Andrew Rothman
anr248@stanford.edu

November 12th, 2018
SUID: 06325118

Problem #1:

Exercise 1 from section 6.8 of ITSL textbook

We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p + 1$ models, containing $0,1,2, \dots, p$ predictors. Explain your answers:

(a) Which of the three models with k predictors has the smallest training RSS?

Answer:

"Best subset" selection will by definition provide the same or smallest training RSS with k predictors as compared to forward and backward stepwise selection.

Given it's the case we are assessing fit vis RSS, let us assume we are running linear regression. In the "Best subset" selection approach, we enumerate all possible models with k predictors, store the k-predictor model with the lowest RSS, and repeat the same procedure for all k (i.e. $k = 0,1,2, \dots, p$). We can then use an appropriate fit metric for assessing models with different numbers of predictors (i.e. C_p, BIC, adjusted R^2, cross validation error, etc) to choose the "best" model among the p+1 models.

Let us concentrate for a moment on a specific "k". For a given "k", "Best subset" selection will choose the model with the lowest training RSS among the $\binom{p}{k}$ combinations of the p predictors. This can be computationally expensive, given the number of models that needed to be assessed. Both forward and backward stepwise selection are greedy approaches meant to approximate a solution to the "best subset" problem in a computationally efficient manner by searching over a more restricted number of potential models. For instance, in forward stepwise, when assessing possible models with "k" predictors, we exclusively start with the "best" model using "k-1" predictors. Meaning all of the predictors contained in the best "k-1" model will be forced into the possible "k" models. The same can be said for backward stepwise selection, but in reverse (i.e. the best "k" model will start with the best "k+1" model). This is where forward and backward stepwise methods differ from best subset selection. If best subset selection for the "k" in question happens to have more than one predictor in it that is not contained in the "best" forward stepwise model with "k-1" predictors, or the "best" backward stepwise model with "k+1" predictors, than the best "k" model for best subset will differ from the best "k" model for forward and/or backward stepwise and will have a lower RSS.

(b) Which of the three models with k predictors has the smallest test RSS?

Answer:

If testing the accuracy on a hold-out test set to assess the test RSS, any of the three approaches (best subset, forward selection, or backward selection) may produce the smallest test RSS with k predictors. It is not possible to verify before-hand deterministically which model would perform best, given the testing data were not used to recover parameters estimates for these models.

(c) True or False:

i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.

Answer:

True. Forward stepwise selection is a greedy approach meant to somewhat approximate the solution to the "best subset" problem in a computationally efficient manner by searching over a more restricted number of possible models. By definition, forward stepwise with "k+1" predictors will start with the "best" model with "k" predictors, and force those "k" variables into the model for the possible "k+1" models. This therefore means all of the predictors in the k-variable model will be a subset of the predictors in the (k+1)-variable model.

ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.

Answer:

True. Backward stepwise selection is a greedy approach meant to somewhat approximate the solution to the "best subset" problem in a computationally efficient manner by searching over a more restricted number of possible models. By definition, backward stepwise with "k" predictors will start with the "best" model with "k+1" predictors, and remove the one variable from the model the lowers the training error most. This therefore means all of the predictors in the k-variable model will be a subset of the predictors in the (k+1)-variable model.

iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.

Answer:

False. I have described the logic of both forward and backward selection in parts (i) and (ii) above. By definition, the k-variable model for forward stepwise is chosen starting with the "best" model from the "k-1" forward stepwise results. The k-variable model for backward stepwise is chosen starting with the "best" model from the "k+1" backward stepwise results. It is possible for the "k-1" model from forward stepwise and the "k+1" model from backward stepwise to differ in more than two variables. If they do differ in more than two variables, then by definition the variables in the k-variable model identified by backward stepwise cannot all be a subset of the (k+1)-variable model identified by forward stepwise.

iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.

Answer:

False. I have described the logic of both forward and backward selection in parts (i) and (ii) above. By definition, the k-variable model for forward stepwise is chosen starting with the "best" model from the "k-1" forward stepwise results. The k-variable model for backward stepwise is chosen starting with the "best" model from the "k+1" backward stepwise results. It is possible for the "k-1" model from forward stepwise and the "k+1" model from backward stepwise to differ in more than two variables. If they do differ in more than two variables, then by definition the variables in the k-variable model identified by forward stepwise cannot all be a subset of the (k+1)-variable model identified by backward stepwise.

v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k+1)-variable model identified by best subset selection.

Answer:

False. In best subset selection, for each value of "k" we choose the model the has the lowest training error among the $\binom{p}{k}$ combinations of the p predictors. All of the variables contained in the k-variable model identified by best subset will not necessarily also be contained in the (k+1)-variable model identified by best subset.

Problem #2:

Modified from exercise 2, section 6.8 of ITSL textbook

For parts (a) and (b), indicate which choice is correct. Justify your answer:

(a) The lasso, relative to least squares, is:

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

The LASSO coefficient estimation $\hat{\beta}^{LASSO}$ are values that minimize:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

When $\lambda = 0$, the objective function above reduces to the RSS, providing an unbiased estimate of the RSS. As λ increases from 0, the flexibility of the LASSO decreases, leading to increase in bias but decrease in variance in the testing error. If the increase in bias in the testing error is less than the decrease in variance, than we will get improved prediction accuracy with LASSO regression as compared to least squares.

(b) The ridge regression, relative to least squares, is:

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

The ridge regression coefficient estimation $\hat{\beta}^R$ are values that minimize:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2 = RSS + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

When $\lambda = 0$, the objective function above reduces to the RSS, providing an unbiased estimate of the RSS. As λ increases from 0, the flexibility of the ridge regression decreases, leading to increase in bias but decrease in variance in the testing error. If the increase in bias in the testing error is less than the decrease in variance, than we will get improved prediction accuracy with ridge regression as compared to least squares.

Problem #3:

Exercise 4 from section 6.8 of ITSL textbook

Suppose we estimate the regression coefficients in a linear regression model by minimizing:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

for a particular value of λ. For parts (a) through (e), indicate which choice is correct. Justify your answer:

Answers for several of the questions below can be aided by referring to the following plot in the ITSL textbook:

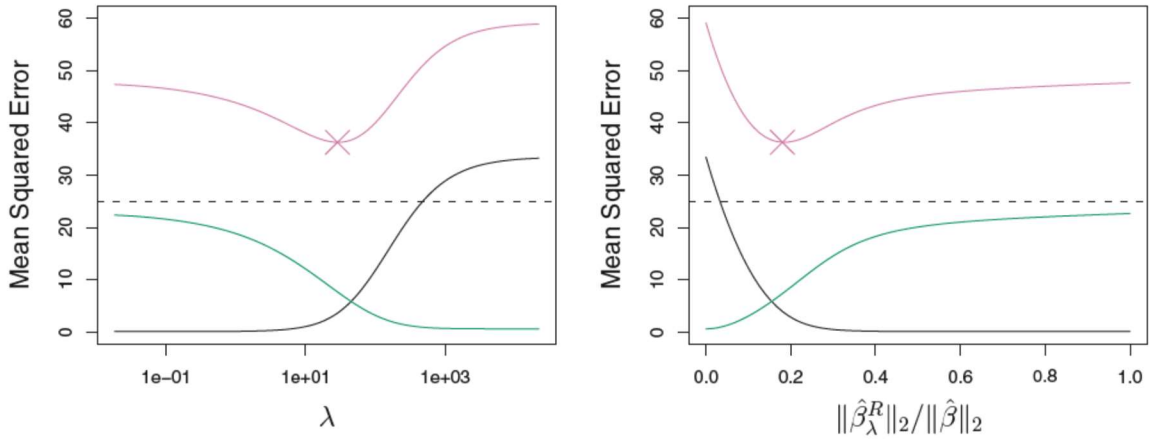


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

(a) As we increase λ from 0, the training RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

In the objective function below:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

the term

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

is equal to the training RSS for ordinary least squares. Therefore we can see that as we increase λ from 0, the whole objective function above must increase due to the term "$+\lambda\sum_{j=1}^{p}{\beta_j}^2$"

This will mean the training RSS steadily increases.

(b) As we increase λ from 0, the test RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

Referring to the left graph in Figure 6.5 above, we can see the purple line refers to the test MSE as a function of increasing λ from 0. Given the MSE is a scaled version of the RSS ($MSE = \frac{1}{n} * RSS$), the test RSS will have a similar shape and trend to this purple line. As we can see, the line decreases initially, and then starts increasing in a U shape as a function of λ from 0. This is because at first, the reduction in variance from the regularization term is outweighing the increase in bias. However, eventually with λ sufficiently large, the increase in bias by the regularization term outweighs the decrease in variance, and the test RSS increases.

(c) As we increase λ from 0, the variance will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

Referring to the left graph in Figure 6.5 above, we can see the green line refers to the variance as a function of increasing λ from 0. As we can see, the line steadily decreases as a function of increasing λ from 0. This is because as the regularization term has more prominence in the objective function with increasing λ, the shrinkage of the coefficient estimates causes the variance to continue to decrease.

(d) As we increase λ from 0, the squared bias will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

Referring to the left graph in Figure 6.5 above, we can see the black line refers to the squared bias as a function of increasing λ from 0. As we can see, the line steadily increases as a function of increasing λ from 0. This is because as the regularization term has more prominence in the objective function with increasing λ, the shrinkage of the coefficient estimates causes continued increase in the squared bias.

