Project 1: Comparison of Model Selection Methods for Boston Dataset

submitted by Gap Kim (guk90@psu.edu (mailto:guk90@psu.edu))

INTRODUCTION

The performance of four selections methods, Best subsets, Ridge regression, Lasso regression, and Manual selection, have been compared using the 'Boston' dataset included in the MASS package. The model objective is to determine the relationship between per capita crime rate, 'crim', on other 13 predictors where 2 are categorical variables with 2 levels and 9 levels each. First, an exploratory data analysis is performed to analyze the distribution of variables and to investigate preliminary relationship between 'crim' and predictors. Due to high skewedness of variables leading to violations of linear regression assumptions, transformed variables are used throughout the model selection process. The Best subsets elminated 6 predictors and had the highest prediction accuracy followed by Manual selection method. The increase in model bias through Ridge or Lasso regression did not result in significant improvement in prediction accuracy for the transformed variables.

EXPLORATORY DATA ANALYSIS

The dataset, Boston, from MASS library package contains housing values and other information about Boston suburbs. The dataset contains 506 observations for the following 14 variables:

```
crim: per capita crime rate by town
zn: proportion of residential land zoned for lots over 25,000 sq.ft.
indus: proportion of non-retail business acres per town
chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
nox: nitrogen oxides concentration (parts per 10 million)
rm: average number of rooms per dwelling
age: proportion of owner-occupied units built prior to 1940
dis: weighted mean of distances to five Boston employment centres
rad: index of accessibility to radial highways
tax: full-value property-tax rate per \$10,000
ptratio: pupil-teacher ratio by town
black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
lstat: lower status of the population (percent)
medv: median value of owner-occupied homes in \$1000s
```

Among 14 variables, 'chas' is a binary categorical variable with following reponses.

Table 1: Summary of categorical variable, 'chas'

Bounds river	471
Otherwise	35

During the initial assessment of variables, 'rad' was identified to have discrete integer values as shown in Figure 1. Since the index does not seem to be discrete quatitative, it is treated as categorical variable having 9 levels in this study.

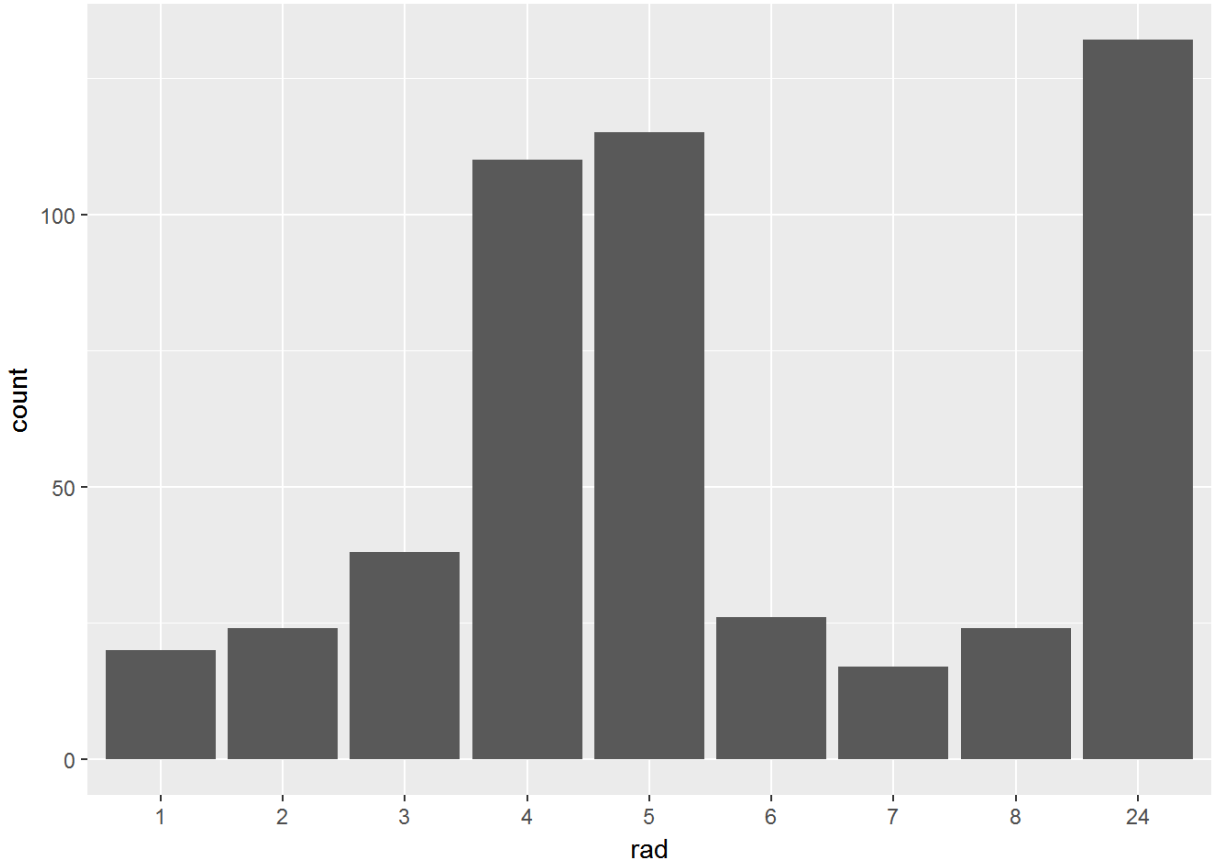

Figure 1: Histogram of variable 'rad'

To understand the distribution of the variables, boxplots have been provided in Figure 2. The variables, 'crim', 'zn' and 'black', are highly skewed with outliers.

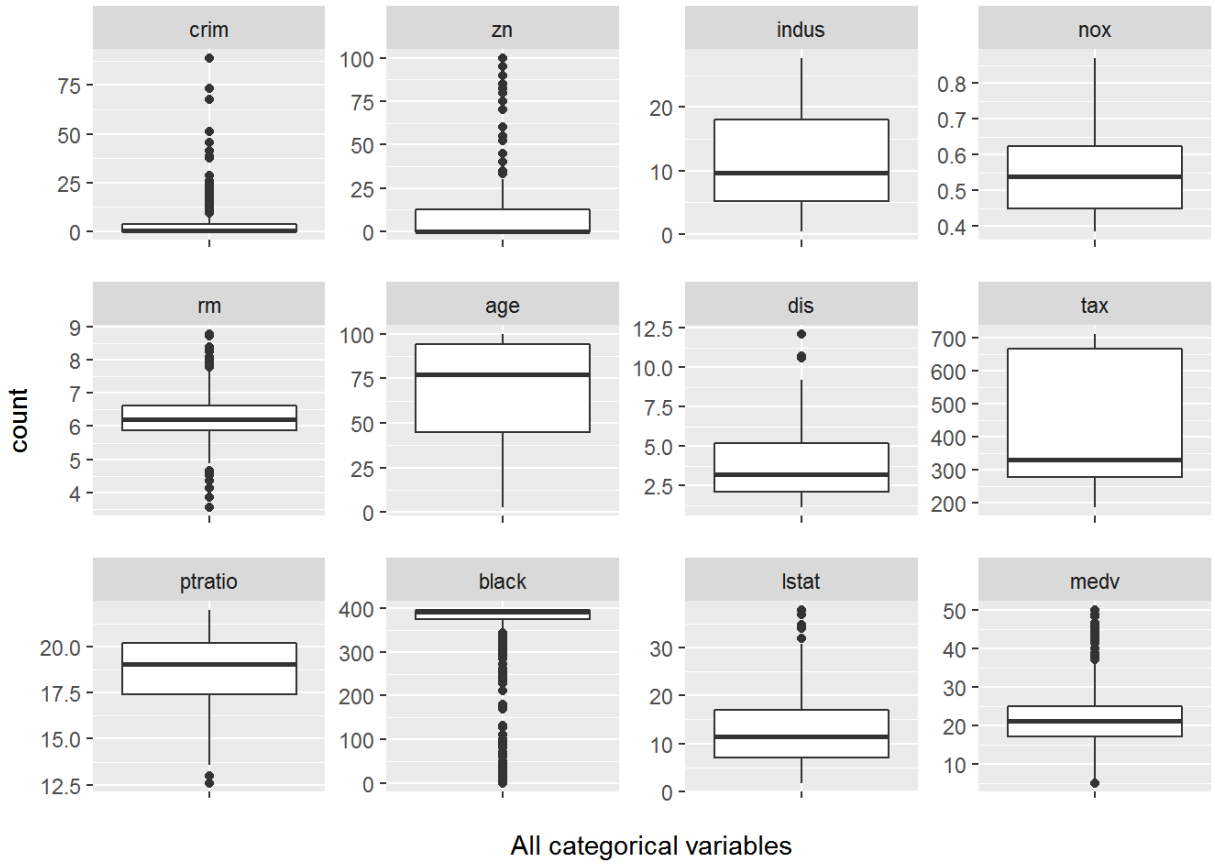

Figure 2: Boxplots of all 12 quantitative variables

A histogram of reponse variable, ‘crim’, is plotted on logarithmic x scale in Figure. 3 along with summary statistics in Table 2. A large descrepancy between the median and mean indicates a severely skewed distribution for the variable ‘crim’.

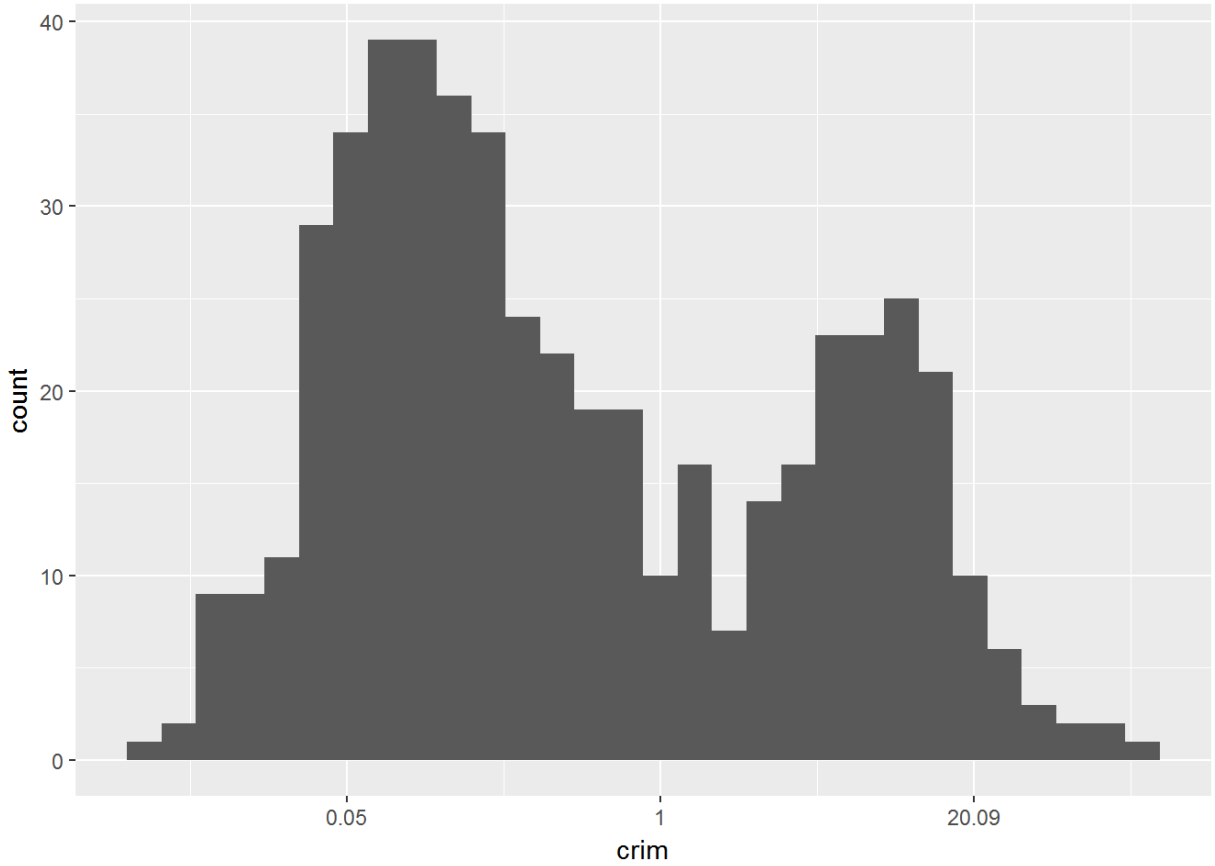

Figure 3: Histogram of variable ‘crim’

Table 2: Summary statistics of ‘crim’

Min. : 0.01
1st Qu.: 0.08
Median : 0.26
Mean : 3.61
3rd Qu.: 3.68
Max. :88.98

To get an initial look at the bivariate relationship, pairs scatterplots and correlation coefficients have been generated as shown Figure 4. Strong correlations are found for dis~indus, tax~indus, nox~indus, dis~nox, dis~age, and lstat~medv, which have correlation coefficients higher than 0.7.

