Homework 5 Peer Assessment

Richard Albright
ISYE6414
Spring 2020

Spring Semester 2020

Background

The Boston Housing Price Dataset was obtained from the StatLib library which is maintained at Carnegie Mellon University. It contains US census data concerning houses in various areas around the city of Boston.

The dataset consists of 506 observations of 14 attributes. Below is a brief description of each feature and the outcome in our dataset:

1. *crim* - per capita crime rate by town
2. *zn* - proportion of residential land zoned for lots over 25,000 sq.ft
3. *indus* - proportion of non-retail business acres per town
4. *chas* - Charles River dummy variable (1 if tract bounds river; else 0)
5. *nox* - nitric oxides concentration (parts per 10 million)
6. *rm* - average number of rooms per dwelling
7. *age* - proportion of owner-occupied units built prior to 1940
8. *dis* - weighted distances to five Boston employment centres
9. *rad* - index of accessibility to radial highways
10. *tax* - full-value property-tax rate per $10,000
11. *ptratio* - pupil-teacher ratio by town
12. *black* - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. *lstat* - % lower status of the population
14. *medv* - Median value of owner-occupied homes in $1000's

Please load the dataset "Boston" and then split the dataset into a train and test set in a 80:20 ratio. Use the training set to build the models in Questions 1-6. Use the test set to help evaluate model performance in Question 7. Please make sure that you are using R version 3.6.X.

Read Data

```
set.seed(100)

fullData = read.csv("Boston.csv",header=TRUE)
testRows = sample(nrow(fullData),0.2*nrow(fullData))
testData = fullData[testRows, ]
trainData = fullData[-testRows, ]
```

Question 1: Full Model

(a) Fit a standard linear regression with the variable *medv* as the response and the other variables as predictors. Call it *model1*. Display the model summary.

```r
model1 <- glm(medv~., data = trainData)
summ(model1, digits=4)
```

Observations	405
Dependent variable	medv
Type	Linear regression

$\chi^2(13)$	24759.6755
Pseudo-R² (Cragg-Uhler)	0.7416
Pseudo-R² (McFadden)	0.1864
AIC	2419.2813
BIC	2479.3396

	Est.	S.E.	t val.	p
(Intercept)	33.9226	5.9279	5.7225	0.0000
crim	-0.1170	0.0379	-3.0867	0.0022
zn	0.0503	0.0148	3.3935	0.0008
indus	0.0361	0.0681	0.5302	0.5963
chas	3.1730	0.9688	3.2752	0.0011
nox	-18.5471	4.4502	-4.1677	0.0000
rm	4.0803	0.5003	8.1560	0.0000
age	-0.0004	0.0151	-0.0281	0.9776
dis	-1.4244	0.2220	-6.4155	0.0000
rad	0.3062	0.0755	4.0538	0.0001
tax	-0.0126	0.0042	-3.0069	0.0028
ptratio	-0.9102	0.1427	-6.3765	0.0000
black	0.0082	0.0031	2.6477	0.0084
lstat	-0.4754	0.0578	-8.2222	0.0000

Standard errors: MLE

(b) Which regression coefficients are significant at the 95% confidence level? At the 99% confidence level?

The following coefficient is significant at a 95% confidence level.

Standard errors: MLE

	Est.	S.E.	t val.	p
crim	-0.0899	0.0399	-2.2504	0.0250

The following coefficients are significant at a 99% confidence level.

Standard errors: MLE

	Est.	S.E.	t val.	p
(Intercept)	33.1309	5.7243	5.7878	0.0000
zn	0.0399	0.0149	2.6728	0.0078
chas	3.0409	0.9888	3.0753	0.0023
nox	-16.7945	4.3047	-3.9015	0.0001

	Est.	S.E.	t val.	p
rm	4.1985	0.4662	9.0053	0.0000
dis	-1.3552	0.2161	-6.2710	0.0000
rad	0.2997	0.0768	3.9008	0.0001
tax	-0.0142	0.0043	-3.2930	0.0011
ptratio	-0.9112	0.1431	-6.3689	0.0000
black	0.0082	0.0029	2.7983	0.0054
lstat	-0.4673	0.0569	-8.2167	0.0000

(c) What are the 10-fold and leave one out cross-validation scores for this model?

```
set.seed(100)
model1.cv = cv.glm(data=trainData, glmfit=model1, K=10)
```

10-fold cross-validation MSE: 23.8536146

```
  model1.loocv = cv.glm(data=trainData, glmfit=model1, K=nrow(trainData))
```

LOOCV cross-validation MSE: 23.8217281

(d) What are the Mallow's Cp, AIC, and BIC criterion values for this model?