(e) As we increase λ from 0, the irreducible error will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

Referring to the left graph in Figure 6.5 above, we can see the dashed line refers to the irreducible error as a function of increasing λ from 0. As we can see, this line is flat and remains constant. By definition, the irreducible error is the smallest possible testing error attainable, and is error that by definition cannot be explained or accounted for by the model.

Problem #4:

Modified from exercise 8, section 6.8 of ITSL textbook

In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Use the "rt()" function with 15 degrees of freedom to generate a predictor X of length n=100, as well as a noise vector ϵ of length n=100.

Answer:

Vectors $\boldsymbol{X}$ and $\boldsymbol{\epsilon}$ were generated accordingly in R. Summary statistics of these vectors are shown below:

	Size "n"	Mean	Median	Standard Deviation
$\boldsymbol{X}$	100	0.12	0.1	1.13
$\boldsymbol{\epsilon}$	100	0.09	0.01	1.17

(b) Generate a response vector Y of length n=100 according to the model:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

Where $\beta_0, \beta_1, \beta_2, \beta_3$ are all equal to 0.5

Answer:

Vector $\boldsymbol{Y}$ was generated accordingly in R. Summary statistics of the vector is shown below:

	Size "n"	Mean	Median	Standard Deviation
$\boldsymbol{Y}$	100	2.29	2.03	1.79

(c) Use the "regsubsets()" function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \dots, X^{10}$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained.

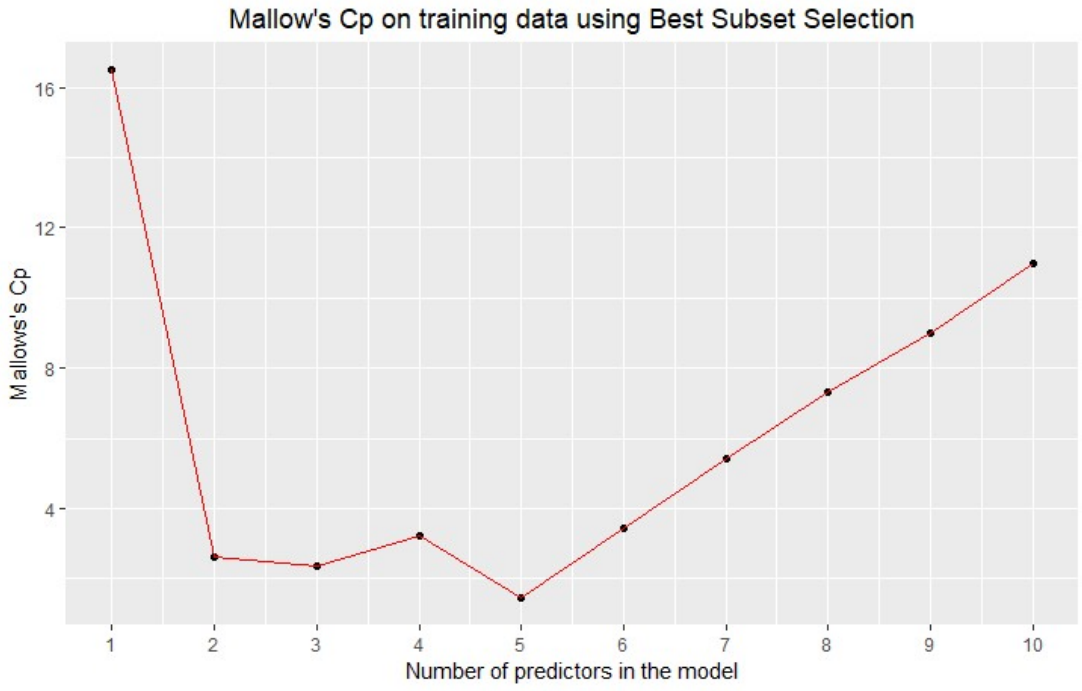
Answer:

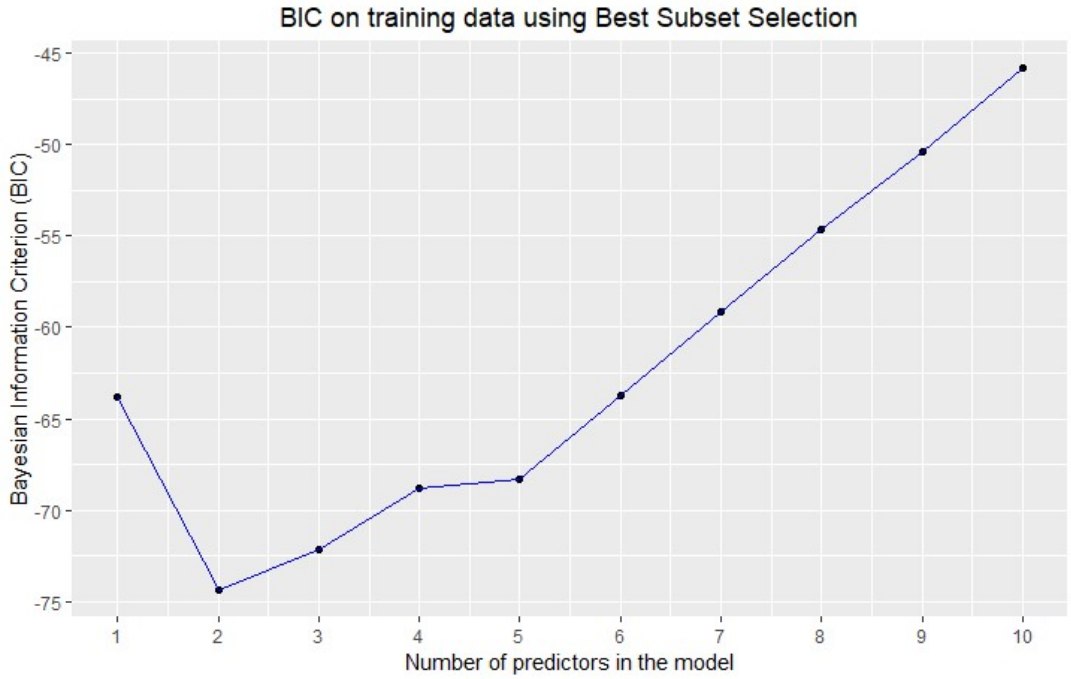
Using best subset selection, the "k" variables in each of the "best" models for each value of "k" from 1-10 is shown below:

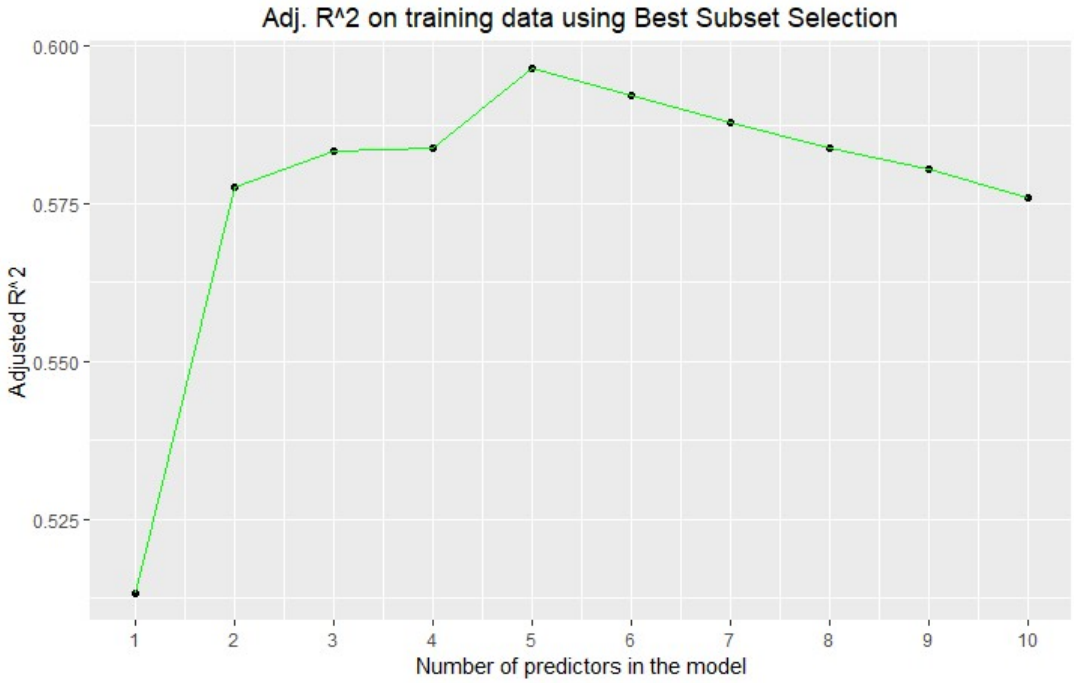
Results of Selection Algorithm using Best Subset Selection

Number of Variables included in the model	X	X^2	X^3	X^4	X^5	X^6	X^7	X^8	X^9	X^{10}
1		*								
2	*	*								
3	*	*								*
4	*	*						*		*
5	*	*		*		*		*		
6	*	*		*		*		*		*
7	*	*		*		*		*	*	*
8	*	*	*	*	*	*	*	*		
9		*	*	*	*	*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*

For the "best" model from best subset selection for each value of k, the Mallow's C_p, BIC, and adjusted R^2 were recorded for each value of k from 1-10 and plotted below:




BIC on training data using Best Subset Selection
Bayesian Information Criterion (BIC)
-45
-50
-55
-60
-65
-70
-75
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model


Adj. R^2 on training data using Best Subset Selection
Adjusted R^2
0.600
0.575
0.550
0.525
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model

From the plots above, we can see there is discrepancy as to the "best" model of the 10 possible models (one model for each k from 1-10) depending on the fit metric that is used.