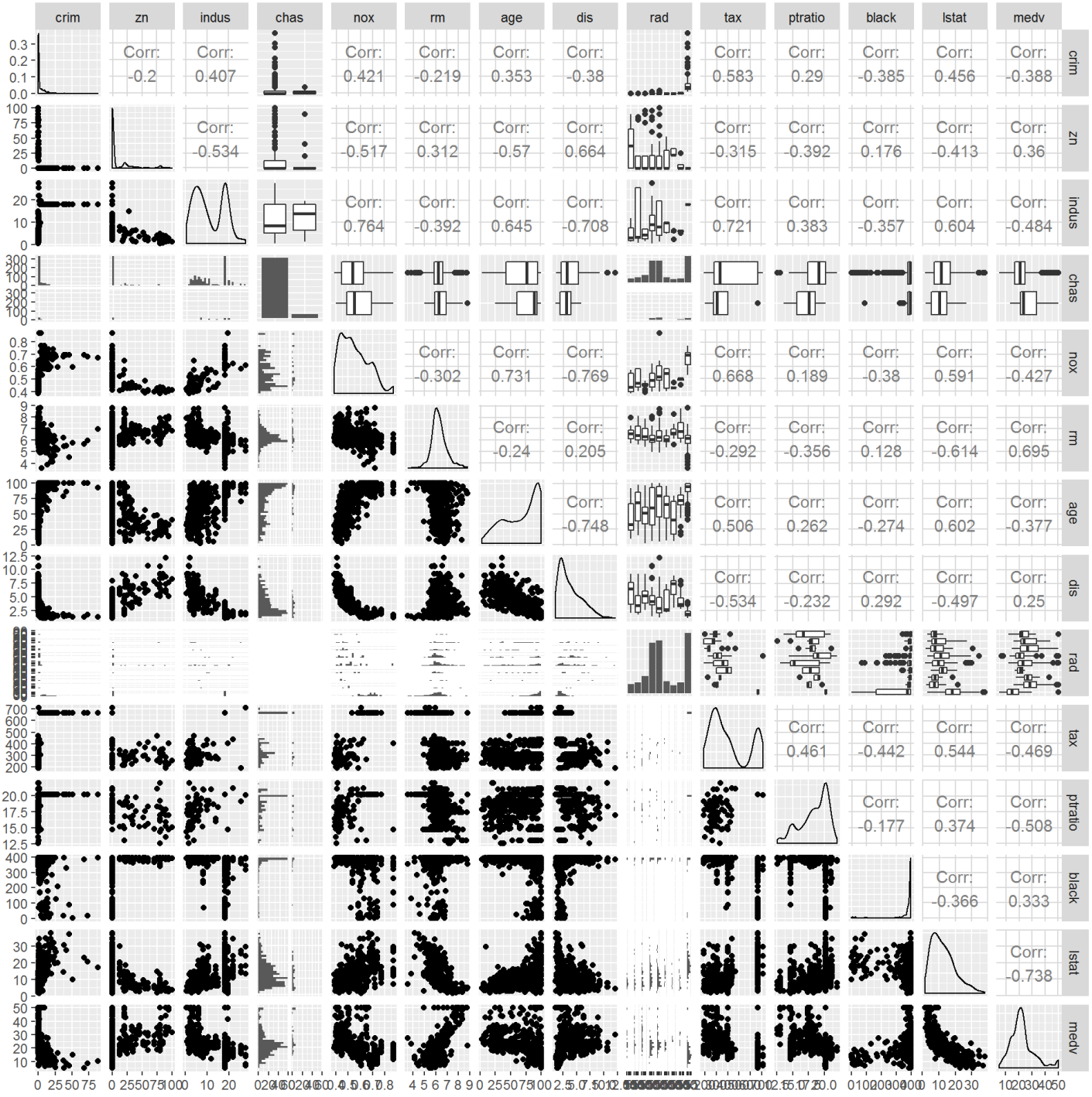

Figure 4: Scatterplot and correlation plot matrix of the dataset

Bivarate scatterplots of response variable ‘crim’ versus other variables are shown in Figure 5. An interesting attribute of the scatterplots are the localized, high spikes for variables, ‘zn’, ‘indus’, ‘rad’, ‘tax’, and ‘ptratio’.

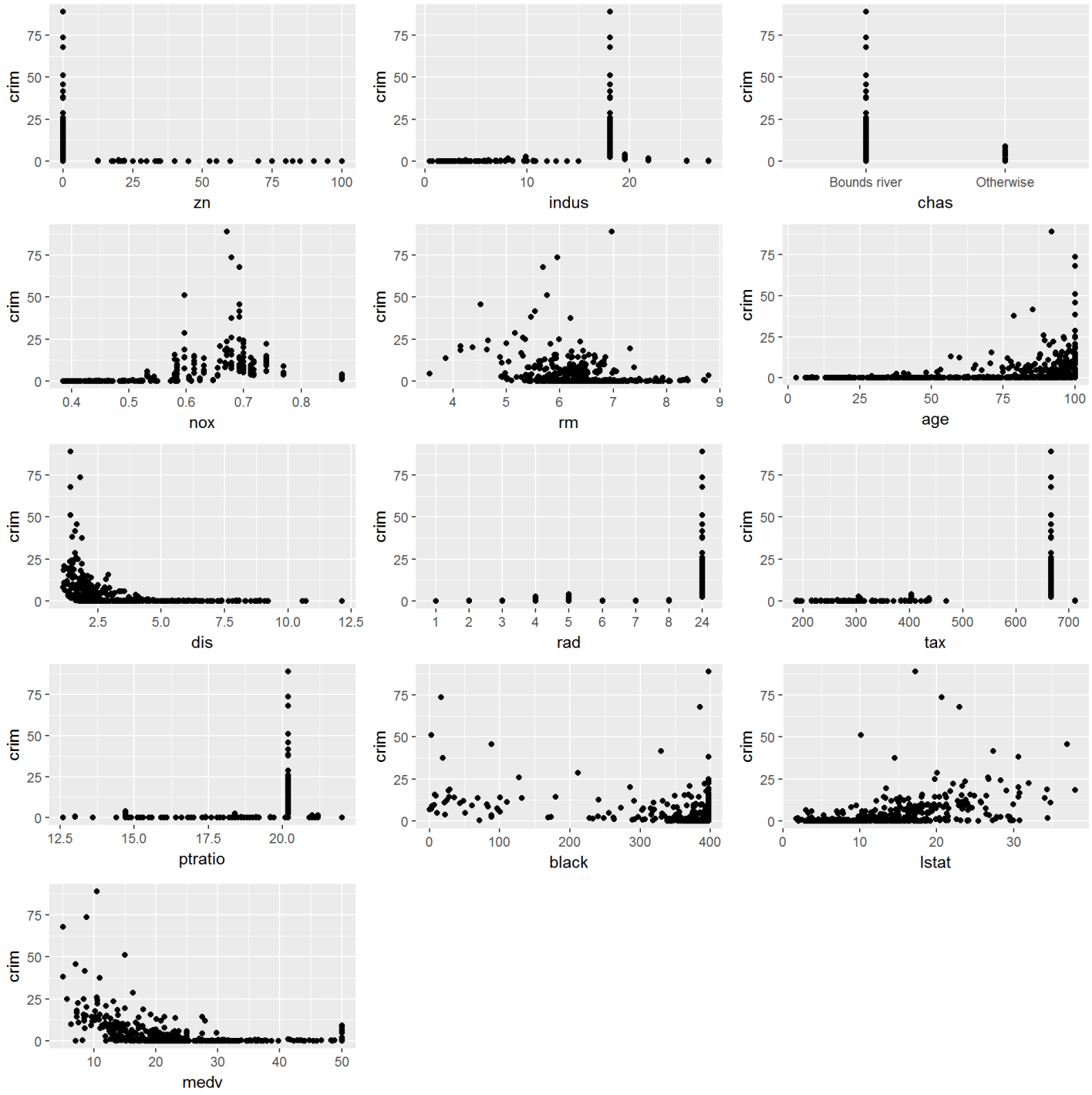

Figure 5: Scatterplots of 'crim' versus other variables

In Figure 5, much of the data are clustered around the lower values of y ('crim'), and thus, logarithmic transformation of y-axis may help to spread out the data. It is apparent that the bivariate plots show more difinitive patterns with 'crim' on a logarithmic scale as shown in Figure 6.

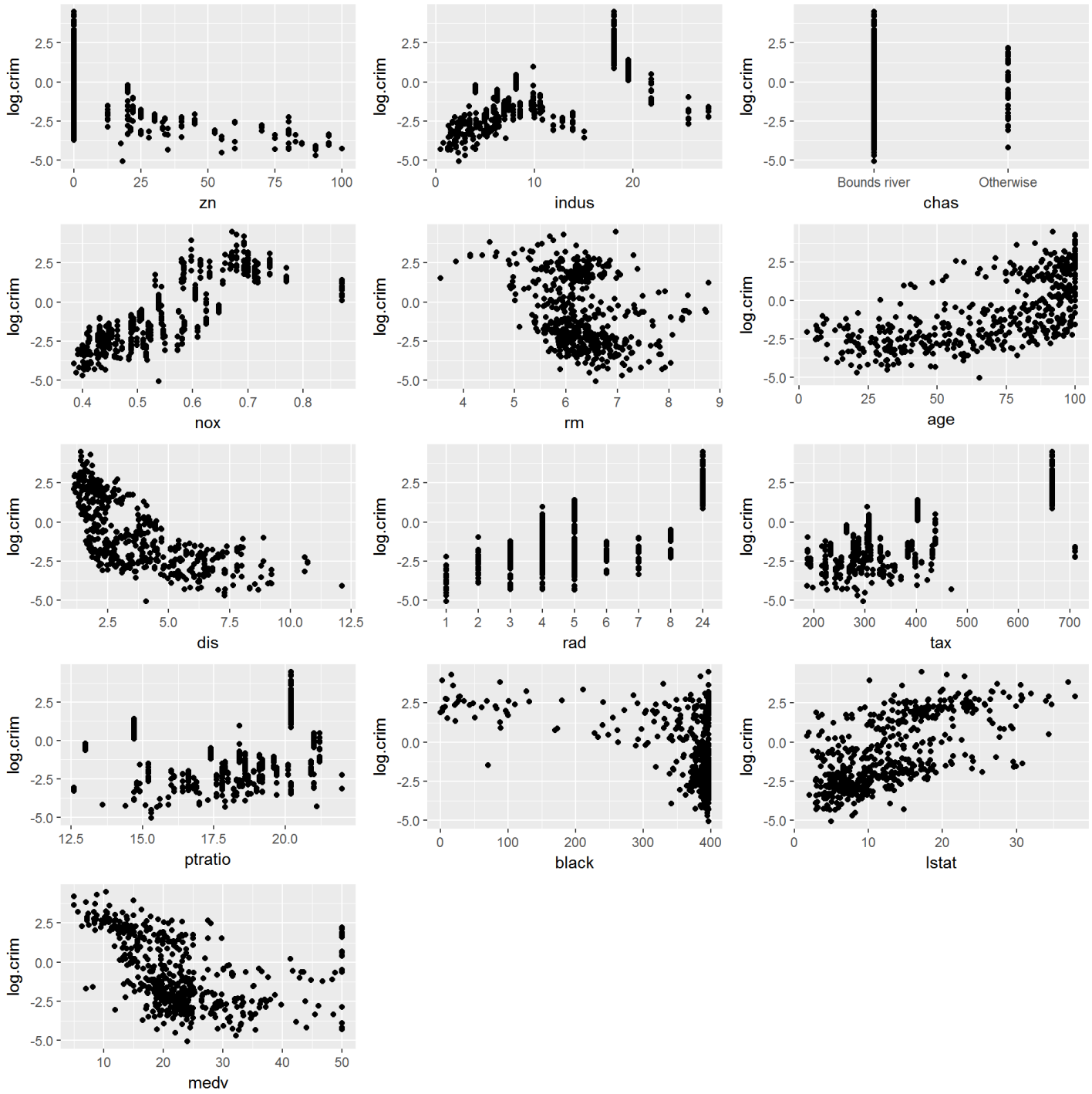

Figure 6: Scatterplot of 'log.crim' versus other variables

It is also observed that the variable 'black' has much of the data clustered around high values. This is not only observed in Figure 6, but as well as in Figure 4. To spread out the data around high values of 'black', an exponential function has been applied as the following:

$$\exp.\text{black} = \frac{1}{100} \times e^{(10 \times \frac{black}{max(black)})}$$

It can be observed from Figure 7 that the exponential scale helps to spread out the high values of 'black' although some clustering is still noted at very low and high values of 'crim'.

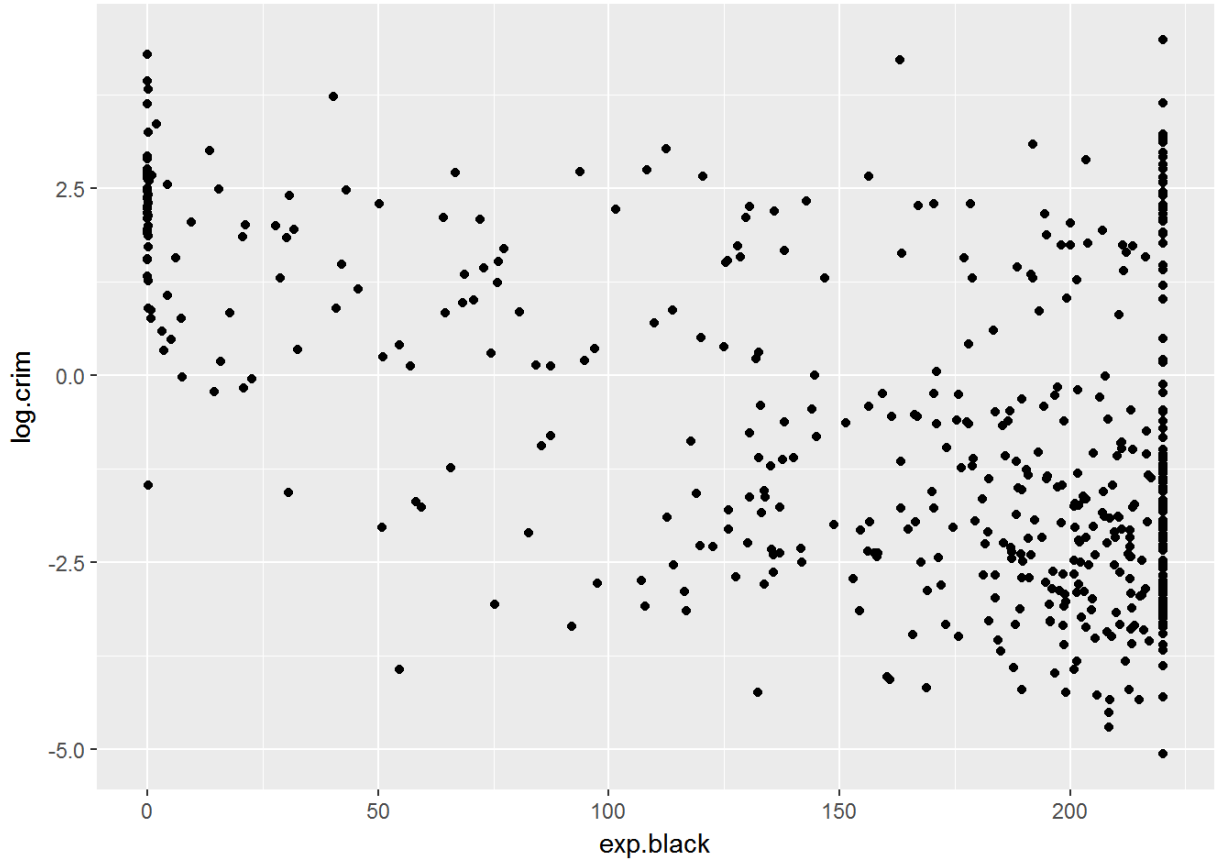

Figure 7: Scatterplot of 'log.crim' versus 'exp.black'

LINEAR REGRESSION AND MANUAL MODEL SELECTION

Full Model with Original Variables

The objective is to uncover the relationship between response 'crim' and other predictors given in the Boston dataset. First, a full model is analyzed where 'crim' is regressed on all 13 predictors. The results show that the statistically significant predictors are: 'zn', 'nox', 'dis', factor level 'rad24', and 'medv'. The model has $R^2 = 46.0415\%$, and overall F-test confirms that the model is significant.

```
Call:
lm(formula = crim ~ ., data = mydata)

Residuals:
   Min     1Q Median     3Q    Max
 -9.91  -1.83  -0.27   0.93  74.82

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    21.28668    7.72082    2.76  0.00605 **
zn              0.03874    0.01966    1.97  0.04931 *
indus          -0.07884    0.08737   -0.90  0.36730
chasOtherwise  -0.75071    1.19561   -0.63  0.53038
nox           -10.81204    5.44114   -1.99  0.04747 *
rm              0.39765    0.62184    0.64  0.52281
age             0.00190    0.01817    0.10  0.91681
dis            -1.01633    0.29016   -3.50  0.00050 ***
rad2           -0.70404    2.03142   -0.35  0.72906
rad3            0.55521    1.85713    0.30  0.76510
rad4            0.20719    1.63888    0.13  0.89945
rad5            0.49125    1.66862    0.29  0.76858
rad6           -0.92578    2.01193   -0.46  0.64562
rad7            1.61448    2.17836    0.74  0.45897
rad8            1.60824    2.06982    0.78  0.43754
rad24          12.04502    2.44013    4.94  1.1e-06 ***
tax            -0.00312    0.00538   -0.58  0.56136
ptratio        -0.35118    0.20691   -1.70  0.09028 .
black          -0.00703    0.00369   -1.91  0.05708 .
lstat           0.12199    0.07680    1.59  0.11282
medv           -0.20533    0.06167   -3.33  0.00094 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.45 on 485 degrees of freedom
Multiple R-squared:  0.46,  Adjusted R-squared:  0.438
F-statistic: 20.7 on 20 and 485 DF,  p-value: <2e-16
```

Linear regression assumptions are assessed from Figure 8 and Figure 9. The residuals versus fits plot shows a slight curvature and increasing variance. The model may be potentially violating the assumptions of ‘linearity’ and ‘equal variance of errors’. In addition, the normal Q-Q plot of residuals shows that the assumption of ‘normally distributed errors’ may be violated.