From the summary of using the glm function the AIC and BIC is as follows:

AIC: 2419.28

BIC: 2479.34

Extracting the residual standard error from the model. Mallow's Cp is as follows:

```
set.seed(100)
model1.rse <- sqrt(deviance(model1)/df.residual(model1))
model1.cp <-Cp(model1, S2=model1.rse)
```

Mallow's Cp: 1462.21

(e) Build a new model on the training data with only the variables which coefficients were found to be statistically significant at the 99% confident level. Call it *model2*. Perform an ANOVA test to compare this new model with the full model. Which one would you prefer? Is it good practice to select variables based on statistical significance of individual coefficients? Explain.

```
set.seed(100)
model2 <- glm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat, data=trainData)
summ(model2, digits=4)
```

Observations	405
Dependent variable	medv
Type	Linear regression

$\chi^2(11)$	24753.4264
Pseudo-R^2 (Cragg-Uhler)	0.7414
Pseudo-R^2 (McFadden)	0.1863
AIC	2415.5737
BIC	2467.6242

```
print_output(anova(model2, model1, test='F'))
```

	Est.	S.E.	t val.	p
(Intercept)	33.7726	5.8469	5.7762	0.0000
crim	-0.1178	0.0378	-3.1166	0.0020
zn	0.0497	0.0146	3.4118	0.0007
chas	3.2405	0.9569	3.3865	0.0008
nox	-17.9460	4.0128	-4.4722	0.0000
rm	4.0499	0.4834	8.3783	0.0000
dis	-1.4473	0.2099	-6.8965	0.0000
rad	0.2968	0.0729	4.0711	0.0001
tax	-0.0117	0.0039	-3.0510	0.0024
ptratio	-0.8995	0.1400	-6.4256	0.0000
black	0.0081	0.0031	2.6375	0.0087
lstat	-0.4735	0.0533	-8.8906	0.0000

Standard errors: MLE

```
Analysis of Deviance Table

Model 1: medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
    black + lstat
Model 2: medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
    tax + ptratio + black + lstat
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1       393     8657.6
2       391     8651.3  2   6.2491 0.1412 0.8683
```

The null hypothesis is the removed variables' coefficients are not different from 0. The alternate hypothesis is that the removed variables' coefficients are 0. At a 95% confidence level, we accept the null hypothesis and choose the reduced model. It is not good practice to select variables based on statistical significance base on individual coefficients. A variable may not be statistically significant only in the context of the current model selected, and may be statistically sigificant in a different model.

Question 2: Full Model Search

(a) Compare all possible models using Mallow's Cp. What is the total number of possible models with the full set of variables? Display a table indicating the variables included in the best model of each size and the corresponding Mallow's Cp value.

Hint: The table must include 13 models. You can use nbest parameter.

```
set.seed(100)

out = leaps(trainData[,c(1:13)],
            trainData$medv,
            nbest=1,
            method = "Cp",
            names=c('crim', 'zn', 'indus', 'chas',
                    'nox', 'rm', 'age', 'dis', 'rad',
                    'tax', 'ptratio', 'black', 'lstat'))
print_output(cbind(as.matrix(out$which),out$Cp))
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	
1	0	0	0	0	0	0	0	0	0	0	0	0	1	298.07927
2	0	0	0	0	0	1	0	0	0	0	0	0	1	146.92399
3	0	0	0	0	0	1	0	0	0	0	1	0	1	87.73031
4	0	0	0	1	0	1	0	0	0	0	1	0	1	71.19702
5	0	0	0	0	1	1	0	1	0	0	1	0	1	50.68647
6	0	0	0	1	1	1	0	1	0	0	1	0	1	37.41118
7	0	1	0	1	1	1	0	1	0	0	1	0	1	31.71986
8	0	1	0	1	1	1	0	1	0	0	1	1	1	25.32215
9	1	1	0	1	1	1	0	1	0	0	1	1	1	22.81194
10	1	1	0	1	1	1	0	1	1	1	1	0	1	15.20861
11	1	1	0	1	1	1	0	1	1	1	1	1	1	10.28243
12	1	1	1	1	1	1	0	1	1	1	1	1	1	12.00079
13	1	1	1	1	1	1	1	1	1	1	1	1	1	14.00000

(b) How many variables are in the model with the lowest Mallow's Cp value? Which variables are they? Fit this model and call it *model3*.