- As per the BIC, the model with the lowest BIC (-74.40606) is the two-variable model (with predictors X, X^2).
- As per Mallow's C_p and adjusted R^2, the model with the lowest C_p (1.431442) and highest adjusted R^2 (0.5965466) is the five-variable model (with predictors X, X^2, X^4, X^6, X^8).

The fitted two-variable and five-variable models are shown below:

Best model from best subset selection as per BIC:

$$\hat{Y} = 1.4899 + 0.4339X + 0.5858X^2$$

Best model from best subset selection as per C_p and adjusted R^2:

$$\hat{Y} = 1.244454 + 0.420541X + 1.677202X^2 - 0.639485X^4 + 0.114449X^6 - 0.006104X^8$$

(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results to (c)?

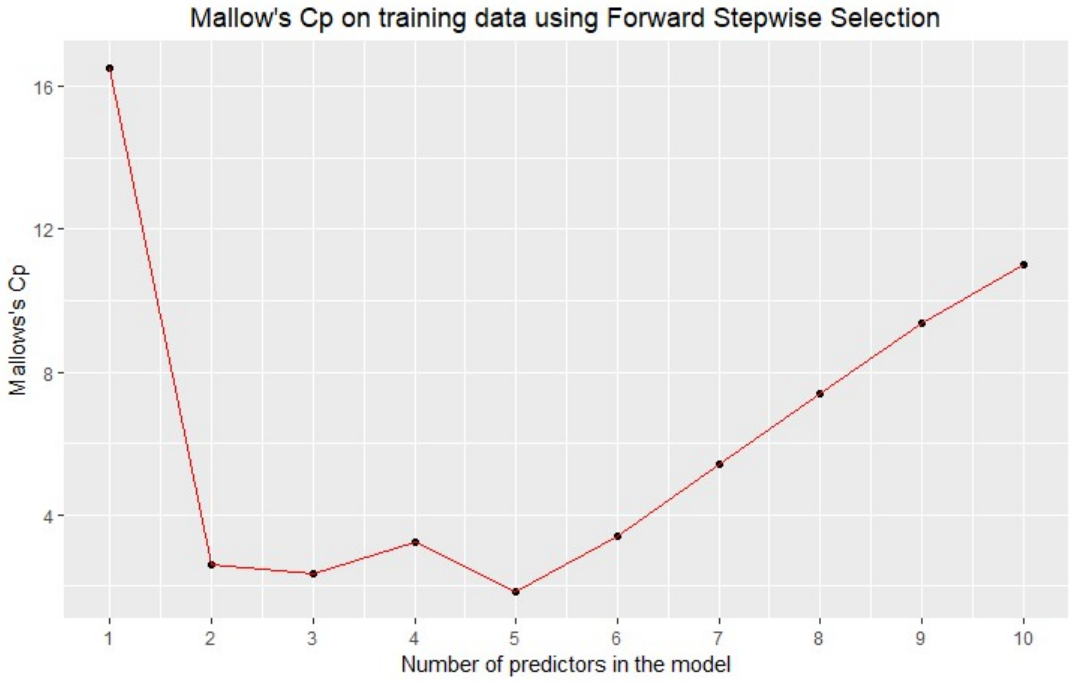
Answer:

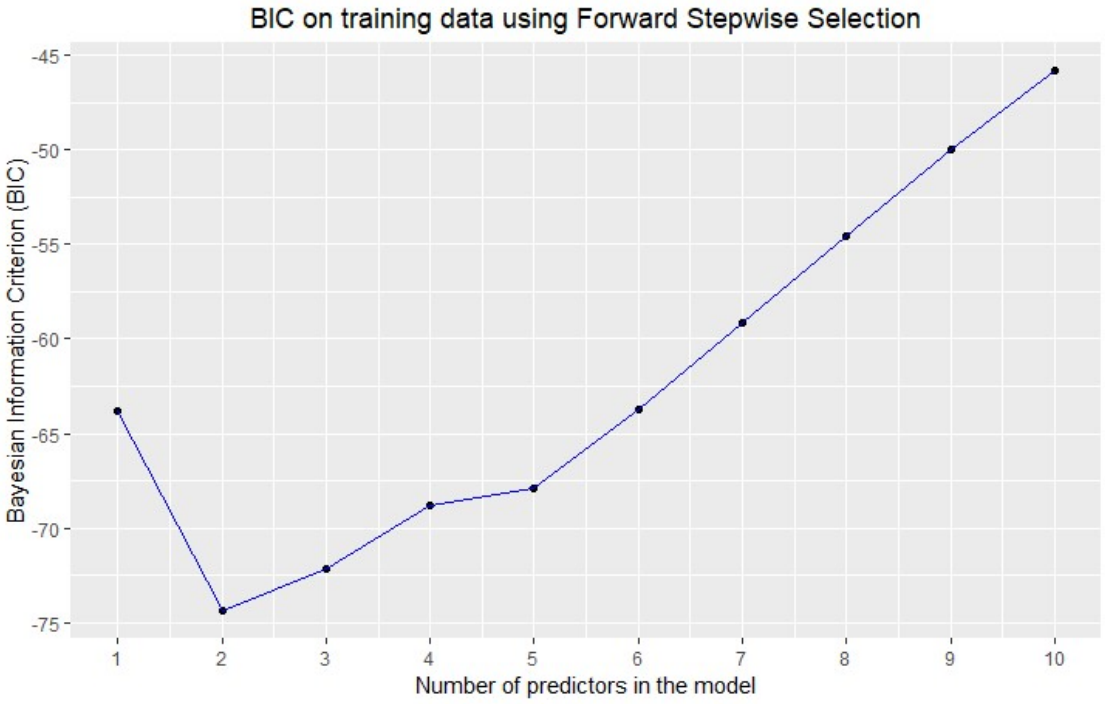
Using forward stepwise selection, the "k" variables in each of the "best" models for each value of "k" from 1-10 is shown below:

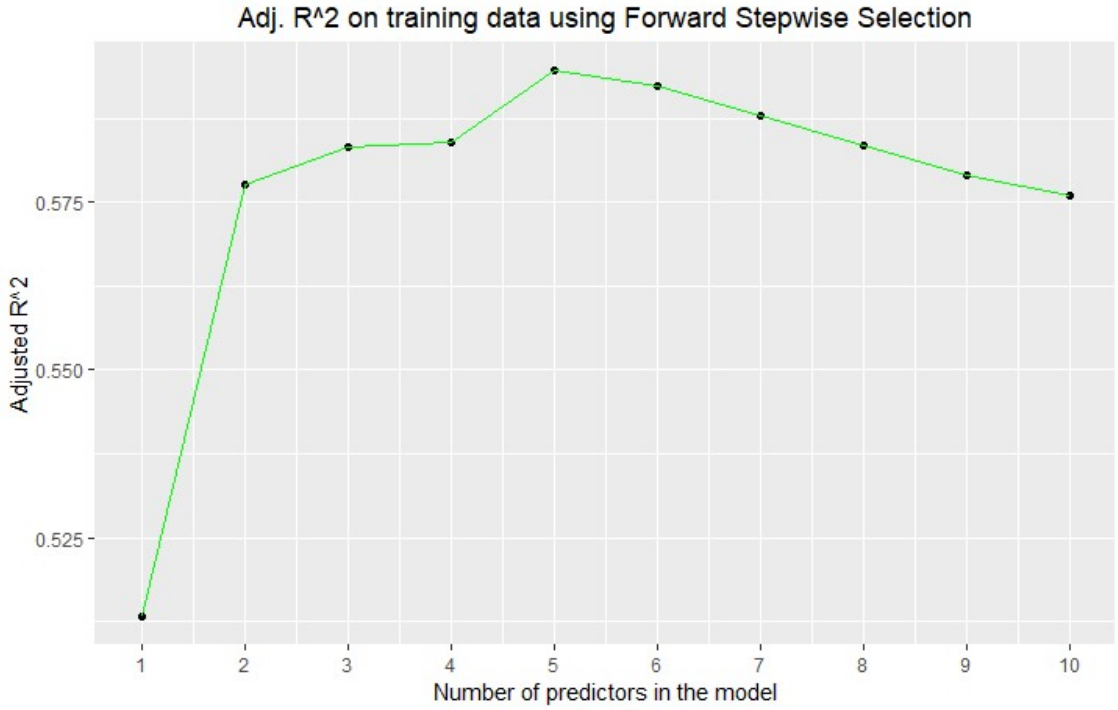
Results of Selection Algorithm using Forward Stepwise Selection

Number of Variables included in the model	X	X^2	X^3	X^4	X^5	X^6	X^7	X^8	X^9	X^{10}
1		*								
2	*	*								
3	*	*								*
4	*	*						*		*
5	*	*		*				*		*
6	*	*		*		*		*		*
7	*	*		*		*		*	*	*
8	*	*		*		*	*	*	*	*
9	*	*	*	*		*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*

For the "best" model from forward stepwise selection for each value of k, the Mallow's C_p, BIC, and adjusted R^2 were recorded for each value of k from 1-10 and plotted below:




BIC on training data using Forward Stepwise Selection
Bayesian Information Criterion (BIC)
-45
-50
-55
-60
-65
-70
-75
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model


Adj. R^2 on training data using Forward Stepwise Selection
Adjusted R^2
0.575
0.550
0.525
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model

From the plots above, we can see there is discrepancy as to the "best" model of the 10 possible models (one model for each k from 1-10) depending on the fit metric that is used.