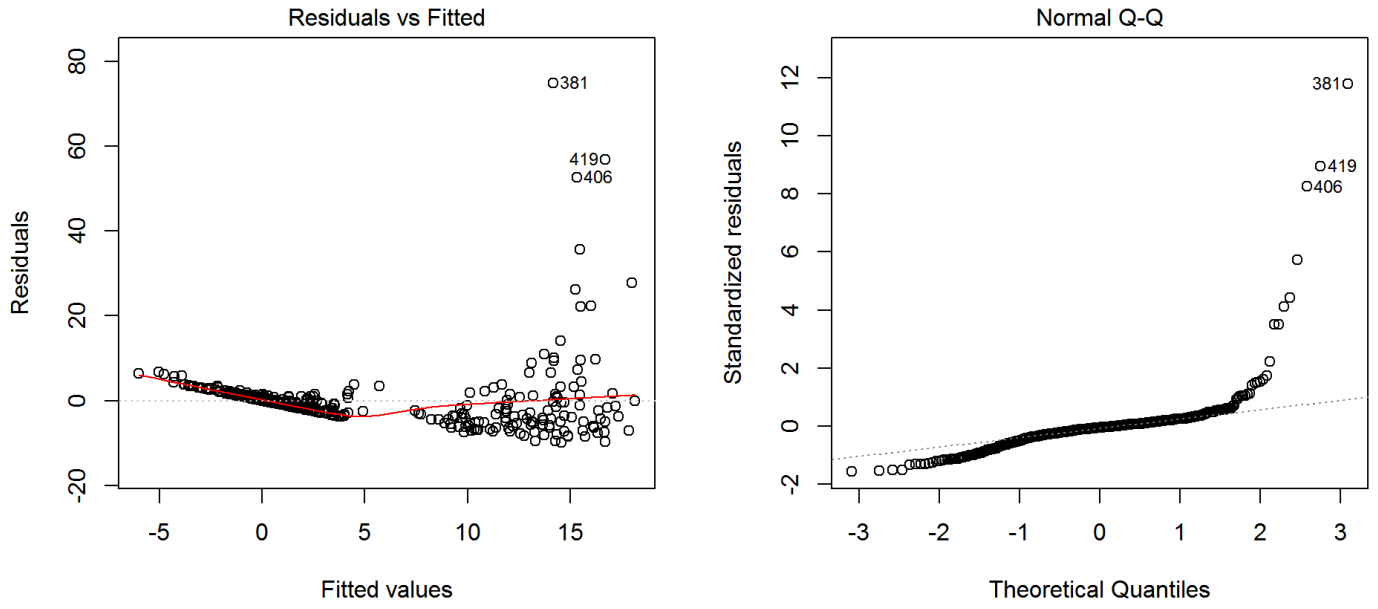

Figure 8: Residual vs. fits plot (left), and Normal Q-Q plot of residuals(right)

Standard residuals versus fits plot shows that there are outliers beyond 3 standard deviation of residuals (indicated by the blue line in Figure 9(left)). The right side of Figure 9 provides standard residuals versus leverage with contours of Cook's distance. Although no points are considered influential with Cook's distance under 0.5, potential outliers and high leverage points are observed.

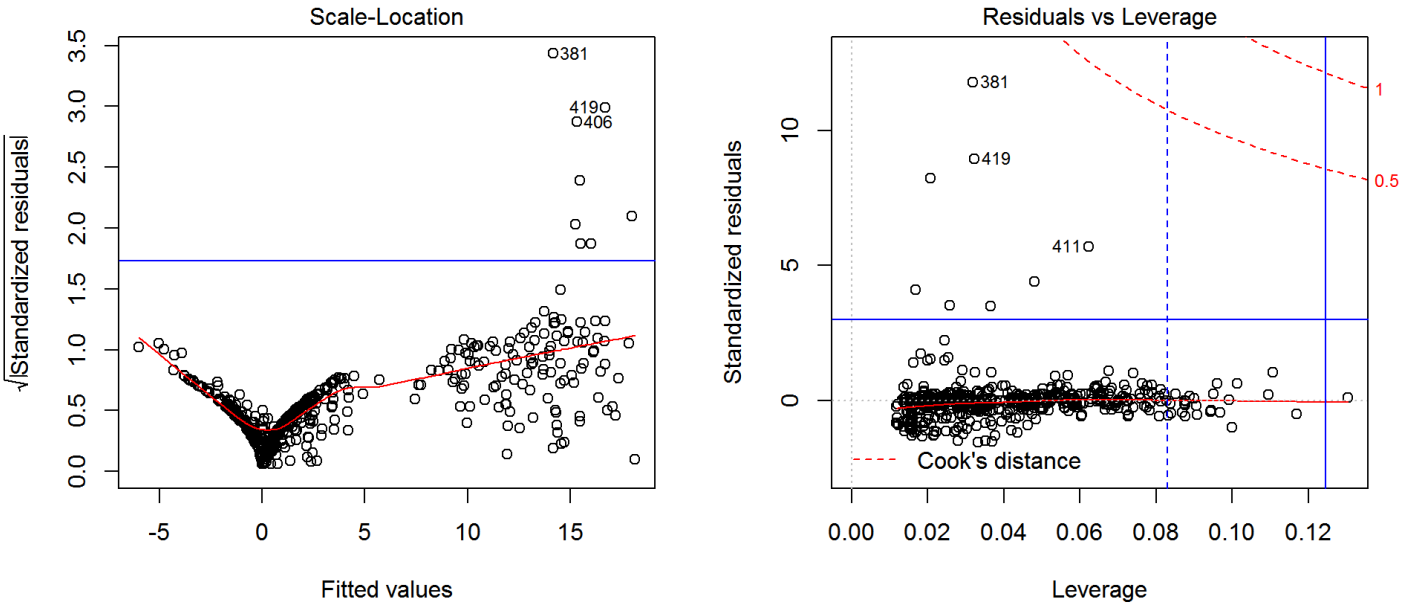

Figure 9: Standardized residuals vs. fits plot (left), and Standard residuals vs. leverage with contours of Cook's distance

Multicollinearity can be checked with Generalized Variance Inflation Factor (GVIF) when variables have multilevel factors with more than one degree of freedom. In Table 3, the right column provides adjusted GVIF accounting for multiple degrees of freedom. Applying $\sqrt{5} = 2.24$ as high multicollinearity criterion, predictor 'tax' may be causing multicollinearity.

Table 3: VIFs from full model with original variables

	GVIF	Df	GVIF^(1/(2*Df))
zn	2.553	1	1.598
indus	4.364	1	2.089
chas	1.120	1	1.058
nox	4.830	1	2.198
rm	2.319	1	1.523
age	3.178	1	1.783
dis	4.535	1	2.130
rad	18.410	8	1.200
tax	9.973	1	3.158
ptratio	2.438	1	1.561
black	1.377	1	1.173
lstat	3.654	1	1.911
medv	3.909	1	1.977

Full Model with Transformed Variables

Transformation of response variable, ‘crim’, may improve the model assumptions that seem to be violating as discussed in Figure 8. Therefore, a logarithmic transformation of ‘crim’ and an exponential transformation of ‘black’ as discussed in the exploratory data analysis section have been employed in the subsequent sections:

$$\log.\text{crim} = log(crim)$$
$$\exp.\text{black} = \frac{1}{100} \times e^{(10 \times \frac{black}{max(black)})}$$

The regression results with tranformed variables are shown below. It is evident that the transformation significantly improved the model accuracy to $R^2 = 89.171\%$. Moreover, there are more predictors that have been identified as statistically significant in the tranformed full model.

```
Call:
lm(formula = log.crim ~ ., data = mydata3)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2339 -0.5003 -0.0467  0.4499  2.5486

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.161564   0.858306   -3.68  0.00026 ***
zn            -0.010955   0.002213   -4.95  1.0e-06 ***
indus          0.007585   0.009871    0.77  0.44260
chasOtherwise -0.153417   0.134596   -1.14  0.25492
nox            3.761995   0.614544    6.12  1.9e-09 ***
rm            -0.019904   0.069176   -0.29  0.77368
age            0.005949   0.002042    2.91  0.00374 **
dis           -0.042284   0.032676   -1.29  0.19626
rad2           0.143025   0.228785    0.63  0.53217
rad3           0.666981   0.209068    3.19  0.00151 **
rad4           1.294473   0.184539    7.01  7.8e-12 ***
rad5           0.927583   0.187825    4.94  1.1e-06 ***
rad6           0.734747   0.226574    3.24  0.00126 **
rad7           1.467611   0.245315    5.98  4.3e-09 ***
rad8           1.739149   0.233313    7.45  4.2e-13 ***
rad24          3.827114   0.273499   13.99  < 2e-16 ***
tax           -0.000121   0.000607   -0.20  0.84161
ptratio       -0.078898   0.023374   -3.38  0.00080 ***
lstat          0.027582   0.008636    3.19  0.00149 **
medv           0.003270   0.006843    0.48  0.63299
exp.black     -0.002586   0.000512   -5.06  6.1e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.726 on 485 degrees of freedom
Multiple R-squared:  0.892, Adjusted R-squared:  0.887
F-statistic:  200 on 20 and 485 DF,  p-value: <2e-16
```

Model Selection by Hypothesis Testing

We want to test whether all the non-significant variables found in the previous section could be removed from the model. Thus, following hypothesis is formulated:

$$H_0 : \beta_{indus} = \beta_{chas} = \beta_{rm} = \beta_{dis} = \beta_{tax} = \beta_{medv} = 0$$
$$H_A : At\ least\ one\ \beta_j \neq 0\ (j = indus, chas, rm, dis, tax, medv)$$

Since predictor ‘rad’ was treated as a categorical variable, it was considered significant as a whole although a sigle level, ‘rad2’, was insignificant. To test the hypothesis, a linear regression has been sequentially ordered resulting in ANOVA with sequential sum of squares as shown in Table 4.

Table 4: ANOVA result of transformed full model
(Type I, Sequential sum of squares)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
zn	1	631.1865	631.1865	1197.5339	0.0000
nox	1	875.6491	875.6491	1661.3467	0.0000
rad	8	551.0217	68.8777	130.6799	0.0000

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
ptratio	1	5.2721	5.2721	10.0025	0.0017
exp.black	1	17.0841	17.0841	32.4132	0.0000
age	1	13.2537	13.2537	25.1459	0.0000
lstat	1	9.0559	9.0559	17.1816	0.0000
chas	1	0.4740	0.4740	0.8994	0.3434
indus	1	0.5893	0.5893	1.1181	0.2908
dis	1	1.2259	1.2259	2.3259	0.1279
rm	1	0.0080	0.0080	0.0151	0.9021
tax	1	0.0325	0.0325	0.0617	0.8039
medv	1	0.1203	0.1203	0.2283	0.6330
Residuals	485	255.6299	0.5271	NA	NA

Therefore, F-statistic can be calculated as:

$$F^* = \frac{(0.4740 + 0.5893 + 1.226 + 0.007984 + 0.03252 + 0.1203)}{6} \div 0.5271 = 0.7747$$

With $Fcdf(0.7747, 6, 485) = 0.41$, this results in p-value of 0.59. Thus, we failed to reject the null hypothesis and conclude that predictors, ‘indus’, ‘chas’, ‘rm’, ‘dis’, ‘tax’, and ‘medv’, are insignificant in a model that already contains predictors, ‘zn’, ‘nox’, ‘age’, ‘rad’, ‘ptratios’, ‘exp.black’, and ‘lstat’.

Therefore, the model is fitted with only the significant predictors, ‘zn’, ‘nox’, ‘age’, ‘rad’, ‘ptratios’, ‘exp.black’, and ‘lstat’. The results are shown below with $R^2 = 89.0672\%$, which is a significant improvement over the original full model of $R^2 = 46.0415\%$. In fact, comparison with transformed full model of $R^2 = 89.171\%$, the reduced model only show very little decrease in the R^2 value even after dropping 6 predictors.

```
Call:
lm(formula = log.crim ~ zn + nox + rad + ptratio + exp.black +
    age + lstat, data = mydata3)

Residuals:
   Min     1Q Median     3Q    Max
-2.259 -0.520 -0.038  0.461  2.545

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.589252   0.551488   -6.51  1.9e-10 ***
zn          -0.012551   0.001916   -6.55  1.5e-10 ***
nox          4.166806   0.527027    7.91  1.8e-14 ***
rad2         0.220887   0.223387    0.99  0.32324
rad3         0.683849   0.204824    3.34  0.00091 ***
rad4         1.332142   0.181605    7.34  9.2e-13 ***
rad5         0.946647   0.185770    5.10  5.0e-07 ***
rad6         0.767149   0.220354    3.48  0.00054 ***
rad7         1.455679   0.241189    6.04  3.1e-09 ***
rad8         1.715180   0.227199    7.55  2.1e-13 ***
rad24        3.859345   0.200916   19.21  < 2e-16 ***
ptratio     -0.081064   0.021638   -3.75  0.00020 ***
exp.black   -0.002595   0.000504   -5.14  3.9e-07 ***
age          0.006885   0.001889    3.65  0.00030 ***
lstat        0.026381   0.006356    4.15  3.9e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.725 on 491 degrees of freedom
Multiple R-squared:  0.891, Adjusted R-squared:  0.888
F-statistic:  286 on 14 and 491 DF,  p-value: <2e-16
```

The diagnostic plots are shown in Figure 10, which indicates much improved conformance to linear regression assumptions. The residuals versus fits plot shows that assumptions of ‘linearity’ and ‘equal variance of error terms’ are not violated. Comparing with Figure 9, the transformed and reduced model has fewer outliers. Moreover, the single high leverage point has moved below the 3 times the average leverage due to the transformation and model reduction. The two outliers are well under Cook’s distance of 0.5, and thus, are not considered influential points.