```
set.seed(100)
best.model = which(out$Cp==min(out$Cp))
print_output(cbind(as.matrix(out$which),out$Cp)[best.model,])
```

```
     crim        zn     indus      chas       nox        rm       age       dis
  1.00000   1.00000   0.00000   1.00000   1.00000   1.00000   0.00000   1.00000
      rad       tax   ptratio     black     lstat
  1.00000   1.00000   1.00000   1.00000   1.00000  10.28243
```

There are 11 variables in the model with the lowest Cp of 10.2824307. Those are crim, zn, chas, nox, rm, dis, rad, tax, ptratio, black, and lstat.

```
model3 <- glm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat, data=trainData)
summ(model3, digits=4)
```

Observations	405
Dependent variable	medv
Type	Linear regression

$\chi^2(11)$	24753.4264
Pseudo-R^2 (Cragg-Uhler)	0.7414
Pseudo-R^2 (McFadden)	0.1863
AIC	2415.5737
BIC	2467.6242

Question 3: Stepwise Regression

(a) Perform backward stepwise regression using BIC. Allow the minimum model to be the model with only an intercept, and the full model to be *model1*. Display the model summary of your final model. Call it *model4*

```
set.seed(100)
minmodel <- glm(medv~1, data=trainData)
model4 <- step(model1,
```

	Est.	S.E.	t val.	p
(Intercept)	33.7726	5.8469	5.7762	0.0000
crim	-0.1178	0.0378	-3.1166	0.0020
zn	0.0497	0.0146	3.4118	0.0007
chas	3.2405	0.9569	3.3865	0.0008
nox	-17.9460	4.0128	-4.4722	0.0000
rm	4.0499	0.4834	8.3783	0.0000
dis	-1.4473	0.2099	-6.8965	0.0000
rad	0.2968	0.0729	4.0711	0.0001
tax	-0.0117	0.0039	-3.0510	0.0024
ptratio	-0.8995	0.1400	-6.4256	0.0000
black	0.0081	0.0031	2.6375	0.0087
lstat	-0.4735	0.0533	-8.8906	0.0000

Standard errors: MLE

```
                scope=(list(lower=minmodel, upper=model1)),
                direction='backward',
                k=log(nrow(trainData)))
```

Start: AIC=2475.34 medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat

```
      Df Deviance    AIC
```

- age 1 8651.4 2469.3
- indus 1 8657.6 2469.6 8651.3 2475.3
- black 1 8806.5 2476.5
- tax 1 8851.4 2478.6
- crim 1 8862.2 2479.1
- chas 1 8888.7 2480.3
- zn 1 8906.2 2481.1
- rad 1 9015.0 2486.0
- nox 1 9035.7 2486.9
- ptratio 1 9551.0 2509.4
- dis 1 9562.0 2509.9
- rm 1 10123.2 2533.0
- lstat 1 10147.2 2533.9

Step: AIC=2469.33 medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax + ptratio + black + lstat

```
      Df Deviance    AIC
```

- indus 1 8657.6 2463.6 8651.4 2469.3
- black 1 8806.7 2470.5
- tax 1 8851.5 2472.6
- crim 1 8862.2 2473.1
- chas 1 8889.1 2474.3
- zn 1 8912.5 2475.4
- rad 1 9018.1 2480.1
- nox 1 9084.8 2483.1
- ptratio 1 9563.1 2503.9
- dis 1 9617.5 2506.2
- rm 1 10201.3 2530.1

- lstat 1 10397.3 2537.8

Step: AIC=2463.62 medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat

```
      Df Deviance    AIC
```

8657.6 2463.6 - black 1 8810.8 2464.7 - tax 1 8862.7 2467.1 - crim 1 8871.6 2467.5 - chas 1 8910.2 2469.3 - zn 1 8914.0 2469.4 - rad 1 9022.7 2474.3 - nox 1 9098.2 2477.7 - ptratio 1 9567.2 2498.1 - dis 1 9705.3 2503.9 - rm 1 10204.0 2524.2 - lstat 1 10398.9 2531.8

```
# force the 2nd model output from the step
# function to be model4, knitr chooses
# chooses the 1st model with the lowest BIC
# while in markdown the 2nd model with the
# lowest AIC gets picked
model4 <- glm(medv~zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat, data=trainData)
summ(model4, digits=4)
```

Observations	405
Dependent variable	medv
Type	Linear regression

$\chi^2(10)$	24539.4466
Pseudo-R^2 (Cragg-Uhler)	0.7350
Pseudo-R^2 (McFadden)	0.1829
AIC	2423.4619
BIC	2471.5086

	Est.	S.E.	t val.	p
(Intercept)	33.2440	5.9087	5.6263	0.0000
zn	0.0457	0.0147	3.1129	0.0020
chas	3.3841	0.9663	3.5022	0.0005
nox	-17.3366	4.0521	-4.2784	0.0000
rm	4.0298	0.4887	8.2466	0.0000
dis	-1.3889	0.2113	-6.5721	0.0000
rad	0.2338	0.0708	3.3012	0.0011
tax	-0.0115	0.0039	-2.9600	0.0033
ptratio	-0.8906	0.1415	-6.2942	0.0000
black	0.0093	0.0031	3.0135	0.0027
lstat	-0.5031	0.0530	-9.4963	0.0000

Standard errors: MLE

(b) How many variables are in *model4*? Which regression coefficients are significant at the 99% confidence level?

There are 10 variables in model4. They are zn, chas, nox, rm, dis, rad, tax, ptratio, black, and lstat. Al but zn are significant at the 99% confidence level. The intercept is significant at the 99% confidence level as well. The variable zn is significant at the 95% confidence level.

Standard errors: MLE

	Est.	S.E.	t val.	p
(Intercept)	32.7546	5.7046	5.7418	0.0000
zn	0.0367	0.0147	2.4992	0.0129
chas	3.1955	0.9810	3.2575	0.0012
nox	-16.1645	3.9888	-4.0525	0.0001
rm	4.1279	0.4544	9.0848	0.0000
dis	-1.2937	0.2025	-6.3891	0.0000
rad	0.2432	0.0709	3.4310	0.0007
tax	-0.0131	0.0039	-3.3431	0.0009
ptratio	-0.8987	0.1425	-6.3081	0.0000
black	0.0084	0.0029	2.8684	0.0043
lstat	-0.4987	0.0517	-9.6550	0.0000

(c) Perform forward stepwise selection with AIC. Allow the minimum model to be the model with onlyan intercept, and the full model to be *model1*. Display the model summary of your final model. Call it *model5*. Do the variables included in *model5* differ from the variables in *model4*?

```r
set.seed(100)
model5 <- step(model1,
               scope=(list(lower=minmodel, upper=model1)),
               direction='forward')
```

Start: AIC=2419.28 medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat

```r
print_output(summary(model5, digits=4), cex=0.35)
```

```
Call:
glm(formula = medv ~ crim + zn + indus + chas + nox + rm + age + 
    dis + rad + tax + ptratio + black + lstat, data = trainData)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-16.1428   -2.6946   -0.5747    1.9087   27.2846  

