                       0                         0
## 9                       0                         0
##   `btc_cost_per_transaction_percent _day1` btc_cost_per_transaction_day1
## 8                                        0                             0
## 8                                        0                             0
## 8                                        0                             0
## 9                                        0                             0
## 9                                        0                             0
## 9                                        0                             0
## 9                                        0                             0
## 9                                        0                             0
## 9                                        0                             0
## 9                                        0                             0
##   btc_n_unique_addresses_day1 btc_n_transactions_day1 btc_n_transactions_total_day1
## 8                           0                       0                             1
## 8                           0                       0                             1
## 8                           0                       0                             0
## 9                           0                       0                             1
## 9                           0                       0                             1
## 9                           0                       0                             1
## 9                           0                       0                             1
## 9                           0                       0                             0
## 9                           0                       0                             0
## 9                           0                       0                             0
##   btc_n_transactions_excluding_popular_day1
## 8                                         0
## 8                                         0
## 8                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
##   btc_n_transactions_excluding_chains_longer_than_100_day1 btc_output_volume_day1
## 8                                                        0                      0
## 8                                                        0                      0
## 8                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                      0                    0
## 9                                      0                    0
## 9                                      0                    0
## 9                                      0                    0
##   btc_estimated_transaction_volume_day1 btc_estimated_transaction_volume_usd_day1
## 8                                     0                                         0
## 8                                     0                                         0
## 8                                     0                                         0
## 9                                     0                                         0
## 9                                     0                                         0
## 9                                     0                                         0
## 9                                     0                                         0
## 9                                     0                                         0
## 9                                     0                                         0
## 9                                     0                                         0
##   btc_market_price_day2 btc_total_bitcoins_day2 btc_market_cap_day2 btc_trade_volume_day2
## 8                     1                       1                   0                     0
## 8                     1                       0                   0                     0
## 8                     1                       1                   0                     0
## 9                     1                       1                   0                     0
## 9                     1                       0                   0                     0
## 9                     1                       1                   0                     0
## 9                     1                       0                   0                     0
## 9                     1                       1                   0                     0
## 9                     1                       0                   0                     0
## 9                     1                       1                   0                     0
##   btc_blocks_size_day2 btc_avg_block_size_day2 btc_n_orphaned_blocks_day2
## 8                    0                       0                          0
## 8                    0                       0                          0
## 8                    0                       0                          0
## 9                    1                       0                          0
## 9                    1                       0                          0
## 9                    0                       0                          0
## 9                    0                       0                          0
## 9                    1                       0                          0
## 9                    1                       0                          0
## 9                    0                       0                          0
##   btc_n_transactions_per_block_day2 btc_median_confirmation_time_day2 btc_hash_rate_day2
## 8                                 0                                 0                  0
## 8                                 0                                 0                  0
## 8                                 0                                 0                  0
## 9                                 0                                 0                  0
## 9                                 0                                 0                  0
## 9                                 0                                 0                  0
## 9                                 0                                 0                  0
## 9                                 0                                 0                  0
## 9                                 0                                 0                  0
## 9                                 0                                 0                  0
##   btc_difficulty_day2 btc_miners_revenue_day2 btc_transaction_fees_day2
## 8                   0                       0                         0
## 8                   0                       0                         0
## 8                   0                       0                         0
## 9                   0                       0                         0
## 9                   0                       0                         0
## 9                  0                        0                          0
## 9                  0                        0                          0
## 9                  0                        0                          0
## 9                  0                        0                          0
## 9                  0                        0                          0
##   `btc_cost_per_transaction_percent _day2` btc_cost_per_transaction_day2
## 8                                        0                             1
## 8                                        0                             1
## 8                                        0                             1
## 9                                        0                             1
## 9                                        0                             1
## 9                                        0                             1
## 9                                        0                             1
## 9                                        0                             1
## 9                                        0                             1
## 9                                        0                             1
##   btc_n_unique_addresses_day2 btc_n_transactions_day2 btc_n_transactions_total_day2
## 8                           0                       0                             0
## 8                           0                       0                             0
## 8                           0                       0                             1
## 9                           0                       0                             0
## 9                           0                       0                             0
## 9                           0                       0                             0
## 9                           0                       0                             0
## 9                           0                       0                             1
## 9                           0                       0                             1
## 9                           0                       0                             1
##   btc_n_transactions_excluding_popular_day2
## 8                                         0
## 8                                         0
## 8                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
## 9                                         0
##   btc_n_transactions_excluding_chains_longer_than_100_day2 btc_output_volume_day2
## 8                                                        0                      0
## 8                                                        0                      0
## 8                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
## 9                                                        0                      0
##   btc_estimated_transaction_volume_day2 btc_estimated_transaction_volume_usd_day2   rsq
## 8                                     0                                         1 0.992
## 8                                     0                                         1 0.992
## 8                                     0                                         1 0.992
## 9                                     0                                         1 0.992
## 9                                    0                                         1 0.992
## 9                                    0                                         1 0.992
## 9                                    0                                         1 0.992
## 9                                    0                                         1 0.992
## 9                                    0                                         1 0.992
## 9                                    0                                         1 0.992
##   adjr2     cp       bic
## 8 0.991 66.264 -10218.02
## 8 0.991 66.264 -10218.02
## 8 0.991 66.564 -10217.72
## 9 0.992 56.138 -10222.23
## 9 0.992 56.144 -10222.23
## 9 0.992 56.477 -10221.90
## 9 0.992 56.483 -10221.89
## 9 0.992 56.577 -10221.80
## 9 0.992 56.584 -10221.80
## 9 0.992 56.952 -10221.43
```

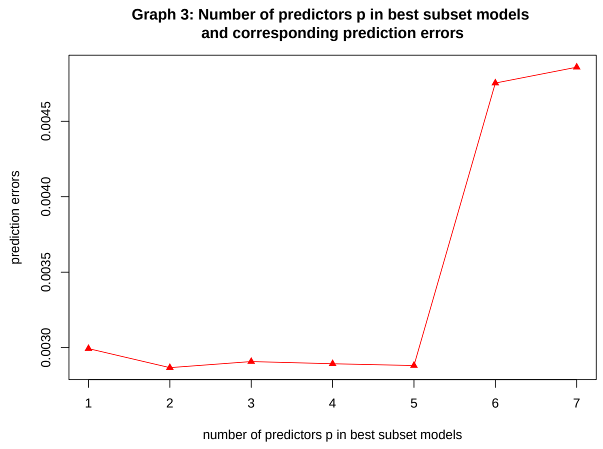
```
table_results<-data.frame(with(all_output, round(cbind(which, rsq, adjr2, cp, bic), 3)))