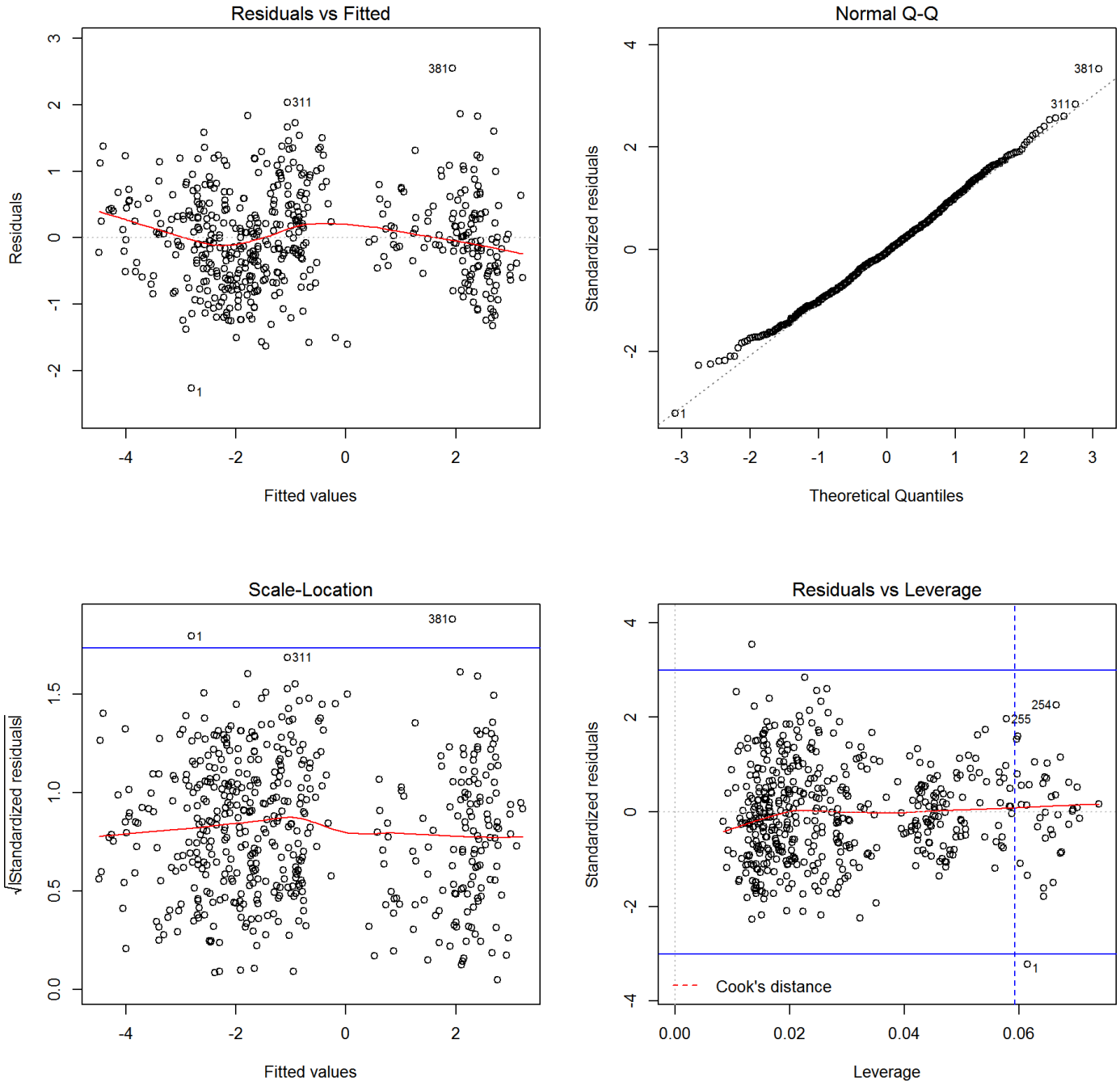

Figure 10: Diagnostic plots for transformed and reduce model

Finally, we check for multicollinearity by calculating GVIFs of the reduced transformed model, which is given in Table 5. The adjusted GVIFs are all under $\sqrt{5} = 2.24$, and thus, multicollinearity is not a concern for the model.

Table 5: VIFs from reduced transformed model

	GVIF	Df	GVIF^(1/(2*Df))
zn	1.919	1	1.385
nox	3.583	1	1.893
rad	3.924	8	1.089
ptratio	2.108	1	1.452
exp.black	1.297	1	1.139
age	2.715	1	1.648
lstat	1.979	1	1.407

BEST SUBSETS REGRESSION

Best subset regression was performed using the transformed variables, log(crim) and exp(black), as shown in the following equation:

$$log(crim) = zn + indus + chas(2\ levels) + nox + rm + age + dis + rad(9\ levels) + tax + ptratio + lstat + medv + exp(black)$$

To find optimal number of predictors, a cross validation approach was used with 10 folds. Both training and test MSE were calculated and plotted in Figure 11 where the smallest test MSE was observed at 14 predictors. The training MSE continually decreased as more predictors are included while the test MSE had a minimum value where the model bias and variance were balanced at 14 predictors.

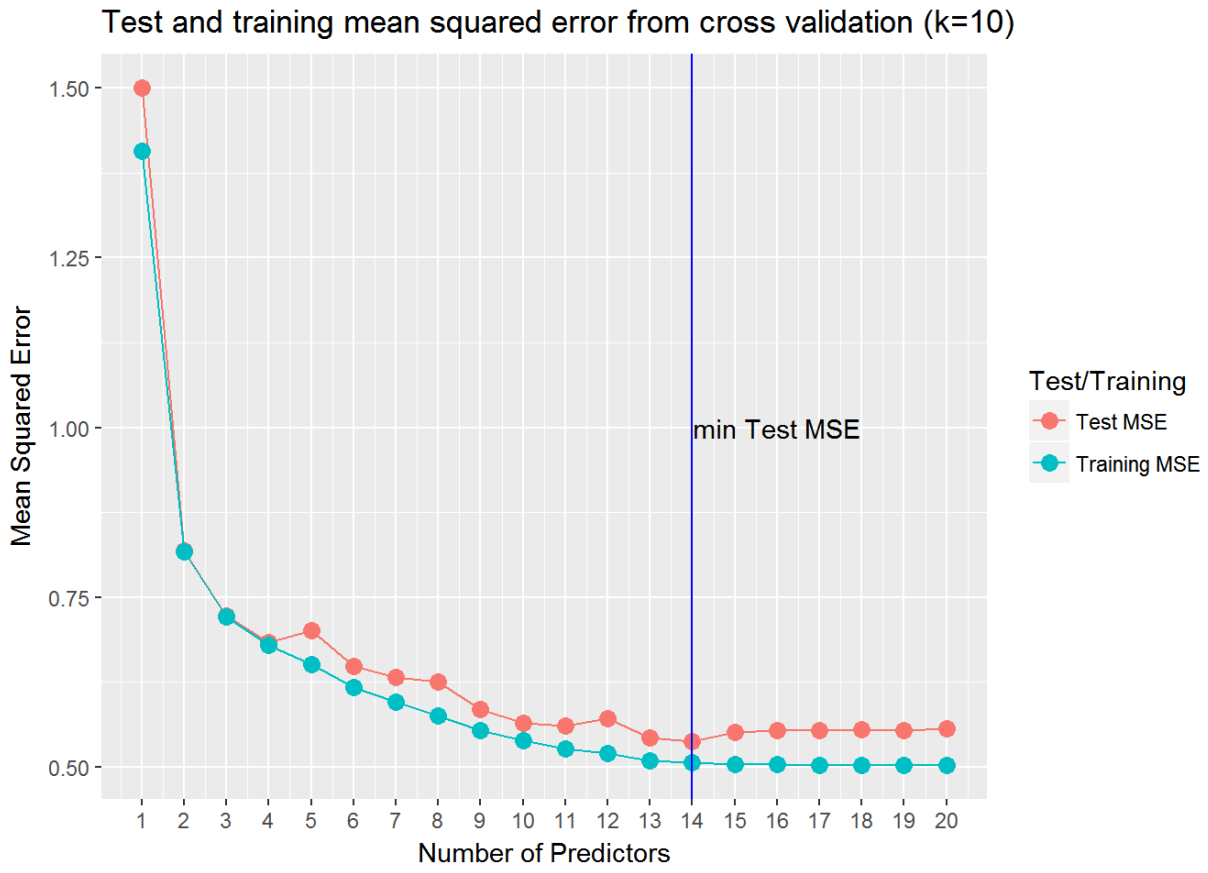

Figure 11: Testing and training MSE of Best Subsets

In addition, optimal number of predictors selected based on Mallow's Cp, Adj. R^2, and BIC are given in Table 6. The Mallow's Cp and cross validation had the same optimal number of predictors.

Table 6: Number of optimal predictors selected by Best Subsets

Method	Number.of.Predictors
Mallows Cp	14
Adj R2	15
BIC	13
Cross_Validation	14

The 14 predictors and their coefficients selected by the Best Subsets cross validation are summarized in Table 7. The Best subsets model selected with Adj. R^2 had an additional predictor 'chas(Othersise)', and selected with BIC removed 'dis' from the predictors selected by the cross validation.

Table 7: Coefficients selected by Best Subsets CV

(Intercept)	-3.0457
zn	-0.0112
nox	3.7692
age	0.0059
dis	-0.0525
rad3	0.5729
rad4	1.2113
rad5	0.8347
rad6	0.6340
rad7	1.3876
rad8	1.6143
rad24	3.7171
ptratio	-0.0787
lstat	0.0276
exp.black	-0.0026

RIDGE REGRESSION

The ridge regression employed an external cross validation method with 5 folds. Within each fold, cv.glmnet with nfolds=10 was used to get the best lambda. On the left panel of Figure 12, λ_{min} that gave minimum MSE for the training set in each fold is shown. On the right panel, the Test MSE was calculated for the corresponding λ_{min} in each fold.

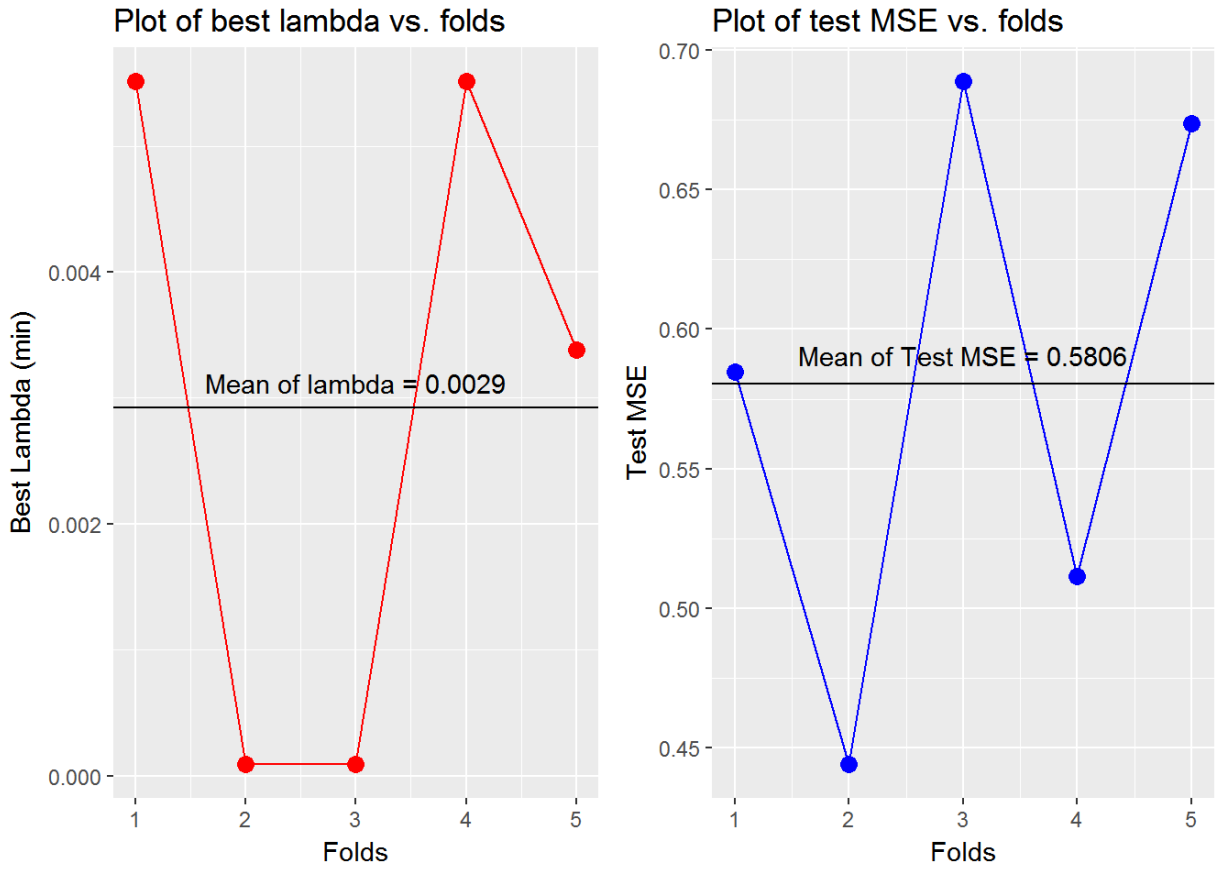

Figure 12: Plot of Best lambda(min) vs. folds (left), and plot of Test MSE vs. folds (right) for Ridge Regression

In Figure 13, MSE versus $log(\lambda)$ is plotted for the last fold in Figure 12 to assess whether pursuing the largest λ_{1se} that is within 1 standard error of the minimum MSE for each fold may be beneficial. Since the MSE stays flat until steep ascent begins towards the high λ, seeking λ_{1se} may be beneficial without much loss in prediction accuracy.