- As per the BIC, the model with the lowest BIC (-74.40606) is the two-variable model (with predictors X, X^2).
- As per Mallow's C_p and adjusted R^2, the model with the lowest C_p (1.829393) and highest adjusted R^2 (0.5947513) is the five-variable model (with predictors X, X^2, X^4, X^8, X^{10}).

The fitted two-variable and five-variable models are shown below:

Best model from forward stepwise selection as per BIC:

$$\hat{Y} = 1.4899 + 0.4339X + 0.5858X^2$$

Best model from best subset selection as per C_p and adjusted R^2:

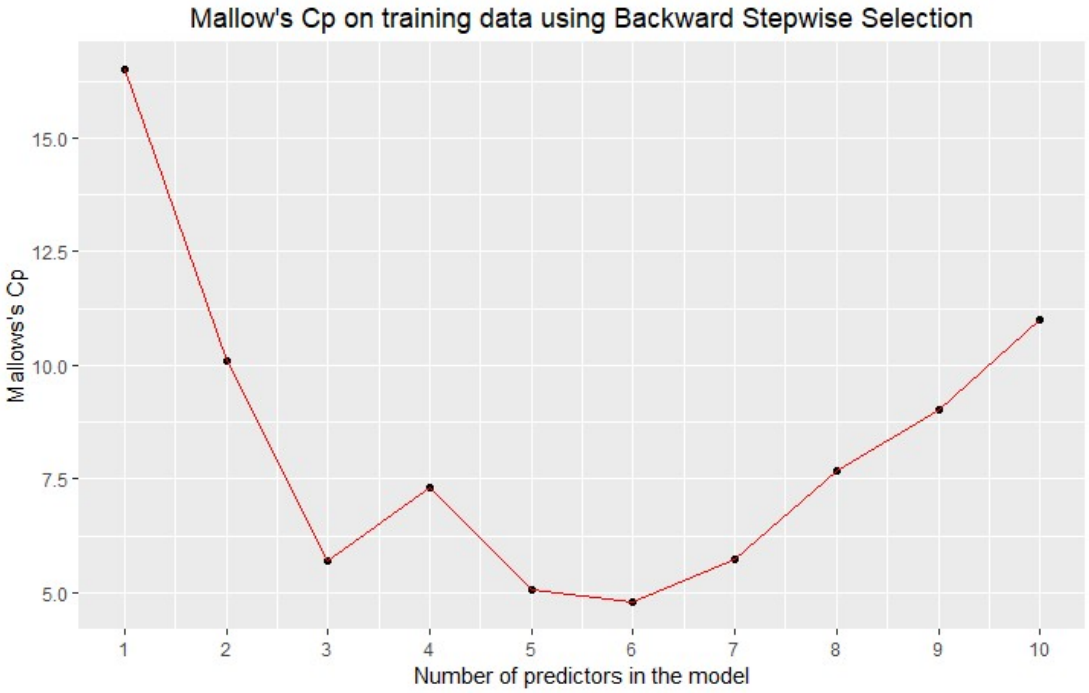
$$\hat{Y} = 1.3051636 + 0.4216466X + 1.2659763X^2 - 0.2631169X^4 + 0.0075466X^8 - 0.0005521X^{10}$$

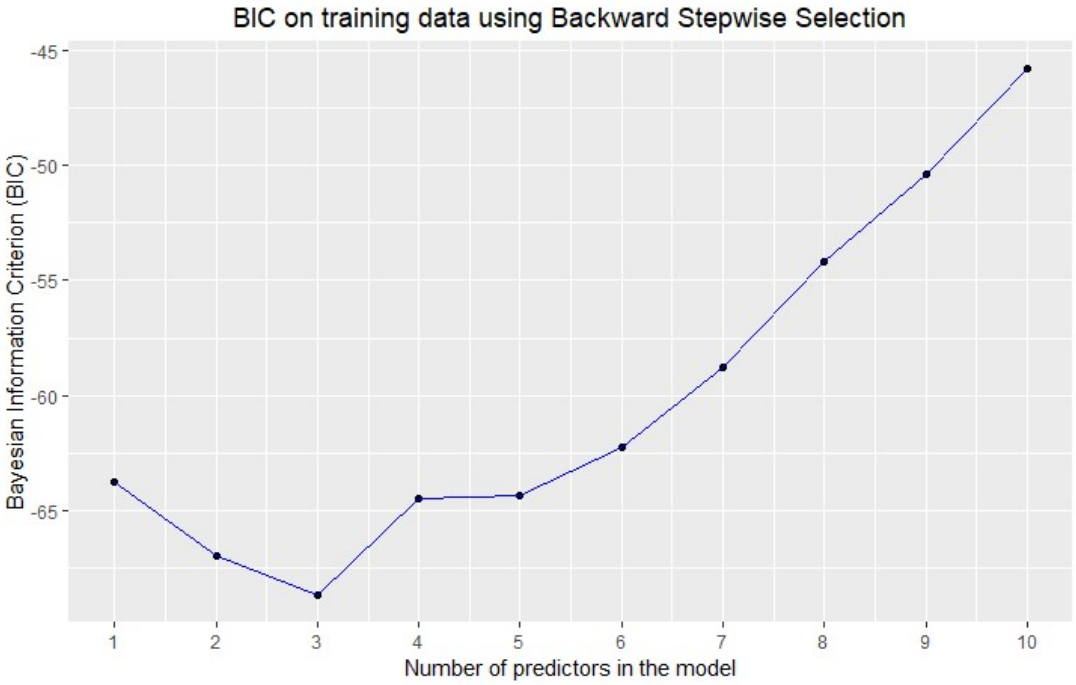
Using backward stepwise selection, the "k" variables in each of the "best" models for each value of "k" from 1-10 is shown below:

Results of Selection Algorithm using Backward Stepwise Selection

Number of Variables included in the model	X	X^2	X^3	X^4	X^5	X^6	X^7	X^8	X^9	X^{10}
1		*								
2		*	*							
3		*	*					*		
4		*	*			*		*		
5		*	*	*		*		*		
6		*	*	*	*	*		*		
7		*	*	*	*	*	*	*		
8		*	*	*	*	*	*	*	*	
9		*	*	*	*	*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*

For the "best" model from backward stepwise selection for each value of k, the Mallow's C_p, BIC, and adjusted R^2 were recorded for each value of k from 1-10 and plotted below:


Mallow's Cp on training data using Backward Stepwise Selection
Mallows's Cp
15.0
12.5
10.0
7.5
5.0
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model


BIC on training data using Backward Stepwise Selection
Bayesian Information Criterion (BIC)
-45
-50
-55
-60
-65
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model

From the plots above, we can see there is discrepancy as to the "best" model of the 10 possible models (one model for each k from 1-10) depending on the fit metric that is used.

- As per Mallow's C_p, the model with the lowest C_p (4.781005) is the six-variable model (with predictors $X^2, X^3, X^4, X^5, X^6, X^8$).
- As per the BIC, the model with the lowest BIC (-68.70039) is the three-variable model (with predictors X^2, X^3, X^8).
- As per the adjusted R^2, the model with the highest adjusted R^2 (0.5863764) is the seven-variable model (with predictors $X^2, X^3, X^4, X^5, X^6, X^7, X^8$).