#1 predictor: R^2=0.99, Cp=317.598, BIC=-10026.688

model_p1<-lm(btc_market_price_day7~btc_market_price_day2,train_new)

#2 predictors: R^2=0.991, Cp=188.574, BIC=-10136.5
model_p2<-lm(btc_market_price_day7 ~ btc_market_price_day2 + btc_estimated_transaction_volume_usd_day2,

#3 predictors: R^2=0.991, Cp=168.290, BIC=-10149.52

model_p3<-lm(btc_market_price_day7~btc_market_price_day2+btc_transaction_fees_day1+ btc_estimated_transa


#4 predictors: R^2=0.991, Cp=144.234, BIC=-10166.23

model_p4<-lm(btc_market_price_day7~btc_trade_volume_day1+btc_market_price_day2 + btc_transaction_fees_da


#5 predictors: R^2=0.991, Cp=130.618, BIC=-10173.31

model_p5<-lm(btc_market_price_day7 ~ btc_trade_volume_day1 + btc_market_price_day2 + btc_transaction_fee
     btc_cost_per_transaction_day2+btc_estimated_transaction_volume_usd_day2,train_new)

#6 predictors: R^2=0.991, Cp=93.556, BIC=-10202.98

model_p6<-lm(btc_market_price_day7~ btc_n_transactions_total_day1 + btc_market_price_day2 + btc_total_bi

#7 predictors: R^2=0.991, Cp=77.702, BIC=-10212.59

model_p7<-lm(btc_market_price_day7~ btc_trade_volume_day1 + btc_n_transactions_total_day1 + btc_market_p

best_subsets <- data.frame(matrix(ncol = 51, nrow = 0))

k=1
```

```r
for( i in 1:7){
  best_subsets[i, ] = table_results[k, ]
  k=k+7
}
names(best_subsets)<-names(table_results)