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.392e+01  5.928e+00   5.723 2.09e-08 ***
crim        -1.170e-01  3.791e-02  -3.087  0.00217 ** 
zn           5.029e-02  1.482e-02   3.394  0.00076 ***
indus        3.611e-02  6.811e-02   0.530  0.59625    
chas         3.173e+00  9.688e-01   3.275  0.00115 ** 
nox         -1.855e+01  4.450e+00  -4.168 3.79e-05 ***
rm           4.080e+00  5.003e-01   8.156 4.77e-15 ***
age         -4.247e-04  1.512e-02  -0.028  0.97760    
dis         -1.424e+00  2.220e-01  -6.415 4.06e-10 ***
rad          3.062e-01  7.553e-02   4.054 6.08e-05 ***
tax         -1.262e-02  4.198e-03  -3.007  0.00281 ** 
ptratio     -9.102e-01  1.427e-01  -6.376 5.12e-10 ***
black        8.180e-03  3.089e-03   2.648  0.00843 ** 
lstat       -4.754e-01  5.782e-02  -8.222 2.98e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 22.1262)

    Null deviance: 33411.0  on 404  degrees of freedom
Residual deviance:  8651.3  on 391  degrees of freedom
AIC: 2419.3

Number of Fisher Scoring iterations: 2
```

Model 5 includes all 13 predictor variables, while Model 4 only had 10.

(d) Compare the adjusted R^2, Mallow's Cp, AICs and BICs of the full model(*model1*), the model found in Question 2 (*model3*), and the model found using backward selection with BIC (*model4*). Which model is preferred based on these criteria and why?

Adjusted R^2 for models 1, 3, and 4.

```
modelsrsq <- data.frame(
  rsq(model1, adj=TRUE),
  rsq(model3, adj=TRUE),
  rsq(model4, adj=TRUE))
names(modelsrsq)[1] <- "Adj Rsq Model 1"
names(modelsrsq)[2] <- "Adj Rsq Model 3"
names(modelsrsq)[3] <- "Adj Rsq Model 4"
xtable(modelsrsq, digits=4)
```

	Adj Rsq Model 1	Adj Rsq Model 3	Adj Rsq Model 4
1	0.7325	0.7336	0.7277

AIC for models 1, 3, and 4.

```
set.seed(100)
modelsaic <-data.frame(
  AIC(model1),
  AIC(model3),
  AIC(model4))
names(modelsaic)[1] <- "AIC Model 1"
names(modelsaic)[2] <- "AIC Model 3"
names(modelsaic)[3] <- "AIC Model 4"
xtable(modelsaic)
```

	AIC Model 1	AIC Model 3	AIC Model 4
1	2419.28	2415.57	2423.46

BIC for models 1, 3, and 4.

```
set.seed(100)
modelsbic <- data.frame(
  AIC(model1, k=log(nrow(trainData))),
  AIC(model3, k=log(nrow(trainData))),
  AIC(model4, k=log(nrow(trainData))))
names(modelsbic)[1] <- "BIC Model 1"
names(modelsbic)[2] <- "BIC Model 3"
names(modelsbic)[3] <- "BIC Model 4"
xtable(modelsbic)
```

	BIC Model 1	BIC Model 3	BIC Model 4
1	2479.34	2467.62	2471.51

Mallow's CP for models 1, 3, and 4.

```
set.seed(100)
model1.rse <- sqrt(deviance(model1)/df.residual(model1))
model1.cp <-Cp(model1, S2=model1.rse)
model3.rse <- sqrt(deviance(model3)/df.residual(model3))
```

```r
model3.cp <-Cp(model3, S2=model3.rse)
model4.rse <- sqrt(deviance(model4)/df.residual(model4))
model4.cp <-Cp(model4, S2=model4.rse)
modelscp <-data.frame(model1.cp, model3.cp, model4.cp)
names(modelscp)[1] <- "Cp Model 1"
names(modelscp)[2] <- "Cp Model 3"
names(modelscp)[3] <- "Cp Model 4"
xtable(modelscp)
```

	Cp Model 1	Cp Model 3	Cp Model 4
1	1462.21	1463.57	1486.60

All 3 Models have the similar adjusted R^2 @ approx 0.73. Model 3 has the lowest AIC @ 2411. Models 3 and 4 have the lowest BIC @ approx 2463. Model 1 has the lowest Mallow's Cp @ 1452. Model 3 is preferred because is has the lowest AIC and BIC, and is a less complex model with 10 variables as opposed to 13 in model 4.

Question 4: Ridge Regression

(a) Perform ridge regression on the training set. Use cv.glmnet() to find the lambda value that minimizes the cross-validation error using 10 fold CV.