The fitted six-variable, three-variable, and seven-variable models are shown below:

Best model from backward stepwise selection as per C_p:

$$\hat{Y} = 1.234947 + 1.774953X^2 + 0.241254X^3 - 0.685498X^4 - 0.027840X^5 + 0.118355X^6 - 0.005890X^8$$

Best model from backward stepwise selection as per BIC:

$$\hat{Y} = 1.4440823 + 0.6717126X^2 + 0.1144256X^3 - 0.0003668X^8$$

Best model from backward stepwise selection as per adjusted R^2:

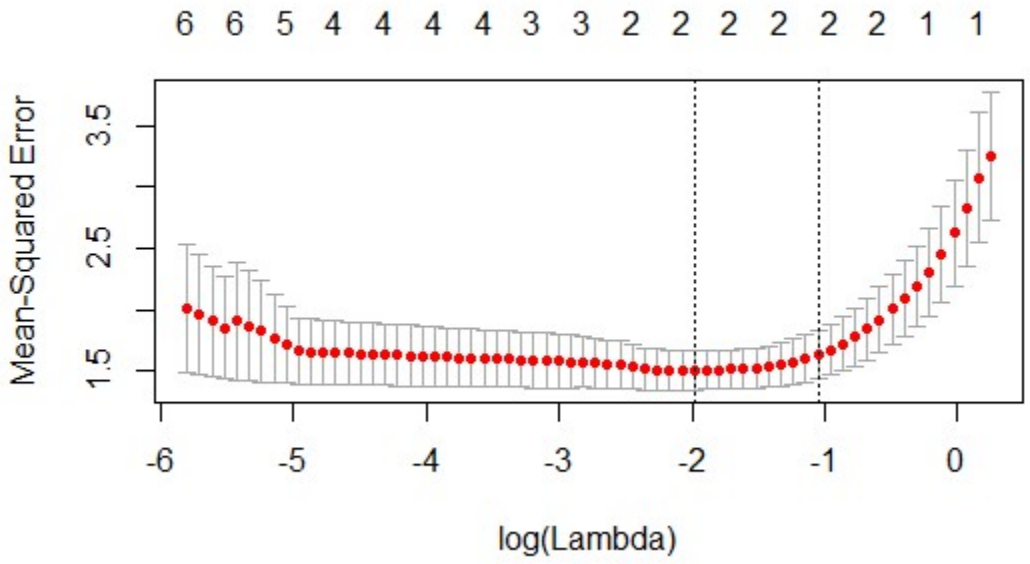
$$\hat{Y} = 1.229974 + 1.815732X^2 + 0.499450X^3 - 0.777759X^4 - 0.155375X^5 + 0.156954X^6 + 0.013722X^7 - 0.009964X^8$$

Comparing these results to the best subset selection results in part (c), we can see that other than the best model selection via BIC using forward stepwise selection, the results are all quite different than those generated using best subset selection. This shows that the choice of modeling approach, as well as choice of fit metric, can have a large impact on the interpretation of results.

(e) Not fit a LASSO model to the simulated data, again using $X, X^2, \ldots, X^{10}$ as predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

Answer:

Cross-validation was used to select the optimal value of λ for the LASSO. The plot of the cross-validation error as a function of λ is shown below:



The optimal value of λ is estimated to be $\lambda = 0.1377144$. The resulting coefficient estimates of the final LASSO model fit are shown below:

$$\hat{Y} = 1.571379 + 0.3390864X + 0.5311858X^2$$

As we can see from the final equation above, the LASSO model shrunk all of the coefficient estimates to zero other than the coefficients for X and X^2. We can also see that this model is similar to the "two-variable" OLS model with X and X^2 as predictors. But as compared to the OLS "two-variable" model (as shown in part (c) of this problem for the best model from best subset selection as per BIC) the regression coefficients for X and X^2 are shrunk more towards zero as compared to the OLS estimates ($\hat{Y} = 1.4899 + 0.4339X + 0.5858X^2$). This is what we would expect from the LASSO with $\lambda > 0$.

(f) Now generate a response vector Y according to the model:

$$Y = \beta_0 + \beta_7 X^7 + \epsilon$$

And perform best subset selection and the LASSO. Discuss the results obtained.

Answer:

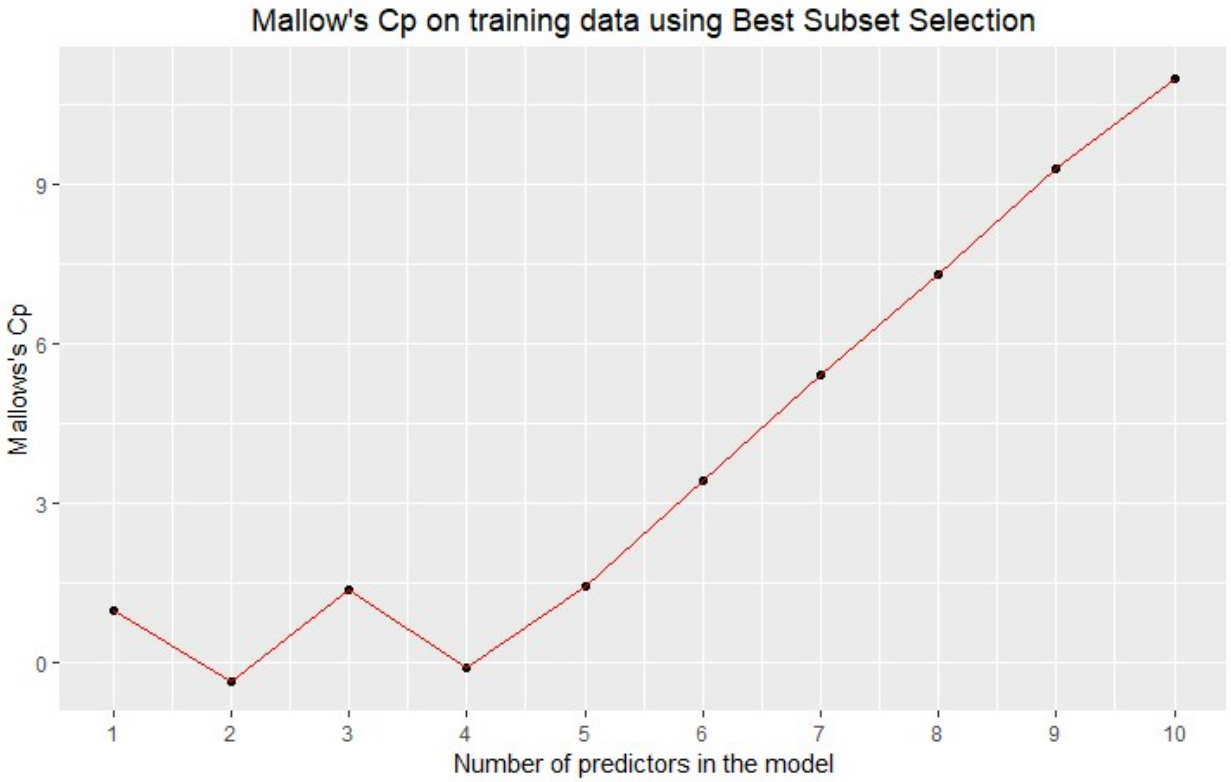
Best subset selection:

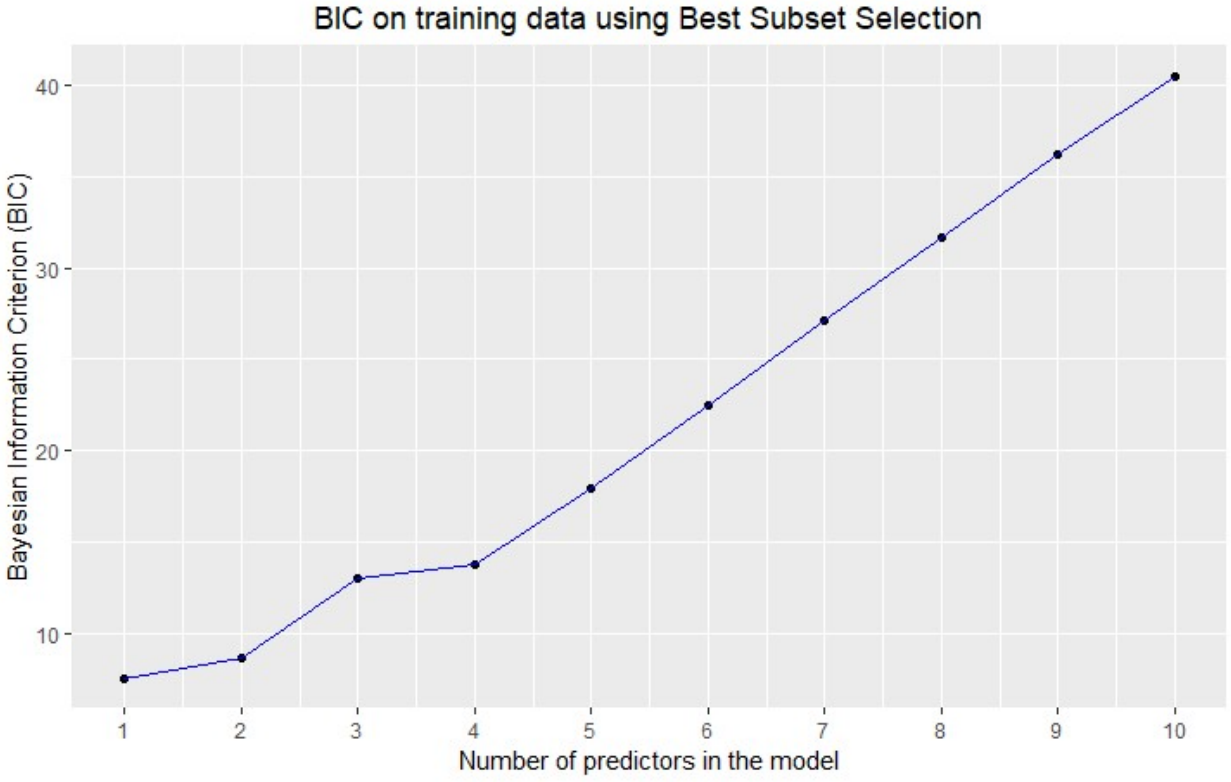
Using best subset selection, the "k" variables in each of the "best" models for each value of "k" from 1-10 is shown below:

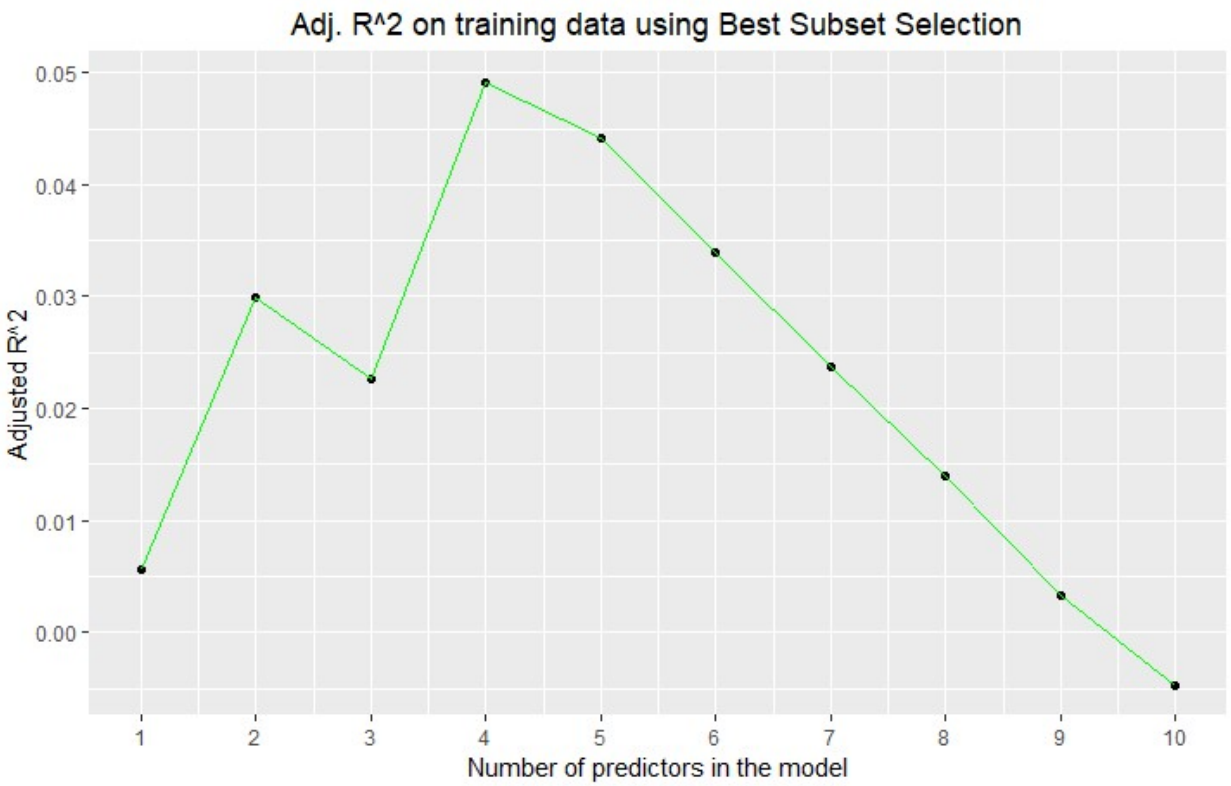
Results of Selection Algorithm using Best Subset Selection

Number of Variables included in the model	X	X^2	X^3	X^4	X^5	X^6	X^7	X^8	X^9	X^{10}
1		*								
2								*		*
3		*						*		*
4		*		*		*		*		
5	*	*		*		*		*		
6	*	*		*		*		*		*
7	*	*		*		*		*	*	*
8	*	*	*	*	*	*	*	*		
9	*	*	*	*	*	*	*	*		*
10	*	*	*	*	*	*	*	*	*	*

For the "best" model from best subset selection for each value of k, the Mallow's C_p, BIC, and adjusted R^2 were recorded for each value of k from 1-10 and plotted below:


Mallow's Cp on training data using Best Subset Selection
Mallows's Cp
0
3
6
9
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model


BIC on training data using Best Subset Selection
Bayesian Information Criterion (BIC)
10
20
30
40
1
2
3
4
5
6
7
8
9
10
Number of predictors in the model



From the plots above, we can see there is discrepancy as to the "best" model of the 10 possible models (one model for each k from 1-10) depending on the fit metric that is used.

- As per Mallow's C_p, the model with the lowest C_p (-0.33652829) is the two-variable model (with predictors X^8, X^{10}).
- As per the BIC, the model with the lowest BIC (7.625395) is the one-variable model (with predictor X^2).
- As per the adjusted R^2, the model with the highest adjusted R^2 (0.049190624) is the four-variable model (with predictors X^2, X^4, X^6, X^8).

The fitted six-variable, three-variable, and seven-variable models are shown below:

Best model from best subset selection as per C_p:

$$\hat{Y} = 1.534 + 0.001501X^8 - 0.000147X^{10}$$

Best model from best subset selection as per BIC:

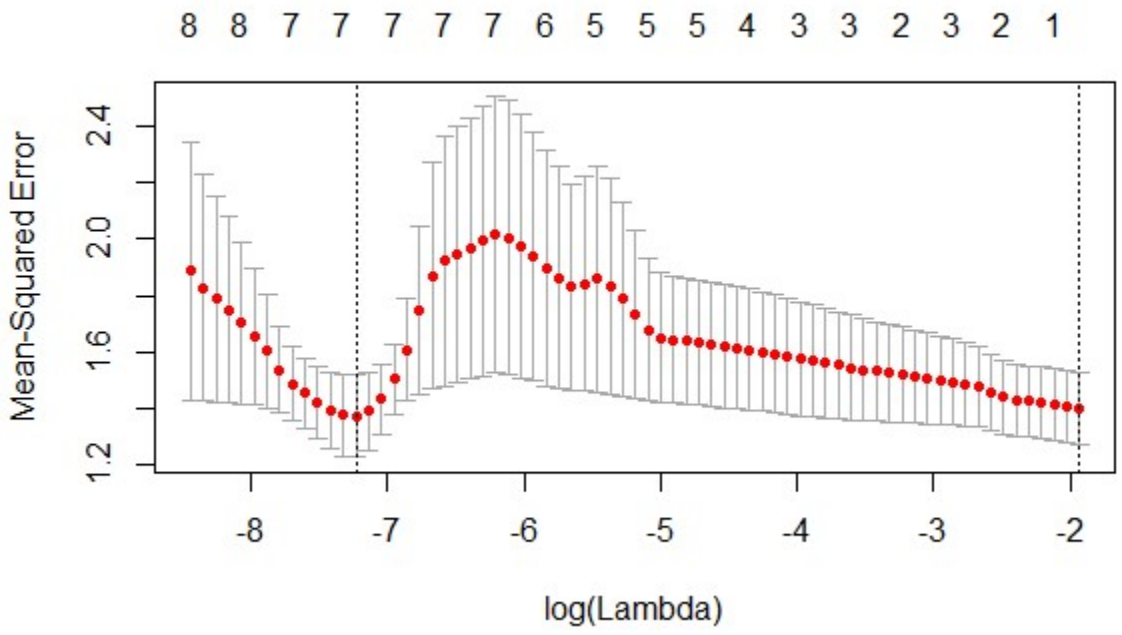
$$\hat{Y} = 1.49607 + 0.07463X^2$$

Best model from best subset selection as per adjusted R^2:

$$\hat{Y} = 1.249839 + 1.125182X^2 - 0.603910X^4 + 0.107738X^6 - 0.005766X^8$$

LASSO Model:

Cross-validation was used to select the optimal value of λ for the LASSO. The plot of the cross-validation error as a function of λ is shown below:



The optimal value of λ is estimated to be $\lambda = 0.000723535$. The resulting coefficient estimates of the final LASSO model fit are shown below:

$$\hat{Y} = 1.325966 - 0.09070205X + 0.7460786X^2 - 0.3393815X^4 + 0.003467248X^5 + 0.04561203X^6 - 0.00005961914X^9 - 0.0001722417X^{10}$$

As we can see from the final equation above, the LASSO model shrunk the coefficient estimates for X^3, X^7, X^8 to zero.

Ultimately, the results from the best-subset selection and from the LASSO are different. Even the "best" model within the best-subset selection methodology is different depending on the fit metric used. This exercise is a good lesson to illustrate how different modeling approaches and fit metrics impose different assumptions, and results can be quite sensitive to those assumptions.