best_subsets
```

```
##    X.Intercept. btc_market_price_day1 btc_total_bitcoins_day1 btc_market_cap_day1
## X1            1                     0                       0                   0
## X2            1                     0                       0                   0
## X3            1                     0                       0                   0
## X4            1                     0                       0                   0
## X5            1                     0                       0                   0
## X6            1                     0                       0                   0
## X7            1                     0                       0                   0
##    btc_trade_volume_day1 btc_blocks_size_day1 btc_avg_block_size_day1
## X1                     0                    0                       0
## X2                     0                    0                       0
## X3                     0                    0                       0
## X4                     1                    0                       0
## X5                     1                    0                       0
## X6                     0                    0                       0
## X7                     1                    0                       0
##    btc_n_orphaned_blocks_day1 btc_n_transactions_per_block_day1
## X1                          0                                 0
## X2                          0                                 0
## X3                          0                                 0
## X4                          0                                 0
## X5                          0                                 0
## X6                          0                                 0
## X7                          0                                 0
##    btc_median_confirmation_time_day1 btc_hash_rate_day1 btc_difficulty_day1
## X1                                 0                  0                   0
## X2                                 0                  0                   0
## X3                                 0                  0                   0
## X4                                 0                  0                   0
## X5                                 0                  0                   0
## X6                                 0                  0                   0
## X7                                 0                  0                   0
##    btc_miners_revenue_day1 btc_transaction_fees_day1
## X1                       0                         0
## X2                       0                         0
## X3                       0                         1
## X4                       0                         0
## X5                       0                         0
## X6                       0                         0
## X7                       0                         0
##    X.btc_cost_per_transaction_percent._day1. btc_cost_per_transaction_day1
## X1                                         0                             0
## X2                                         0                             0
## X3                                         0                             0
## X4                                         0                             0
## X5                                        0                             0
## X6                                        0                             0
## X7                                        0                             0
##    btc_n_unique_addresses_day1 btc_n_transactions_day1 btc_n_transactions_total_day1
## X1                           0                       0                             0
## X2                           0                       0                             0
## X3                           0                       0                             0
## X4                           0                       0                             0
## X5                           0                       0                             0
## X6                           0                       0                             1
## X7                           0                       0                             1
##    btc_n_transactions_excluding_popular_day1
## X1                                         0
## X2                                         0
## X3                                         0
## X4                                         0
## X5                                         0
## X6                                         0
## X7                                         0
##    btc_n_transactions_excluding_chains_longer_than_100_day1 btc_output_volume_day1
## X1                                                        0                      0
## X2                                                        0                      0
## X3                                                        0                      0
## X4                                                        0                      0
## X5                                                        0                      0
## X6                                                        0                      0
## X7                                                        0                      0
##    btc_estimated_transaction_volume_day1 btc_estimated_transaction_volume_usd_day1
## X1                                     0                                         0
## X2                                     0                                         0
## X3                                     0                                         0
## X4                                     0                                         0
## X5                                     0                                         0
## X6                                     0                                         0
## X7                                     0                                         0
##    btc_market_price_day2 btc_total_bitcoins_day2 btc_market_cap_day2
## X1                     1                       0                   0
## X2                     1                       0                   0
## X3                     1                       0                   0
## X4                     1                       0                   0
## X5                     1                       0                   0
## X6                     1                       1                   0
## X7                     1                       1                   0
##    btc_trade_volume_day2 btc_blocks_size_day2 btc_avg_block_size_day2
## X1                     0                    0                       0
## X2                     0                    0                       0
## X3                     0                    0                       0
## X4                     0                    0                       0
## X5                     0                    0                       0
## X6                     0                    1                       0
## X7                     0                    1                       0
##    btc_n_orphaned_blocks_day2 btc_n_transactions_per_block_day2
## X1                          0                                 0
## X2                          0                                 0
## X3                          0                                0
## X4                          0                                0
## X5                          0                                0
## X6                          0                                0
## X7                          0                                0
##    btc_median_confirmation_time_day2 btc_hash_rate_day2 btc_difficulty_day2
## X1                                 0                  0                   0
## X2                                 0                  0                   0
## X3                                 0                  0                   0
## X4                                 0                  0                   0
## X5                                 0                  0                   0
## X6                                 0                  0                   0
## X7                                 0                  0                   0
##    btc_miners_revenue_day2 btc_transaction_fees_day2
## X1                       0                         0
## X2                       0                         0
## X3                       0                         0
## X4                       0                         1
## X5                       0                         1
## X6                       0                         0
## X7                       0                         0
##    X.btc_cost_per_transaction_percent._day2. btc_cost_per_transaction_day2
## X1                                         0                             0
## X2                                         0                             0
## X3                                         0                             0
## X4                                         0                             0
## X5                                         0                             1
## X6                                         0                             1
## X7                                         0                             1
##    btc_n_unique_addresses_day2 btc_n_transactions_day2 btc_n_transactions_total_day2
## X1                           0                       0                             0
## X2                           0                       0                             0
## X3                           0                       0                             0
## X4                           0                       0                             0
## X5                           0                       0                             0
## X6                           0                       0                             0
## X7                           0                       0                             0
##    btc_n_transactions_excluding_popular_day2
## X1                                         0
## X2                                         0
## X3                                         0
## X4                                         0
## X5                                         0
## X6                                         0
## X7                                         0
##    btc_n_transactions_excluding_chains_longer_than_100_day2 btc_output_volume_day2
## X1                                                        0                      0
## X2                                                        0                      0
## X3                                                        0                      0
## X4                                                        0                      0
## X5                                                        0                      0
## X6                                                        0                      0
## X7                                                        0                      0
##    btc_estimated_transaction_volume_day2 btc_estimated_transaction_volume_usd_day2   rsq
## X1                                     0                                        0 0.990
## X2                                     0                                        1 0.991
## X3                                     0                                        1 0.991
## X4                                     0                                        1 0.991
## X5                                     0                                        1 0.991
## X6                                     0                                        1 0.991
## X7                                     0                                        1 0.991
##    adjr2      cp       bic
## X1 0.990 317.598 -10026.69
## X2 0.991 188.574 -10136.55
## X3 0.991 168.290 -10149.52
## X4 0.991 144.234 -10166.23
## X5 0.991 130.618 -10173.31
## X6 0.991  93.556 -10202.98
## X7 0.991  77.702 -10212.59
```