```r
set.seed(100)
Xpred = as.matrix(trainData[,c(1:13)])
Y = as.matrix(trainData$medv)
model5.cv <-cv.glmnet(Xpred, Y, alpha=0, nfolds=10)
model5.cv$lambda.min
```

[1] 0.6656113

(b) List the value of coefficients at the optimum lambda value.

```r
set.seed(100)
model5 = glmnet(Xpred, Y, alpha = 0, nlambda = 100)
print_output(coef(model5,s=model5.cv$lambda.min))
```

```
14 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept)  25.264834855
crim         -0.097461503
zn            0.035629976
indus        -0.022614863
chas          3.415609209
nox         -12.107091834
rm            4.305318334
age          -0.006479400
dis          -1.063728585
rad           0.150620785
tax          -0.005941809
ptratio      -0.823954069
black         0.008026823
lstat        -0.422497214
```

(c) How many variables were selected? Give an explanation for this number.

All variables are selected. Ridge regression does not set the values of the coefficents to 0.

Question 5: Lasso Regression

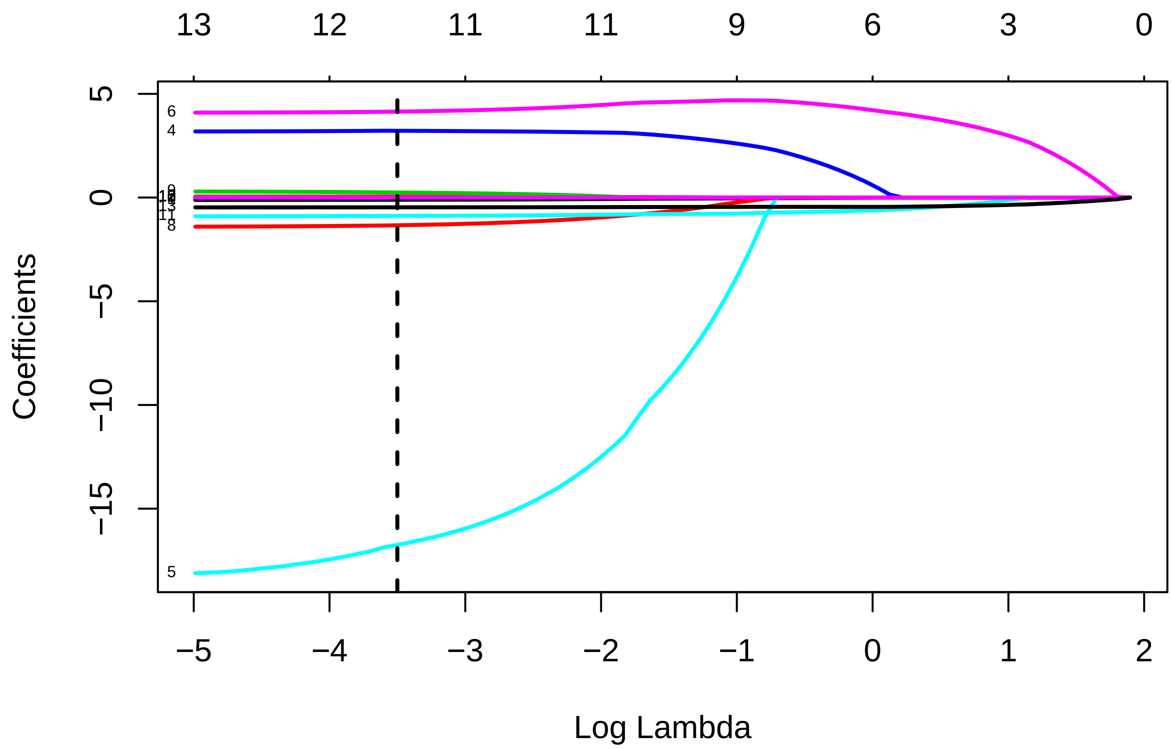
(a) Perform lasso regression on the training set.Use cv.glmnet() to find the lambda value that minimizes the cross-validation error using 10 fold CV.