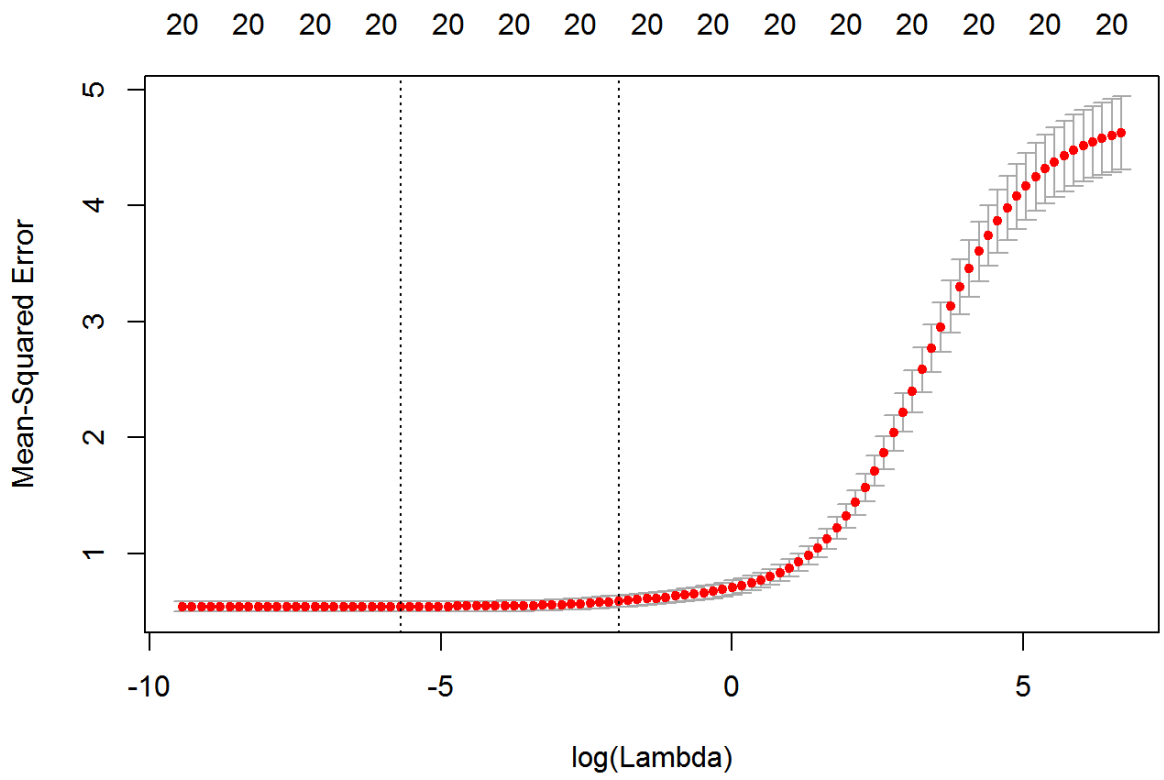

Figure 13: MSE vs. log(lambda) from Ridge Regression of the 10th fold

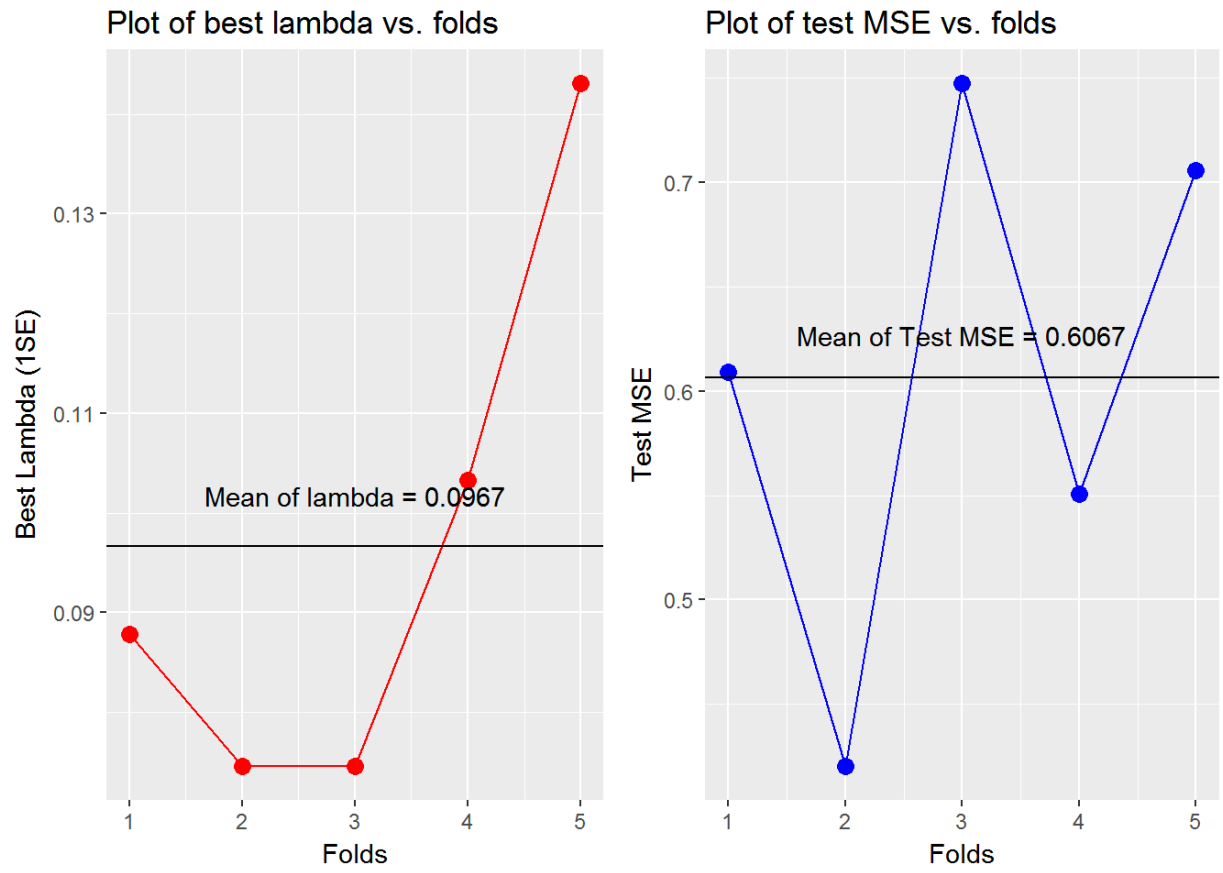

Figure 14: Plot of Best lambda(1se) vs. folds (left), and plot of Test MSE vs. folds (right)

A similar plot assessing the best λ_{1se} and corresponding Test MSE are given in Figure 14. As expected, increase in λ from $\lambda_{min} = 0.0029$ to $\lambda_{1se} = 0.0967$ resulted in minimal increase in the Test MSE of 0.6067 from 0.5806.

Ridge regression with log(crim) regressed on all predictors (with transformed $exp(10 \times \frac{black}{max(black)})$) using full dataset (both training and testing) are given in Figure 15. The coefficients estimated by the ordinary least squares are given when $\lambda = 0$, which is equivalent to 'Full model with transformed variables'. With increasing λ hence increasing model bias, the coefficients eventually all become 0. Ridge regression, however, retains all 20 predictors for λ_{min} and λ_{1se}.

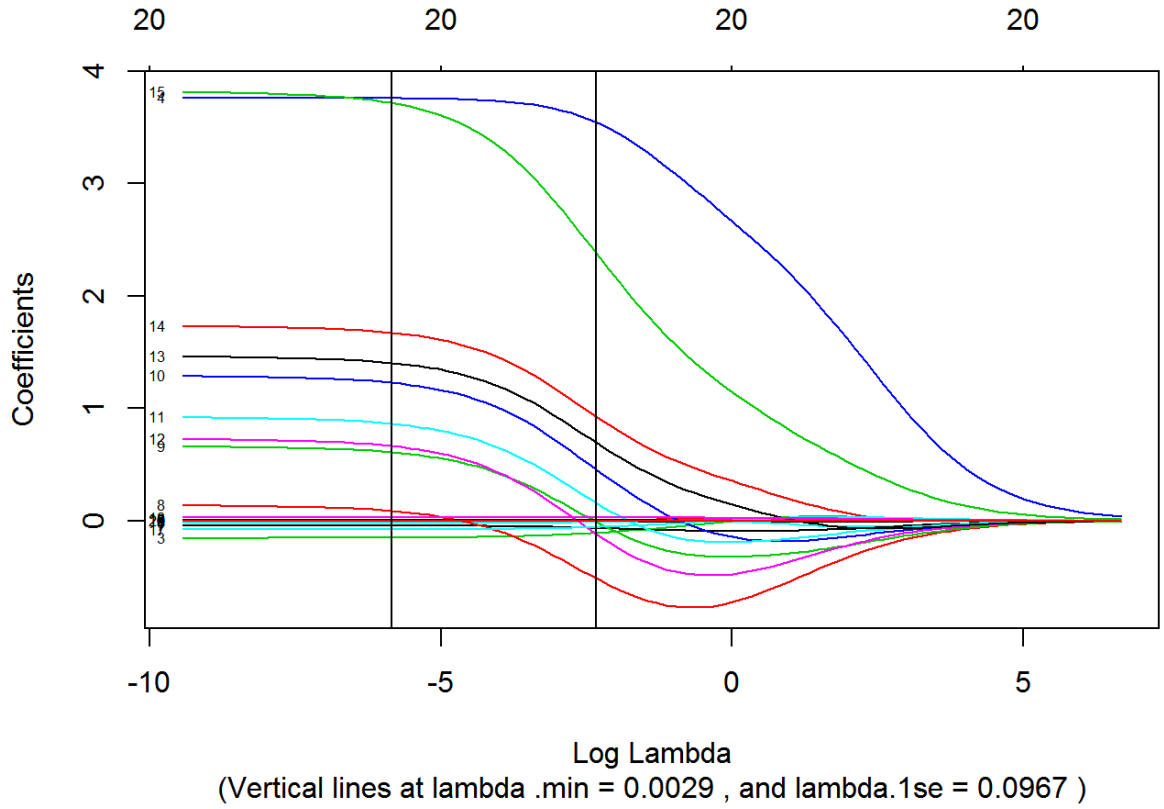

Figure 15: Coefficients vs. log(lambda) for Ridge Regression with full dataset

```
Ridge regression MSE with full data: 0.5054 (lambda.min = 0.0029 )
```

```
Ridge regression MSE with full data: 0.5378 (lambda = 0.0967 )
```

The MSE of the model using the full dataset only increases by 6.4209% from 0.5054 to 0.5378. The coefficients from the Ridge regression are summarized in Table 8 in the DISCUSSION section.

LASSO REGRESSION

The Lasso regression also employed external cross validation with 5 folds using the same seeding as the Ridge Regression. Similarly, a cv.glmnet with nfolds=10 was used to obtain the best λ for each external fold. On the left panel of Figure 16, λ_{min} that gave minimum MSE for the training set in each fold is shown. On the right panel, the Test MSE was calculated for the corresponding λ_{min} in each fold.

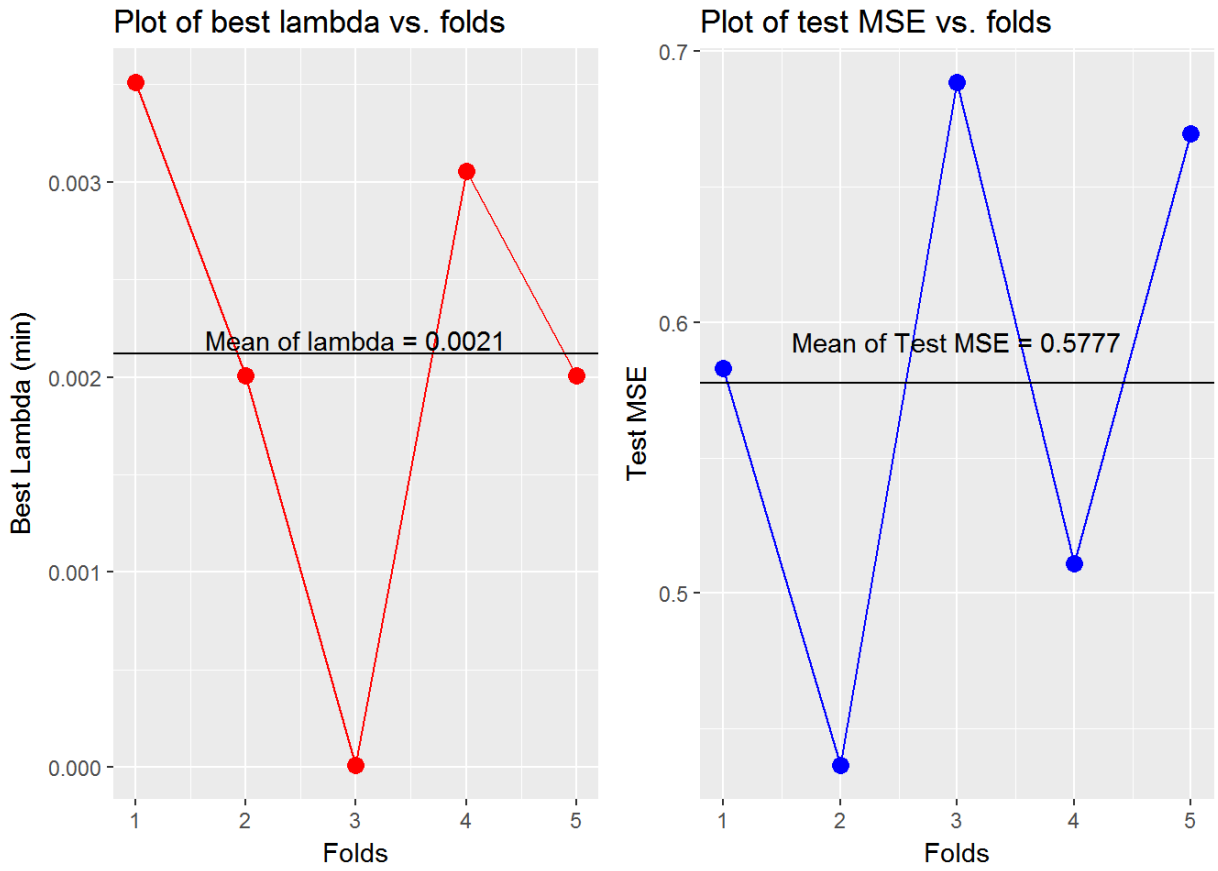

Figure 16: Plot of Best lambda(min) vs. folds (left), and plot of Test MSE vs. folds (right) for Lasso Regression

In Figure 17, the MSE versus $log(\lambda)$ of the last fold from Figure 16 is plotted. The long flat MSE susggests that the use of λ_{1se} may be useful especially considering the decrease in the number of optimal predictors.