R-code:

```
#################
## Problem #4 ##
#################
library(leaps)
library(dplyr)
library(glmnet)
library(psych)
library(ggplot2)
set.seed(10815657)

#############
## Part A ##
#############
n=100
X <- rt(n, 15)
error <- rt(n, 15)
psych::describe(X)
psych::describe(error)


#############
## Part B ##
#############
Y <- 0.5 + (0.5*X) + (0.5*(X^2)) + (0.5&(X^3)) + error
psych::describe(Y)

#############
## Part C ##
#############
train_df <- data.frame(Y,X)
train_df$X2 <- X^2
train_df$X3 <- X^3
train_df$X4 <- X^4
train_df$X5 <- X^5
train_df$X6 <- X^6
train_df$X7 <- X^7
train_df$X8 <- X^8
train_df$X9 <- X^9
train_df$X10 <- X^10
head(train_df)
```

```
num_variables <- c(1:10)
best_subset <- summary(regsubsets(Y~., data=train_df, nvmax=10))
bs_df <- data.frame(num_variables)
bs_df$cp <- best_subset$cp
bs_df$bic <- best_subset$bic
bs_df$adjr2 <- best_subset$adjr2
bs_df
summary(lm(Y~X+X2, data=train_df))
summary(lm(Y~X+X2+X4+X6+X8, data=train_df))

ggplot(bs_df, aes(x=num_variables, y=cp)) + geom_point() + geom_line(color="red") +
  ggtitle("Mallow's Cp on training data using Best Subset Selection") +  xlab("Number of
predictors in the model") + ylab("Mallows's Cp") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(bs_df, aes(x=num_variables, y=bic)) + geom_point() + geom_line(color="blue") +
  ggtitle("BIC on training data using Best Subset Selection") +  xlab("Number of predictors in
the model") + ylab("Bayesian Information Criterion (BIC)") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(bs_df, aes(x=num_variables, y=adjr2)) + geom_point() + geom_line(color="green") +
  ggtitle("Adj. R^2 on training data using Best Subset Selection") +  xlab("Number of predictors
in the model") + ylab("Adjusted R^2") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)


#############
## Part D ##
#############
forward_selection <- summary(regsubsets(Y~., data=train_df, nvmax=10, method="forward"))
fs_df <- data.frame(num_variables)
fs_df$cp <- forward_selection$cp
fs_df$bic <- forward_selection$bic
fs_df$adjr2 <- forward_selection$adjr2
fs_df
summary(lm(Y~X+X2, data=train_df))
summary(lm(Y~X+X2+X4+X8+X10, data=train_df))

ggplot(fs_df, aes(x=num_variables, y=cp)) + geom_point() + geom_line(color="red") +
  ggtitle("Mallow's Cp on training data using Forward Stepwise Selection") +  xlab("Number of
predictors in the model") + ylab("Mallows's Cp") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(fs_df, aes(x=num_variables, y=bic)) + geom_point() + geom_line(color="blue") +
  ggtitle("BIC on training data using Forward Stepwise Selection") +  xlab("Number of predictors
in the model") + ylab("Bayesian Information Criterion (BIC)") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(fs_df, aes(x=num_variables, y=adjr2)) + geom_point() + geom_line(color="green") +
  ggtitle("Adj. R^2 on training data using Forward Stepwise Selection") +  xlab("Number of
predictors in the model") + ylab("Adjusted R^2") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

backward_selection <- summary(regsubsets(Y~., data=train_df, nvmax=10, method="backward"))
bs_df <- data.frame(num_variables)
bs_df$cp <- backward_selection$cp
bs_df$bic <- backward_selection$bic
bs_df$adjr2 <- backward_selection$adjr2
bs_df
summary(lm(Y~X2+X3+X4+X5+X6+X8, data=train_df))
summary(lm(Y~X2+X3+X8, data=train_df))
summary(lm(Y~X2+X3+X4+X5+X6+X7+X8, data=train_df))

ggplot(bs_df, aes(x=num_variables, y=cp)) + geom_point() + geom_line(color="red") +
```

```
  ggtitle("Mallow's Cp on training data using Backward Stepwise Selection") +  xlab("Number of
predictors in the model") + ylab("Mallows's Cp") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(bs_df, aes(x=num_variables, y=bic)) + geom_point() + geom_line(color="blue") +
  ggtitle("BIC on training data using Backward Stepwise Selection") +  xlab("Number of predictors
in the model") + ylab("Bayesian Information Criterion (BIC)") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(bs_df, aes(x=num_variables, y=adjr2)) + geom_point() + geom_line(color="green") +
  ggtitle("Adj. R^2 on training data using Backward Stepwise Selection") +  xlab("Number of
predictors in the model") + ylab("Adjusted R^2") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

#############
## Part E ##
#############
X_train <- as.matrix(dplyr::select(train_df, -Y))
LASSO_CV_tuned <- cv.glmnet(X_train, Y, alpha=1)
plot(LASSO_CV_tuned)
tuned_lambda <- LASSO_CV_tuned$lambda.min
LASSO <- glmnet(X_train, Y, alpha=1, lambda=tuned_lambda)
LASSO$a0
LASSO$beta

#############
## Part F ##
#############
Y_alt <- 0.5 + (0.5&(X^7)) + error

train_df_alt <- data.frame(Y_alt,X)
train_df_alt$X2 <- X^2
train_df_alt$X3 <- X^3
train_df_alt$X4 <- X^4
train_df_alt$X5 <- X^5
train_df_alt$X6 <- X^6
train_df_alt$X7 <- X^7
train_df_alt$X8 <- X^8
train_df_alt$X9 <- X^9
train_df_alt$X10 <- X^10
head(train_df_alt)

num_variables <- c(1:10)
best_subset <- summary(regsubsets(Y_alt~., data=train_df_alt, nvmax=10))
bs_df <- data.frame(num_variables)
bs_df$cp <- best_subset$cp
bs_df$bic <- best_subset$bic
bs_df$adjr2 <- best_subset$adjr2
bs_df

summary(lm(Y_alt~X8+X10, data=train_df_alt))
summary(lm(Y_alt~X2, data=train_df_alt))
summary(lm(Y_alt~X2+X4+X6+X8, data=train_df_alt))

ggplot(bs_df, aes(x=num_variables, y=cp)) + geom_point() + geom_line(color="red") +
  ggtitle("Mallow's Cp on training data using Best Subset Selection") +  xlab("Number of
predictors in the model") + ylab("Mallows's Cp") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(bs_df, aes(x=num_variables, y=bic)) + geom_point() + geom_line(color="blue") +
  ggtitle("BIC on training data using Best Subset Selection") +  xlab("Number of predictors in
the model") + ylab("Bayesian Information Criterion (BIC)") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

ggplot(bs_df, aes(x=num_variables, y=adjr2)) + geom_point() + geom_line(color="green") +
```

```
  ggtitle("Adj. R^2 on training data using Best Subset Selection") +  xlab("Number of predictors
in the model") + ylab("Adjusted R^2") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_x_continuous(breaks=num_variables)

X_train <- as.matrix(dplyr::select(train_df_alt, -Y_alt))
LASSO_CV_tuned <- cv.glmnet(X_train, Y_alt, alpha=1)
plot(LASSO_CV_tuned)
tuned_lambda <- LASSO_CV_tuned$lambda.min
LASSO <- glmnet(X_train, Y_alt, alpha=1, lambda=tuned_lambda)
LASSO$a0
LASSO$beta
```

Problem #5:

Modified from exercise 11, section 6.8 of ITSL textbook

We will not try to predict in the "College" data set.

(a) Try out some of the regression methods explored in this chapter, such as best subset selection, the LASSO, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

Answer:

For this problem, I evaluated six different approaches. For each of the six approaches, 10-fold cross validation was conducted with the 10-fold CV-MSE enumerated. Within each round of the 10-fold cross validation, the hyper-parameters of each of the 6 methods were tuned by:

- **LASSO**: with the hyperparameter λ tuned via cross validation with the lowest MSE
- **Ridge Regression**: with the hyperparameter λ tuned via cross validation with the lowest MSE
- **Best Subset Selection**: with the optimal model of the "k" models chosen via the largest adjusted R^2
- **Forward Stepwise Selection**: with the optimal model of the "k" models chosen via the largest adjusted R^2.
- **Backward Stepwise Selection**: with the optimal model of the "k" models chosen via the largest adjusted R^2.
- **Principal Component Regression**: with the optimal number of principal components to include in the regression chosen via the minimum adjusted CV-MSE

10-fold Cross Validated Mean Squared Error

Method	Mean	Standard Deviation
LASSO	776643.5	942473.00
Ridge Regression	689720.2	1319575.48
Best Subset Selection	803980.5	978130.92
Forward Stepwise Selection	803980.5	978130.92
Backward Stepwise Selection	803980.5	978130.92
Principal Component Regression	788556.7	968838.61

It's important to keep in mind that these results show the mean MSE (over 10-fold cross validated samples), where the decision rules regarding hyperparameter tunings in each respective method were carried out within each cross-validation step. Therefore these results provide a general sense of how we would expect a given method to perform on a hold-out test set. For the final model I will report below, I will be fitting the model on the entire dataset (without a hold-out set). But we can use the CV results above to get a sense of how we would expect this final model to perform on predicting "new" data.