```r
set.seed(100)
model6.cv <-cv.glmnet(Xpred, Y, alpha=1, nfolds=10)
model6.cv$lambda.min
```

[1] 0.03018466

(b) Plot the regression coefficient path.

```r
set.seed(100)
model6 = glmnet(Xpred, Y, alpha = 1, nlambda = 100)
plot(model6,xvar="lambda",label=TRUE,lwd=2)
abline(v=log(model6.cv$lambda.min),col='black',lty = 2,lwd=2)
```



(c) How many variables were selected? Which are they?

```
set.seed(100)
print_output(coef(model6,s=model6.cv$lambda.min))
```

```
14 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)  31.503982486
crim         -0.107461313
zn            0.044801690
indus         .
chas          3.219103798
nox         -16.744510441
rm            4.143063666
age           .
dis          -1.337005901
rad           0.243849796
tax          -0.009515367
ptratio      -0.884114614
black         0.007832232
lstat        -0.470896025
```

All 13 predictor variables were selected.

Question 6: Elastic Net

(a) Perform elastic net regression on the training set. Use cv.glmnet() to find the lambda value that minimizes the cross-validation error using 10 fold CV. Give equal weight to both penalties.

```r
set.seed(100)
model7.cv <-cv.glmnet(Xpred, Y, alpha=0.5, nfolds=10)
model7.cv$lambda.min
```

[1] 0.05500628

(b) List the coefficient values at the optimal lambda. How many variables were selected? How do these variables compare to those from Lasso in Question 5?

```r
set.seed(100)
model7 = glmnet(Xpred, Y, alpha = 0.5, nlambda = 100)
print_output(coef(model7,s=model7.cv$lambda.min))
```

```
14 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept)  31.210394400
crim         -0.107001236
zn            0.044578908
indus         .
chas          3.229422023
nox         -16.554250448
rm            4.150785289
age           .
dis          -1.324334356
rad           0.239754073
tax          -0.009373441
ptratio      -0.881666980
black         0.007855535
lstat        -0.469371448
```

Question 7: Model comparison

(a) Predict *medv* for each of the rows in the test data using the full model, and the models found using backward stepwise regression with BIC, ridge regression, lasso regression, and elastic net.

```r
set.seed(100)
predict1 <- predict(model1, testData)
predict4 <- predict(model4, testData)
predict5 <- predict(model5, s=model5.cv$lambda.min, newx=as.matrix(testData[, c(1:13)]))
predict6 <- predict(model6, s=model6.cv$lambda.min, newx=as.matrix(testData[, c(1:13)]))
predict7 <- predict(model7, s=model7.cv$lambda.min, newx=as.matrix(testData[, c(1:13)]))
```

(b) Compare the predictions using mean squared prediction error. Which model performed the best?

```r
mspe1 <- mean(testData$medv - predict1)^2
mspe4 <- mean(testData$medv - predict4)^2
mspe5 <- mean(testData$medv - predict5)^2
mspe6 <- mean(testData$medv - predict6)^2
mspe7 <- mean(testData$medv - predict7)^2
cat("MSPE for model 1:", mspe1, " 4: ", mspe4, " 5: ", mspe5, " 6: ", mspe6, " 7: ", mspe7 )
```

MSPE for model 1: 0.05279967 4: 0.06864134 5: 0.05601331 6: 0.05603335 7: 0.05600613

Model 5 using Ridge Regression had the lowest MSPE @ 0.0005.

(c) Provide a table listing each method described in Question 7a and the variables selected by each method (see Unit 5.2.3 for an example). Which variables were selected consistently?

	Bkwd Step	Ridge	Lasso	Elastic Net
crim		X	X	X
zn	X	X	X	X
indus		X	X	X
chas	X	X	X	X
nox	X	X	X	X
rm	X	X	X	X
age		X	X	X
dis	X	X	X	X
rad	X	X	X	X
tax	X	X	X	X
ptratio	X	X	X	X
black	X	X	X	X
lstat	X	X	X	X

All predictor variables except crim, indus and age were selected consistently. Those 3 variables were excluded using backwards stepwise regression.