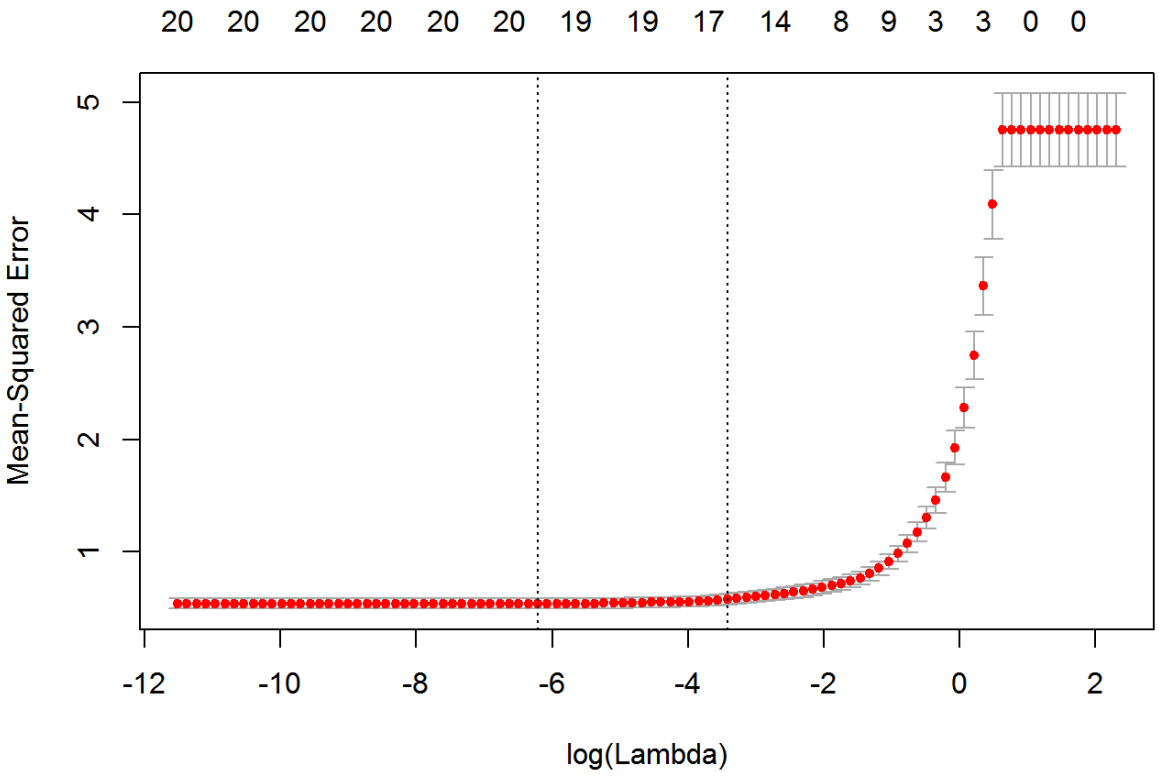

Figure 17: MSE vs. log(lambda) from Lasso Regression of the 10th fold

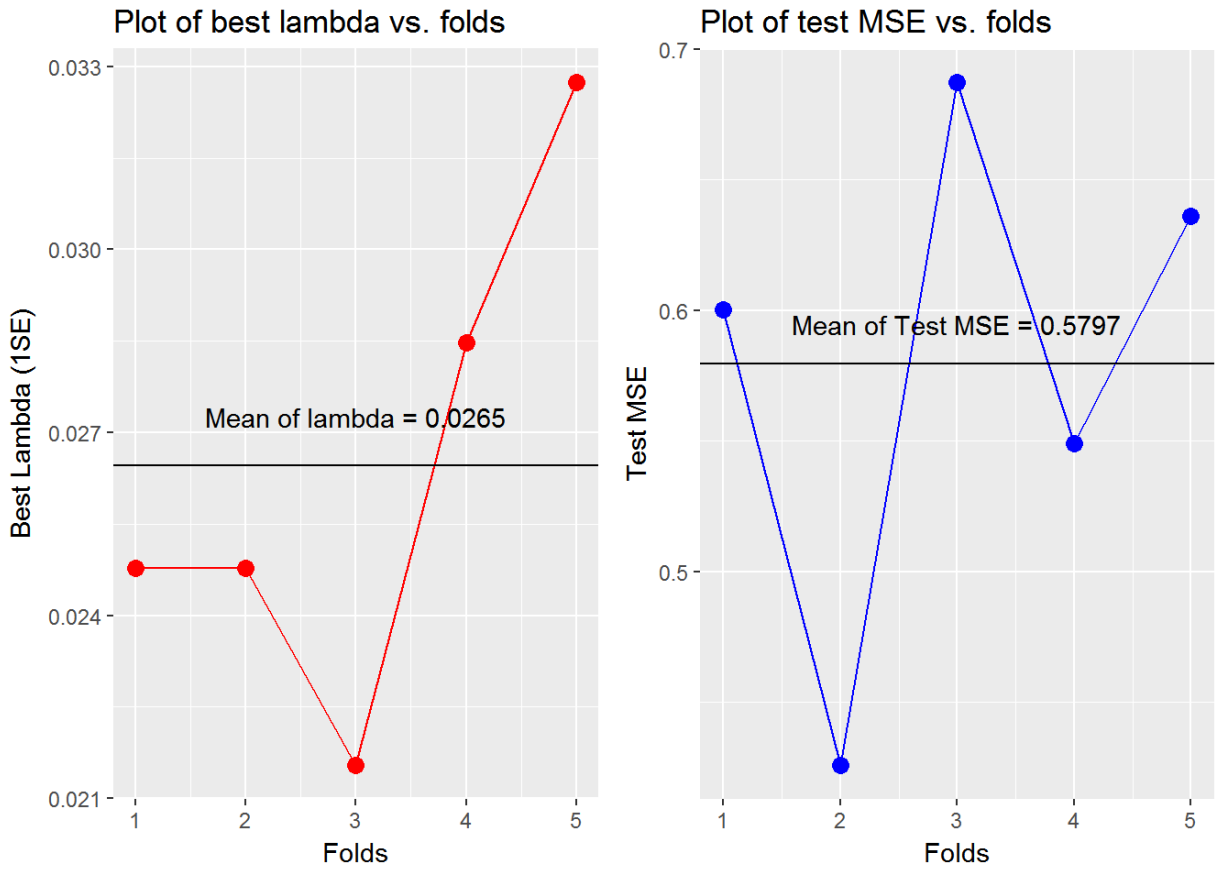

Figure 18: Plot of Best lambda(1se) vs. folds (left), and plot of Test MSE vs. folds (right) for Lasso Regression

In Figure 18, best λ_{1se} and corresponding Test MSEs for each fold are provided. The Test MSE only increased by 0.3453% from 0.5777 to 0.5797 when λ_{1se} was used while the number of optimal predictors decresed from 19 to 15. By increasing λ and thus eliminating predictors, Lasso regression balances model bias and variance with negligible decrease in model prediction accuracy.

Lasso regression with log(crim) regressed on all predictors (with transformed variables) using the full dataset (both training and testing) are given in Figure 19. The model bias is increased by increasing λ, which forces some of the predictors to become 0.

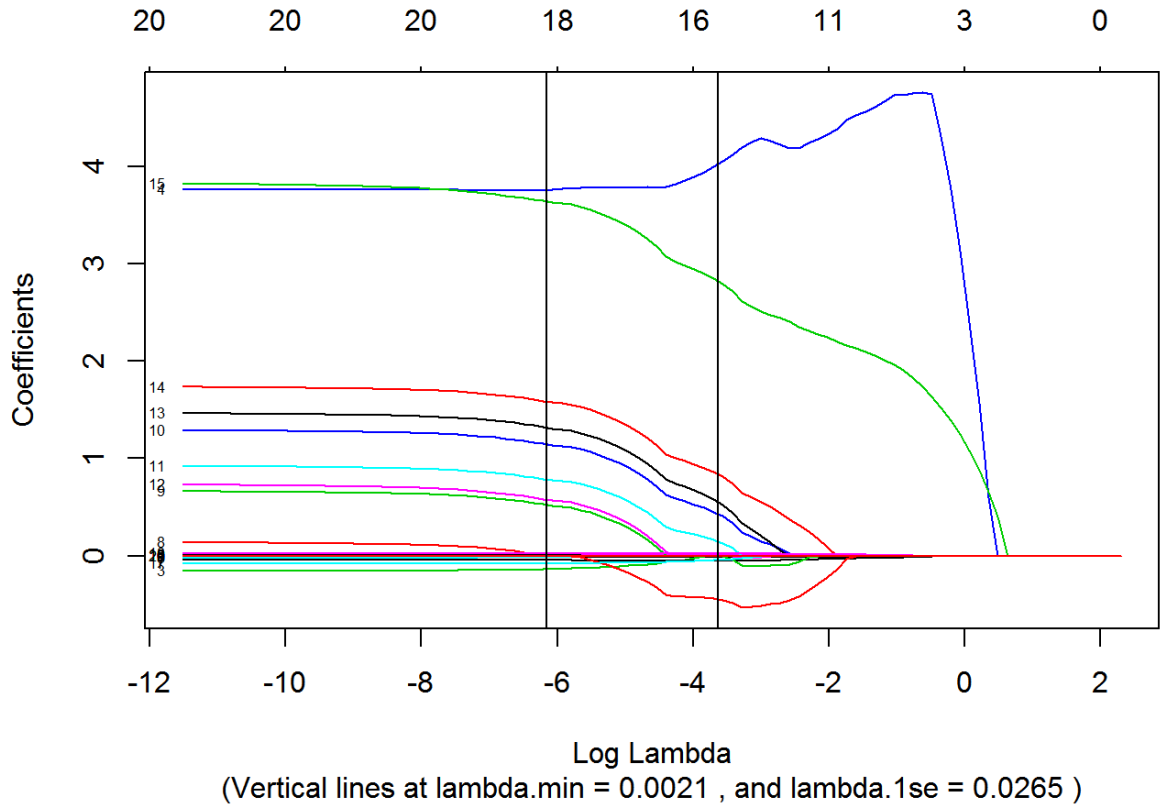

Figure 19: Coefficients vs. log(lambda) for Lasso Regression with full dataset

```
Lasso regression MSE with full data: 0.5061 (lambda = 0.0021 )
```

```
Lasso regression MSE with full data: 0.5383 (lambda = 0.0265 )
```

The MSEs were calculated for the full dataset using the two models selected by Lasso regression with best λ_{min} and λ_{1se}. The MSE only increases by 6.361% from 0.5061 to 0.5383 when λ_{1se} was used. The coefficients acquired from the Lasso regression are summarized in Table 8 in the DISCUSSION section.

DISCUSSION

A summary table of selected coefficients, Test MSE, and MSE using the full dataset for respective model selection approaches are given in Table 8. The comparison were made on 7 models, which were full model with transformed variables ("Full_model"), manually selected model ("Manual"), Best subsets ("Best_sub"), Ridge regression with λ_{min} ("Rdige(min)") and λ_{1se} ("Ridge(1se)"), and Lasso regression with λ_{min} ("Lasso(min)") and λ_{1se} ("Lasso(1se)"). The Test MSE of full and manual models were calculated with ordinary linear squares estimates with the same seeding used for regularized models with 5 folds.

Table 8: Summary table of model metric and coefficients

	Full_model	Manual	Best_sub	Ridge(min)	Ridge(1se)	Lasso(min)	Lasso(1se)
(Intercept)	-3.1616	-3.5893	-3.0457	-3.1593	-3.1821	-3.1084	-3.2725
zn	-0.0110	-0.0126	-0.0112	-0.0111	-0.0112	-0.0111	-0.0104
indus	0.0076	NA	NA	0.0071	0.0055	0.0067	0.0043
chasOtherwise	-0.1534	NA	NA	-0.1509	-0.1062	-0.1373	NA
nox	3.7620	4.1668	3.7692	3.7604	3.5489	3.7541	4.0271
rm	-0.0199	NA	NA	-0.0196	-0.0109	-0.0118	NA
age	0.0059	0.0069	0.0059	0.0059	0.0055	0.0058	0.0053

	Full_model	Manual	Best_sub	Ridge(min)	Ridge(1se)	Lasso(min)	Lasso(1se)
dis	-0.0423	NA	-0.0525	-0.0433	-0.0639	-0.0446	-0.0520
rad2	0.1430	0.2209	NA	0.0890	-0.5099	0.0005	-0.4453
rad3	0.6670	0.6838	0.5729	0.6106	-0.0064	0.5225	NA
rad4	1.2945	1.3321	1.2113	1.2293	0.4596	1.1445	0.4311
rad5	0.9276	0.9466	0.8347	0.8646	0.1609	0.7827	0.1508
rad6	0.7347	0.7671	0.6340	0.6660	-0.1138	0.5759	NA
rad7	1.4676	1.4557	1.3876	1.4064	0.6988	1.3156	0.5547
rad8	1.7391	1.7152	1.6143	1.6744	0.9311	1.5824	0.8402
rad24	3.8271	3.8593	3.7171	3.7194	2.3861	3.6372	2.8199
tax	-0.0001	NA	NA	0.0000	0.0014	NA	0.0002
ptratio	-0.0789	-0.0811	-0.0787	-0.0777	-0.0557	-0.0766	-0.0392
lstat	0.0276	0.0264	0.0276	0.0280	0.0308	0.0276	0.0244
medv	0.0033	NA	NA	0.0036	0.0053	0.0027	NA
exp.black	-0.0026	-0.0026	-0.0026	-0.0026	-0.0027	-0.0026	-0.0025
MSE(all_data)	0.5052	0.5100	0.5079	0.5054	0.5378	0.5061	0.5383
Test_MSE	0.5801	0.5541	0.5373	0.5806	0.6067	0.5777	0.5797

All models had similar prediction capabilities as measured by the Test MSE. The lowest Test MSE was observed for the Best subsets model with 14 predictors, and the Manual selection was the second lowest with the same number of predictors. Since the Ridge regressions resulted in all 20 predictors with similar or lower prediction accuracy, there was not much gained from increasing the model bias through this method. The Lasso regression with λ_{min} eliminated one predictor from the full model while lowering the Test MSE very slightly compared with the full model. The Lasso regression with λ_{1se} removed 5 predictors with similar Test MSE when compared with the full model.

Among the 6 predictors eliminated by the Manual selection and Best subsets, 5 predictors (‘indus’, ‘chas(Otherwise)’, ‘rm’, ‘tax’ and ‘medv’) coincided and ‘dis’ was included in the Manual selection whereas ‘rad2’ was in the Best subsets. The removal of a categorical variable is typically performed as a whole rather than by each level. Therefore, the removal of a single level, ‘rad2’, may need to be justified with the domain knowledge for a meaningful analysis. On the other hand, if the sole purpose of the model selection is to obtain highest prediction accuracy, such discussion becomes a moot point.

Selection of Predictors

In this subsection, predictor selection between Best subsets and Lasso regression has been compared. Similar to Figure 19 created by the Lasso regression, coefficients versus number of predictors from Best subsets selection is provide in Figure 20.

Starting from the right side (Number of predictor = 1), one can observed the changes in the coefficient values as more predictors are added. The larger coefficients enter earlier followed by predictors with smaller coefficients. It is speculated that the fluctuation of coefficient values is caused by severity of collinearity among predictors. The variation of coefficients stablized around 14 predictors, which coincided with number of predictors that minimized the prediction error.

Similarly in Figure 19, it is interesting to note that the magnitude of coefficients stablized around 18-19 predictors, which again coincided with the number of predictors that minimized the prediction error for Lasso regression.