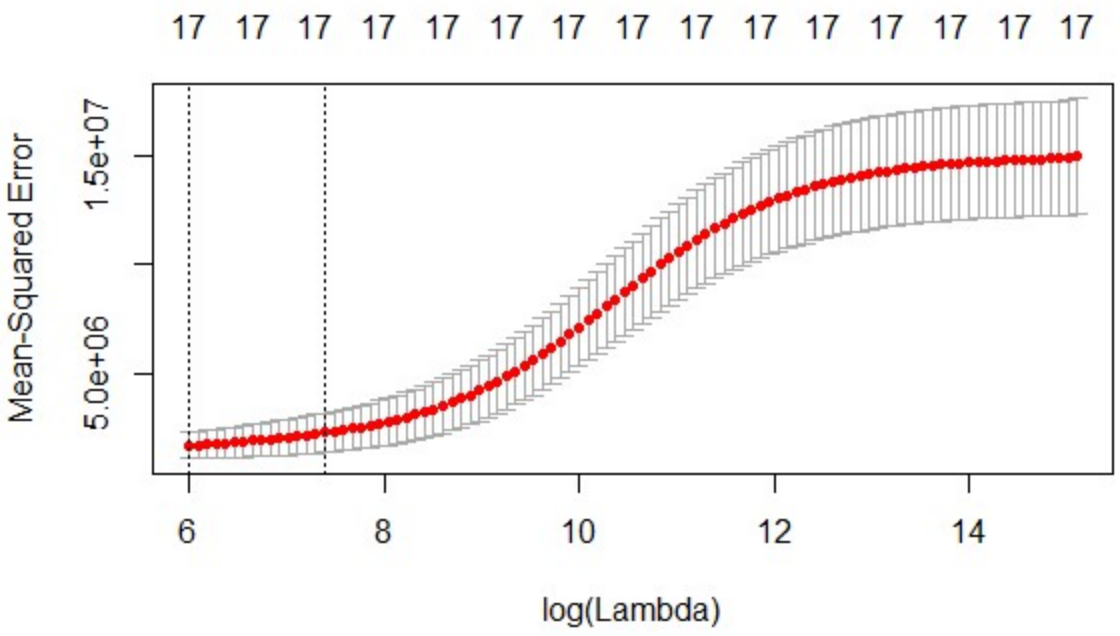
From the results above, we can see the model with the lowest mean CV-MSE is the Ridge Regression model (with mean CV-MSE of 689720). Therefore, for part (b) of this problem I will fit the Ridge Regression methodology to the entire training set and report the resulting coefficient estimates.

(b) Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross-validation, or some other reasonable alternative, as opposed to using training error.

Answer:

As per the explanation in part (a), we fit and report the Ridge Regression model fit on the entire dataset. Following the methodology we used in part (a) in each round of the 10-fold cross-validation, we first use cross-validation (but now on the entire dataset) to tune the hyperparameter λ .

The plot of the cross-validation error as a function of λ is shown below:



The optimal value of λ is estimated to be $\lambda = 400.4766$. The resulting coefficient estimates of the final Ridge Regression model are shown below:

$$\begin{aligned}\widehat{Apps} = -1516.527 &+ (0.9766527 * Accept) + (0.4715153 * Enroll) \\ &+ (24.92872 * Top10perc) + (1.081058 * Top25perc) \\ &+ (0.07645523 * F.Undergrad) + (0.02437302 * P.Undergrad) \\ &- (0.02129295 * Outstate) + (0.2001279 * Room.Board) \\ &+ (0.1354312 * Books) - (0.009192378 * Personal) - (3.763649 * PhD) \\ &- (4.718359 * Terminal) + (12.78196 * S.F.Ratio) \\ &- (8.836976 * perc.alumni) + (0.07527017 * Expand) \\ &+ (11.36678 * Grad.Rate) - (529.3906 * PrivateSchool)\end{aligned}$$

Note that for the model above, the variable "PrivateSchool" is a transformation of the original variable "Private" with the logic:

- Private=="Yes"; PrivateSchool=1
- Private=="No"; PrivateSchool=0

As I mentioned previously, the entire dataset was used to fit the Ridge Regression model above. To get an appropriate estimate of the "test MSE" that we would expect, the mean CV-MSE in part (a) of this problem is an appropriate estimate of the "test MSE". That mean CV-MSE from part (a) for the Ridge Regression is 689720.2

(c) Does your chosen model involve all of the features in the data set? What or why not?

Answer:

Yes, the chosen model does include all 17 of the possible features in the model. This is because the final model used was a Ridge Regression model. Unlike some of the other modeling candidates that may include only a subset of the features (i.e. LASSO, best subset selection, etc), the Ridge Regression includes all of the parameters in the model. The L2-norm regularization term does shrink the resulting coefficient estimates towards the null. However (unlike the LASSO model), in a ridge model the regularization term will not shrink any of the coefficient estimates perfectly to zero. This is why we know all 17 features are included in the final Ridge model.

R-Code:

```
#################
## Problem #5 ##
#################
library(leaps)
library(dplyr)
library(glmnet)
library(psych)
library(ggplot2)
library(ISLR)
library(pls)
set.seed(123456)

## load the "college" dataset
df <- College
head(df)
df$private_school <- 0
df$private_school[df$Private=='Yes'] <- 1
table(df$Private, df$private_school)
df <- dplyr::select(df, -Private)
head(df)

## specify the "CV_id"
k <- 10
df$CV_id <- sample(c(1:k), length(df$Apps), replace=TRUE)
table(df$CV_id)

CV_trial <- c(1:k)
MSE_df <- data.frame(CV_trial)
MSE_df$LASSO <- NA
MSE_df$Ridge <- NA
MSE_df$BestSub <- NA
MSE_df$ForwS <- NA
MSE_df$BackD <- NA
MSE_df$PCR <- NA

for(i in 1:k){
  print(i)
  df_train <- select(filter(df, CV_id!=i), -CV_id)
  df_test <- select(filter(df, CV_id==i), -CV_id)

  Y_train <- as.matrix(select(df_train, Apps))
  Y_test <- as.matrix(select(df_test, Apps))
  X_train <- as.matrix(select(df_train, -Apps))
  X_test <- as.matrix(select(df_test, -Apps))

  predicted_df <- data.frame(Y_test)

  ## LASSO model
  LASSO_CV_tuned <- cv.glmnet(X_train, Y_train, alpha=1)
  lambda_LASSO <- LASSO_CV_tuned$lambda.min
  LASSO <- glmnet(X_train, Y_train, alpha=1, lambda=lambda_LASSO)
  predicted_df$LASSO_p  <- predict(LASSO, s=lambda_LASSO, newx=X_test)
  rm(LASSO_CV_tuned, lambda_LASSO, LASSO)

  ## Ridge Model
  Ridge_CV_tuned <- cv.glmnet(X_train, Y_train, alpha=0)
  lambda_Ridge <- Ridge_CV_tuned$lambda.min
  Ridge <- glmnet(X_train, Y_train, alpha=0, lambda=lambda_Ridge)
  predicted_df$Ridge_p  <- predict(Ridge, s=lambda_Ridge, newx=X_test)
  rm(Ridge_CV_tuned, lambda_Ridge, Ridge)

  ## Best Subset
  bs <- summary(regsubsets(Apps~., data=df_train, nvmax=17))
  bs_vector <- bs$which[which.max(bs$adjr2),]
```

```
  bs_variables <- names(bs_vector[bs_vector==TRUE])[2:length(bs_vector[bs_vector==TRUE])]
  bs_train <- select(df_train, Apps, bs_variables)
  bs_model <- lm(Apps~., data=bs_train)
  predicted_df$bs_p  <- predict(bs_model, newdata=df_test)
  rm(bs, bs_vector, bs_variables, bs_train, bs_model)

  ## Forward Selection
  fs <- summary(regsubsets(Apps~., data=df_train, nvmax=17, method="forward"))
  fs_vector <- fs$which[which.max(fs$adjr2),]
  fs_variables <- names(fs_vector[fs_vector==TRUE])[2:length(fs_vector[fs_vector==TRUE])]
  fs_train <- select(df_train, Apps, fs_variables)
  fs_model <- lm(Apps~., data=fs_train)
  predicted_df$fs_p  <- predict(fs_model, newdata=df_test)
  rm(fs, fs_vector, fs_variables, fs_train, fs_model)

  ## Backward Selection
  bw <- summary(regsubsets(Apps~., data=df_train, nvmax=17, method="backward"))
  bw_vector <- bw$which[which.max(bw$adjr2),]
  bw_variables <- names(bw_vector[bw_vector==TRUE])[2:length(bw_vector[bw_vector==TRUE])]
  bw_train <- select(df_train, Apps, bw_variables)
  bw_model <- lm(Apps~., data=bw_train)
  predicted_df$bw_p  <- predict(bw_model, newdata=df_test)
  rm(bw, bw_vector, bw_variables, bw_train, bw_model)

  ## PCR
  PCR <- pcr(Apps~., data=df_train, scale=TRUE, validation="CV")
  predicted_df$pcr_p  <- predict(PCR, newdata=df_test, ncomp=which.min(PCR$validation$adj))
  rm(PCR)

  MSE_df$LASSO[i] <- (sum(predicted_df$Apps - predicted_df$LASSO_p)^2)/dim(predicted_df)[1]
  MSE_df$Ridge[i] <- (sum(predicted_df$Apps - predicted_df$Ridge_p)^2)/dim(predicted_df)[1]
  MSE_df$BestSub[i] <- (sum(predicted_df$Apps - predicted_df$bs_p)^2)/dim(predicted_df)[1]
  MSE_df$ForwS[i] <- (sum(predicted_df$Apps - predicted_df$fs_p)^2)/dim(predicted_df)[1]
  MSE_df$BackD[i] <- (sum(predicted_df$Apps - predicted_df$bw_p)^2)/dim(predicted_df)[1]
  MSE_df$PCR[i] <- (sum(predicted_df$Apps - predicted_df$pcr_p)^2)/dim(predicted_df)[1]

  rm(predicted_df)
}

describe(MSE_df)

Y <- as.matrix(select(df, Apps))
X <- as.matrix(select(df, -Apps, -CV_id))
Ridge_CV_tuned <- cv.glmnet(X, Y, alpha=0)
plot(Ridge_CV_tuned)
lambda_Ridge <- Ridge_CV_tuned$lambda.min
Ridge <- glmnet(X, Y, alpha=0, lambda=lambda_Ridge)
```