```
#Calculating predicted error
pred_err_best_subsets=rep(0,7)

#Function to calculate predicted error
pred_err<-function(Y, Y_pred){
  mean((Y_pred-Y)^2)
}

#predict Y using test data and calculate MSE for models with p=1,2,3,...7,
#where p is number of covariates

 Ypred1=predict(model_p1, test)
 pred_err_best_subsets[1] = pred_err(test$btc_market_price_day7, Ypred1 )

Ypred2=predict(model_p2, test)
pred_err_best_subsets[2] = pred_err(test$btc_market_price_day7, Ypred2)

Ypred3=predict(model_p3, test)
pred_err_best_subsets[3] = pred_err(test$btc_market_price_day7, Ypred3)

Ypred4=predict(model_p4, test)
pred_err_best_subsets[4] = pred_err(test$btc_market_price_day7, Ypred4)

Ypred5=predict(model_p5, test)
pred_err_best_subsets[5] = pred_err(test$btc_market_price_day7, Ypred5)

Ypred6=predict(model_p6, test)
pred_err_best_subsets[6] = pred_err(test$btc_market_price_day7, Ypred6)

Ypred7=predict(model_p7, test)
pred_err_best_subsets[7] = pred_err(test$btc_market_price_day7, Ypred7)


#Draw graphs

x=1:7
#ylim=range(c(0,2.4))
```

```
plot(x,pred_err_best_subsets, type = "l", col="red", xlab = "number of predictors p in best subset mode
points(x,pred_err_best_subsets, pch=17, col="red")

title(main="Graph 3: Number of predictors p in best subset models \nand corresponding prediction errors
```



Comment

I conducted best subset selection using all covariates for day 1 , day 2 and the response variable market (price for day 7). By selection model criterion the best model has the lowest BIC, lowest C_p and the highest adjusted R^2. So I chose from 7 subsets for number of predictors p where p=1,2,. . . ,7 the best model, build this model and calculated MSE. All 7 best models were summarized in "best_subsets" table.

When p=1, the best model is btc_market_price_day7 ~ btc_market_price_day2

When p=2, the best model is btc_market_price_day7 ~ btc_market_price_day2 + btc_estimated_transaction_volume_us

When p=3, the best model is btc_market_price_day7~btc_market_price_day2+ btc_transaction_fees_day1 + btc_estimated_transaction_volume_usd_day2

When p=4, the best model is btc_market_price_day7~btc_trade_volume_day1+ btc_market_price_day2 + btc_transaction_fees_day2+btc_estimated_transaction_volume_usd_day2

When p=5, the best model is btc_market_price_day7 ~ btc_trade_volume_day1 + btc_market_price_day2 + btc_transaction_fees_day2 + btc_cost_per_transaction_day2 + btc_estimated_transaction_volume_usd_day2

When p=6, the best model is btc_market_price_day7~ btc_n_transactions_total_day1 + btc_market_price_day2 + btc_total_bitcoins_day2 + btc_blocks_size_day2 + btc_cost_per_transaction_day2 + btc_estimated_transaction_vol

When p=7, the best model is btc_market_price_day7~ btc_trade_volume_day1 + btc_n_transactions_total_day1 + btc_market_price_day2 + btc_total_bitcoins_day2 + btc_blocks_size_day2 + btc_cost_per_transaction_day2 + btc_estimated_transaction_volume_usd_day2

Also, I constructed a graph number of predictors p in best subset models vs their corresponding prediction error.