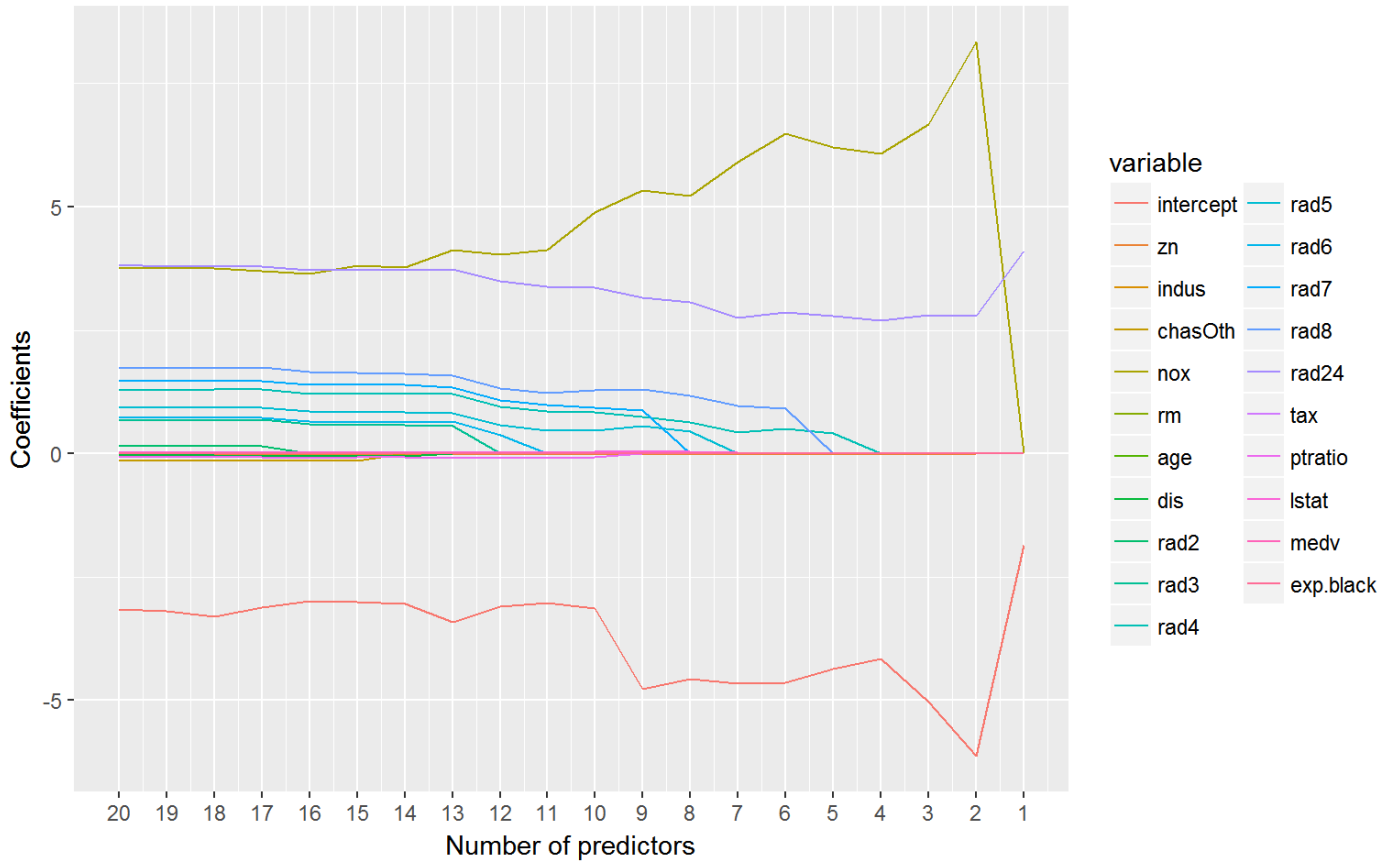

Figure 20: Coefficients vs. number of predictors for Best subsets

For easy comparison between coefficients selected by Best subsets and Lasso regression, first 5 predictors are provided in Table. 9. One can observe the similarities of the dominant coefficients selected by the two model selection methods as predictors are added.

Table 9: Selected coefficients for first 5 predictors from Best subsets and Lasso regression

	Best(n=1)	Lasso(n=1)	Best(n=2)	Lasso(n=2)	Best(n=3)	Lasso(n=3)	Best(n=4)	Lasso(n=4)	Best(n=5)	Lasso(n=5)
(Intercept)	-1.852	-0.8773	-6.134	-1.0075	-5.0354	-4.1608	-4.1666	-4.2660	-4.3580	-4.2802
zn	NA	NA	NA	NA	-0.0155	NA	-0.0157	NA	-0.0154	NA
indus	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
chasOtherwise	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
nox	NA	NA	8.343	NA	6.6670	4.6748	6.0718	4.7694	6.2017	4.7747
rm	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
age	NA	NA	NA	NA	NA	NA	NA	0.0005	NA	0.0007
dis	NA	NA	NA	NA	NA	NA	NA	NA	NA	-0.0009
rad2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
rad3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
rad4	NA	NA	NA	NA	NA	NA	NA	NA	0.4110	NA
rad5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
rad6	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
rad7	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
rad8	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
rad24	4.109	0.3714	2.780	0.4908	2.8090	1.6017	2.6915	1.6456	2.7966	1.6565
tax	NA	NA	NA	0.0002	NA	0.0009	NA	0.0009	NA	0.0009
ptratio	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
lstat	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
medv	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
exp.black	NA	NA	NA	NA	NA	NA	-0.0032	NA	-0.0032	NA

CONCLUSIONS

A comparative study on four model selection methods, Best subsets, Ridge regression, Lasso regression and Manual selection, has been performed on ‘Boston’ dataset included in the MASS package. The objective was to find a relationship between the reponse variable, ‘crim’, and other predictors in the dataset. A cross validation was employed on all three automated selection methods. Following conclusions may be drawn from this analysis:

- Among 14 variables, two variables were considered as categorical: ‘chas’ with 2 levels, and ‘rad’ with 9 levels.
- The exploratory data analysis revealed that the response variable ‘crim’ and predictor ‘black’ had highly skewed distribution.
- The full model with original variables (no transformation), where ‘crim’ was regressed on all 13 predictors, resulted in $R^2 = 46.0415\%$. Diagonostic plots, however, showed violation of linear regression assumptions, which led to following variable transfomation:

$$\begin{aligned} \log.\text{crim} &= log(crim) \\ \text{exp.black} &= \frac{1}{100} \times e^{(10 \times \frac{black}{max(black)})} \end{aligned}$$

- The full model with transformed variables (log.crim and exp.black) resulted in significant improvement with $R^2 = 89.171\%$.
- Using the transformed variables, the four model selection methods were compared. The Manual selection and Best subsets resulted in elimination of 6 predictors where the lowest Test MSE was from the Best subsets followed by the Manual selection. The slight improvements in the prediction accuracy by Manual selection and Best subsets indicate a small degree of overfitting in the full model.
- The Ridge regression was not able to reduce the Test MSE when compared with the full model while the Lasso regression resulted in negligible or minor improvement in the Test MSE. The increase in model bias through Ridge or Lasso regression for the transformed dataset did not result in significant improvement in prediction accuracy.
- For Best subsets and Lasso regression, the number of predictors that gave the lowest Test MSE coincided with where the variation of coefficient values stablized.
- If the prediction accuracy is critical, Best substes model is recommended as it had the lowest Test MSE, but requires justification of removing a single factor level ‘rad2’ for a meaningful analysis.

APPENDIX

All R codes used in producing the results are included below:

```
##############################
### Initial setup

knitr::opts_chunk$set(comment=NA, echo=FALSE, warning=FALSE, message=FALSE,
                      fig.align="center")
options(knitr.table.format="html", knitr.table.align="center")
options(digits=4)

rm(list=ls())

##############################
### Explortory Data Analysis
##############################

library(MASS)
data("Boston")

str(Boston)
sum(is.na(Boston))

mydata = Boston
mydata$chas = factor(mydata$chas, levels=c(0, 1), labels=c("Bounds river","Otherwise"))
mydata$rad = as.factor(mydata$rad)

str(mydata)

# Table 1: Summary of categorical variable, 'chas'

library(knitr)

kable(summary(mydata$chas), col.names="",
      caption="Table 1: Summary of categorical variable, 'chas'",
      format="html", table.attr = "style='width:40%;'")

# Figure 1: Histogram of variable 'rad'

library(ggplot2)

ggplot(data=mydata) +
   geom_histogram(aes(rad), stat="count")

# Figure 2: Boxplots of all 12 quantitative variables

library(reshape2)

melt.data = melt(data=mydata)

ggplot(data=melt.data) +
   geom_boxplot(aes(x="", y=value)) +
   facet_wrap(~variable, scale="free") +
   labs(x="All categorical variables", y="count")

# Figure 3: Histogram of variable 'crim'

library(car)

ggplot(data=mydata) +
   geom_histogram(aes(crim)) +
```

```
   scale_x_continuous(trans="log", labels=function(x) round(x, digits=2))

# Table 2: Summary statistics of 'crim'

table2 = summary(mydata[which.names("crim", colnames(mydata))])

kable(table2, col.names="",
      caption="Table 2: Summary statistics of 'crim'",
      format="html", table.attr = "style='width:40%;'")

# Figure 4: Scatterplot and correlation plot matrix of the dataset

library(GGally)
ggpairs(mydata)

# Figure 5: Scatterplots of 'crim' versus other variables

library(gridExtra)

scatterplot_col = function(column, yvar, data) {
   ggplot(data, aes_string(y = yvar)) +
      geom_point(aes_string(x = column))
}

mplot = lapply(colnames(mydata)[-1], scatterplot_col, yvar="crim", data=mydata)

grid.arrange(mplot[[1]], mplot[[2]], mplot[[3]], mplot[[4]], mplot[[5]],
             mplot[[6]], mplot[[7]], mplot[[8]], mplot[[9]], mplot[[10]],
             mplot[[11]], mplot[[12]], mplot[[13]], ncol=3)

# Figure 6: Scatterplot of 'log.crim' versus other variables

mydata2 = mydata
mydata2$log.crim = log(mydata2$crim)
mydata2 = mydata2[-1]

mplot = lapply(colnames(mydata)[-1], scatterplot_col, yvar="log.crim", data=mydata2)

grid.arrange(mplot[[1]], mplot[[2]], mplot[[3]], mplot[[4]], mplot[[5]],
             mplot[[6]], mplot[[7]], mplot[[8]], mplot[[9]], mplot[[10]],
             mplot[[11]], mplot[[12]], mplot[[13]], ncol=3)

# Figure 7: Scatterplot of 'log.crim' versus 'exp.black'

mydata3 = mydata2
mydata3$exp.black = exp(10*mydata3$black / max(mydata3$black)) / 100
#mydata3$exp.black = (mydata3$black / max(mydata3$black))^6
mydata3 = mydata3[-which.names("black", colnames(mydata3))]

ggplot(data=mydata3) +
   geom_point(aes(x=exp.black, y=log.crim))

########################################
### Full Model with Original Variables
########################################

full.lm = lm(crim ~ ., data=mydata)
full.lm_summary = summary.lm(full.lm)
full.lm_summary
```

```
# Figure 8: Residual vs. fits plot (left), and Normal Q-Q plot of residuals(right)

par(mfrow=c(1,2))

plot(full.lm, which=1)
plot(full.lm, which=2)


# Figure 9: Standardized residuals vs. fits plot (left), and Standard residuals vs. leverage with contour
s of Cook's distance

par(mfrow=c(1,2))

plot(full.lm, which=3)
abline(h=sqrt(3), col='blue')

hii = hatvalues(full.lm)
h2 = 2*sum(hii) / length(hii)
h3 = 3*sum(hii) / length(hii)

plot(full.lm, which=5)
abline(v=c(h2, h3), col=c('blue', 'blue'), lty=c(2, 1))
abline(h=3, col='blue')

# Table 3: VIFs from full model with original variables

kable(vif(full.lm), caption="Table 3: VIFs from full model with original variables",
      format="html", table.attr="style='width:40%;')")

##############################################
### Full Model with Transformed Variables
##############################################

full.lm1 = lm(log.crim ~ ., data=mydata3)
full.lm1_summary = summary.lm(full.lm1)
full.lm1_summary

##############################################
### Manual Model Selection
##############################################

### Ordered full model

full.lm2 = lm(log.crim ~ zn + nox + rad + ptratio + exp.black + age + lstat + chas + indus + dis + rm + t
ax + medv, data=mydata3)

# Table 4: ANOVA result of transformed full model (Type I, Sequential sum of squares)

kable(anova(full.lm2),
      caption="Table 4: ANOVA result of transformed full model
      <br>(Type I, Sequential sum of squares)",
      format="html", table.attr="style='width:60%;')")

### Manual model selected

manual.pick = lm(log.crim ~ zn + nox + rad + ptratio + exp.black + age + lstat, data=mydata3)
manual.pick_summary = summary.lm(manual.pick)
manual.pick_summary
```

```
# Figure 10: Diagnostic plots for transformed and reduce model

par(mfrow=c(2,2))

plot(manual.pick, which=1)
plot(manual.pick, which=2)

plot(manual.pick, which=3)
abline(h=sqrt(3), col='blue')

hii = hatvalues(manual.pick)
h2 = 2*sum(hii) / length(hii)
h3 = 3*sum(hii) / length(hii)

plot(manual.pick, which=5)
abline(v=c(h2, h3), col=c('blue', 'blue'), lty=c(2, 1))
abline(h=c(-3,3), col='blue')


# Table 5: VIFs from reduced transformed model

kable(vif(manual.pick), caption="Table 5: VIFs from reduced transformed model",
      format="html", table.attr="style='width:40%;')")

###################################
### Best Subsets Regression
###################################

library(leaps)

# Code from ISLR for predict() method for regsubsets(): not used in calculation
predict.regsubsets = function(object, newdata, id, ...){
  form = as.formula(object$call[[2]])
  mat = model.matrix(form, newdata)
  coefi = coef(object, id=id)
  xvars = names(coefi)
  mat[,xvars]%*%coefi
}

###################################
# Best Subsets Cross Validation

library(leaps)

npred = 20
nmax = 20  # Total number of predictors including factor levels

k = 10
set.seed(10)
folds = sample(1:k, nrow(mydata3), replace=TRUE)

cv.errors = matrix(NA, k, npred)
train.errors = matrix(NA, k, nmax)

for(j in 1:k) {
   best.subset = regsubsets(log.crim ~ ., data=mydata3[folds!=j,], nvmax=npred)
   train.errors[j,] = summary(best.subset)$rss / best.subset$nn
```

```
   x.test = model.matrix(log.crim ~ ., data=mydata3[folds==j,])
   y.test = mydata3$log.crim[folds==j]

   for(i in 1:npred) {
      coefi = coef(best.subset, id=i)
      pred = x.test[, names(coefi)] %*% coefi
      cv.errors[j,i] = mean((y.test - pred)^2)
   }
}

mean.cv.errors = apply(cv.errors, 2, mean)
mean.train.errors = apply(train.errors, 2, mean)

# Best subset CV optimal number of predictors
index.best = which.min(mean.cv.errors)

# Test MSE of Best Subsets CV
test.MSE_bestsub = mean.cv.errors[index.best]

##########################
### Plots and Tables

#Figure 11: Testing and training MSE of Best Subsets

ggplot() +
   geom_point(aes(x=c(1:npred),y=mean.cv.errors, color="Test MSE"), size=3) +
   geom_line(aes(x=c(1:npred),y=mean.cv.errors, color="Test MSE"), linetype="solid") +
   geom_point(aes(x=c(1:npred),y=mean.train.errors, color="Training MSE"), size=3) +
   geom_line(aes(x=c(1:npred),y=mean.train.errors, color="Training MSE"), linetype="solid") +
   scale_x_continuous(breaks=c(1:npred)) +
   geom_vline(xintercept=c(index.best), color="blue") +
   annotate("text", label="min Test MSE", x=16, y=1) +
   labs(x="Number of Predictors", y="Mean Squared Error", color=c("Test/Training")) +
   ggtitle("Test and training mean squared error from cross validation (k=10)")

###
### Best subset with all data
###

reg.best = regsubsets(log.crim ~ ., data=mydata3, nvmax=nmax)
summary.BS = summary(reg.best)

# Optimal predictor numbers for Mallows Cp, Adj.R2, BIC
BS.Cp = which.min(summary.BS$cp)
BS.AdjR2 = which.max(summary.BS$adjr2)
BS.BIC = which.min(summary.BS$bic)

# Table 6: Number of optimal predictors selected by Best Subsets
BS.summary = data.frame("Method" = c("Mallows Cp", "Adj R2", "BIC", "Cross_Validation"), "Number of Predi
ctors" = c(BS.Cp, BS.AdjR2, BS.BIC, index.best))

kable(BS.summary, caption="Table 6: Number of optimal predictors selected by Best Subsets", format="html"
, table.attr = "style='width:40%;'")

# Table 7: Coefficients selected by Best Subsets CV
kable(coef(reg.best, index.best), col.names="",
      caption="Table 7: Coefficients selected by Best Subsets CV",
      format="html", table.attr = "style='width:30%;'")
```

```
# Chekcing coefficients selected by Adj R2 and BIC
names(coef(reg.best, BS.AdjR2))
names(coef(reg.best, BS.BIC))

##############################
### Ridge Regression
##############################

library(glmnet)

k = 5
x = model.matrix(log.crim ~ ., data=mydata3)[,-1]
y = mydata3$log.crim

grid = 8*10^seq(2, -5, length=100)

# Store errors and best lambda with lambda.min
cv.errors = rep(NA, k)
bestlam_ridge = rep(NA, k)

# Store errors and best lambda with lambda.1se
cv.errors.1se = rep(NA, k)
bestlam_ridge.1se = rep(NA, k)

for(j in 1:k) {
   set.seed(12)
   ridge.cv.out = cv.glmnet(x[folds!=j,], y[folds!=j], alpha=0, nfolds=10, lambda=grid)
   bestlam_min_ridge = ridge.cv.out$lambda.min
   bestlam_1se_ridge = ridge.cv.out$lambda.1se

   bestlam_ridge[j] = bestlam_min_ridge
   ridge.cv.pred = predict(ridge.cv.out, s=bestlam_ridge[j], newx=x[folds==j,])
   cv.errors[j] = mean((ridge.cv.pred - y[folds==j])^2)

   bestlam_ridge.1se[j] = bestlam_1se_ridge
   ridge.cv.pred.1se = predict(ridge.cv.out, s=bestlam_ridge.1se[j], newx=x[folds==j,])
   cv.errors.1se[j] = mean((ridge.cv.pred.1se - y[folds==j])^2)
}

cv.output_ridge = data.frame("folds"=c(1:k), "lambda"=bestlam_ridge, "Test_MSE"=cv.errors)

cv.output_ridge.1se = data.frame("folds"=c(1:k), "lambda"=bestlam_ridge.1se, "Test_MSE"=cv.errors.1se)

##############################
### Ridge Regression Plots

###
### for lambda.min
###

avg.bestlam_ridge = mean(bestlam_ridge)
avg.cv.errors = mean(cv.errors)

# Test MSE from Ridge Regression with lambda.min
test.MSE_ridge.min = avg.cv.errors

# Figure.12, fig.cap="Figure 12: Plot of Best lambda(min) vs. folds (left), and plot of Test MSE vs. fold
s (right)"
pp1 <- ggplot(aes(x=folds, y=lambda), data=cv.output_ridge) +
```

```
   geom_point(color="red", size=3) +
   geom_line(color="red", linetype="solid") +
   geom_hline(yintercept = avg.bestlam_ridge) +
   annotate("text", x=(k+1)/2, y=avg.bestlam_ridge+0.0002,
            label=paste("Mean of lambda =", round(avg.bestlam_ridge,4))) +
   labs(x="Folds", y="Best Lambda (min)") +
   ggtitle("Plot of best lambda vs. folds")


pp2 <- ggplot(aes(x=folds, y=Test_MSE), data=cv.output_ridge) +
   geom_point(color="blue", size=3) +
   geom_line(color="blue", linetype="solid") +
   geom_hline(yintercept = avg.cv.errors) +
   annotate("text", x=(k+1)/2, y=avg.cv.errors+0.01,
            label=paste("Mean of Test MSE =", round(avg.cv.errors,4))) +
   labs(x="Folds", y="Test MSE") +
   ggtitle("Plot of test MSE vs. folds")

grid.arrange(pp1, pp2, ncol=2)

# Figure 13: MSE vs. log(lambda) from Ridge Regression of the 10th fold
plot(ridge.cv.out)

###
### for lambda.1se
###

avg.bestlam_ridge.1se = mean(bestlam_ridge.1se)
avg.cv.errors.1se = mean(cv.errors.1se)

# Test MSE from Ridge Regression with lambda.1se
test.MSE_ridge.1se = avg.cv.errors.1se

pp3 <- ggplot(aes(x=folds, y=lambda), data=cv.output_ridge.1se) +
   geom_point(color="red", size=3) +
   geom_line(color="red", linetype="solid") +
   geom_hline(yintercept = avg.bestlam_ridge.1se) +
   annotate("text", x=(k+1)/2, y=avg.bestlam_ridge.1se+0.005,
            label=paste("Mean of lambda =", round(avg.bestlam_ridge.1se,4))) +
   labs(x="Folds", y="Best Lambda (1SE)") +
   ggtitle("Plot of best lambda vs. folds")


pp4 <- ggplot(aes(x=folds, y=Test_MSE), data=cv.output_ridge.1se) +
   geom_point(color="blue", size=3) +
   geom_line(color="blue", linetype="solid") +
   geom_hline(yintercept = avg.cv.errors.1se) +
   annotate("text", x=(k+1)/2, y=avg.cv.errors.1se+0.02,
            label=paste("Mean of Test MSE =", round(avg.cv.errors.1se,4))) +
   labs(x="Folds", y="Test MSE") +
   ggtitle("Plot of test MSE vs. folds")

grid.arrange(pp3, pp4, ncol=2)

##########################################
### Fit Ridge Regression to full dataset

out_ridge = glmnet(x, y, alpha=0, lambda=grid)
```

```
#Figure.15, fig.ap="Figure 15: Coefficients vs. log(lambda) for Ridge Regression with full dataset
plot(out_ridge, xvar="lambda", label=T,
     sub=paste("(Vertical lines at lambda .min =", round(avg.bestlam_ridge, 4),
               ", and lambda.1se =", round(avg.bestlam_ridge.1se, 4), ")"))
abline(v=c(log(avg.bestlam_ridge), log(avg.bestlam_ridge.1se) ))

###
### Calculate MSE of Ride Regression using the full data
###

# for lambda(min)
pred = predict(out_ridge, s=avg.bestlam_ridge, newx=x, exact=T, x=x, y=y)
MSE_ridge.min = mean((pred - y)^2)
cat("Ridge regression MSE with full data:", MSE_ridge.min,
    "(lambda.min =", round(avg.bestlam_ridge,4),")")

coef.bestlam_ridge = coef(out_ridge, s=avg.bestlam_ridge, exact=T, x=x, y=y)[,1]

# for lambda(1se)
pred = predict(out_ridge, s=avg.bestlam_ridge.1se, newx=x, exact=T, x=x, y=y)
MSE_ridge.1se = mean((pred - y)^2)
cat("Ridge regression MSE with full data:", MSE_ridge.1se,
    "(lambda =", round(avg.bestlam_ridge.1se,4),")")

coef.bestlam_ridge.1se = coef(out_ridge, s=avg.bestlam_ridge.1se,
                              exact=T, x=x, y=y)[,1]

### Sanity check1 (both should equal)
mean(full.lm2$residuals^2)

pred_0 = predict(out_ridge, s=0, newx=x, exact=T, x=x, y=y)
mean((pred_0 - y)^2)

###################################
### Lasso Rgression
###################################

grid = 10^seq(1, -5, length=100)

cv.errors = rep(NA, k)
bestlam_lasso = rep(NA, k)

cv.errors.1se = rep(NA, k)
bestlam_lasso.1se = rep(NA, k)

for(j in 1:k) {
   set.seed(12)
   lasso.cv.out = cv.glmnet(x[folds!=j,], y[folds!=j], alpha=1, nfolds=10, lambda=grid)
   bestlam_min_lasso = lasso.cv.out$lambda.min
   bestlam_1se_lasso = lasso.cv.out$lambda.1se

   bestlam_lasso[j] = bestlam_min_lasso
   lasso.cv.pred = predict(lasso.cv.out, s=bestlam_lasso[j], newx=x[folds==j,])
   cv.errors[j] = mean((lasso.cv.pred - y[folds==j])^2)

   bestlam_lasso.1se[j] = bestlam_1se_lasso
   lasso.cv.pred.1se = predict(lasso.cv.out, s=bestlam_lasso.1se[j], newx=x[folds==j,])
   cv.errors.1se[j] = mean((lasso.cv.pred.1se - y[folds==j])^2)
}
```

```
cv.output_lasso = data.frame("folds"=c(1:k), "lambda"=bestlam_lasso, "Test_MSE"=cv.errors)

cv.output_lasso.1se = data.frame("folds"=c(1:k), "lambda"=bestlam_lasso.1se, "Test_MSE"=cv.errors.1se)

##############################
### Lasso Regression Plots

###
### for lambda.min
###

avg.bestlam_lasso = mean(bestlam_lasso)
avg.cv.errors = mean(cv.errors)

# Test MSE from Lasso Regression with lambda.min
test.MSE_lasso.min = avg.cv.errors

# Figure 16: Plot of Best lambda(min) vs. folds (left), and plot of Test MSE vs. folds (right) for Lasso
 Regression
pp1 <- ggplot(aes(x=folds, y=lambda), data=cv.output_lasso) +
   geom_point(color="red", size=3) +
   geom_line(color="red", linetype="solid") +
   geom_hline(yintercept = avg.bestlam_lasso) +
   annotate("text", x=(k+1)/2, y=avg.bestlam_lasso+0.00007,
            label=paste("Mean of lambda =", round(avg.bestlam_lasso,4))) +
   labs(x="Folds", y="Best Lambda (min)") +
   ggtitle("Plot of best lambda vs. folds")

pp2 <- ggplot(aes(x=folds, y=Test_MSE), data=cv.output_lasso) +
   geom_point(color="blue", size=3) +
   geom_line(color="blue", linetype="solid") +
   geom_hline(yintercept = avg.cv.errors) +
   annotate("text", x=(k+1)/2, y=avg.cv.errors+0.015,
            label=paste("Mean of Test MSE =", round(avg.cv.errors,4))) +
   labs(x="Folds", y="Test MSE") +
   ggtitle("Plot of test MSE vs. folds")

grid.arrange(pp1, pp2, ncol=2)

# Figure 17: MSE vs. log(lambda) from Lasso Regression of the 10th fold
plot(lasso.cv.out)

###
### for lambda.1se
###

avg.bestlam_lasso.1se = mean(bestlam_lasso.1se)
avg.cv.errors.1se = mean(cv.errors.1se)

# Test MSE from Lasso Regression with lambda.1se
test.MSE_lasso.1se = avg.cv.errors.1se

pp3 <- ggplot(aes(x=folds, y=lambda), data=cv.output_lasso.1se) +
   geom_point(color="red", size=3) +
   geom_line(color="red", linetype="solid") +
   geom_hline(yintercept = avg.bestlam_lasso.1se) +
   annotate("text", x=(k+1)/2, y=avg.bestlam_lasso.1se+0.0008,
            label=paste("Mean of lambda =", round(avg.bestlam_lasso.1se,4))) +
```

```
   labs(x="Folds", y="Best Lambda (1SE)") +
   ggtitle("Plot of best lambda vs. folds")


pp4 <- ggplot(aes(x=folds, y=Test_MSE), data=cv.output_lasso.1se) +
   geom_point(color="blue", size=3) +
   geom_line(color="blue", linetype="solid") +
   geom_hline(yintercept = avg.cv.errors.1se) +
   annotate("text", x=(k+1)/2, y=avg.cv.errors.1se+0.015,
            label=paste("Mean of Test MSE =", round(avg.cv.errors.1se,4))) +
   labs(x="Folds", y="Test MSE") +
   ggtitle("Plot of test MSE vs. folds")

grid.arrange(pp3, pp4, ncol=2)

##########################################
### Fit Lasso Regression to full dataset

out_lasso = glmnet(x, y, alpha=1, lambda=grid)

# Figure 19: Coefficients vs. log(lambda) for Ridge Regression with full dataset
plot(out_lasso, xvar="lambda", label=T,
     sub=paste("(Vertical lines at lambda.min =", round(avg.bestlam_lasso, 4),
               ", and lambda.1se =", round(avg.bestlam_lasso.1se, 4), ")"))
abline(v=c(log(avg.bestlam_lasso), log(avg.bestlam_lasso.1se) ))

###
### Calculate MSE of Lasso Regression using the full data
###

# for lambda(min)
pred = predict(out_lasso, s=avg.bestlam_lasso, newx=x, exact=T, x=x, y=y)
MSE_lasso.min = mean((pred - y)^2)
cat("Lasso regression MSE with full data:", MSE_lasso.min,
    "(lambda =", round(avg.bestlam_lasso,4),")")

coef.bestlam_lasso = coef(out_lasso, s=avg.bestlam_lasso, exact=T, x=x, y=y)[,1]

# for lambda(1se)
pred = predict(out_lasso, s=avg.bestlam_lasso.1se, newx=x, exact=T, x=x, y=y)
MSE_lasso.1se = mean((pred - y)^2)
cat("Lasso regression MSE with full data:", MSE_lasso.1se,
    "(lambda =", round(avg.bestlam_lasso.1se,4),")")

coef.bestlam_lasso.1se = coef(out_lasso, s=avg.bestlam_lasso.1se,
                              exact=T, x=x, y=y)[,1]

### Sanity check2 (both should equal)
mean(full.lm2$residuals^2)

pred_0 = predict(out_ridge, s=0, newx=x, exact=T, x=x, y=y)
mean((pred_0 - y)^2)

#########################################################################
### Calculate Test and Training MSE for Full and Manual models

cv.errors_full = c(NA, k)
train.errors_full = c(NA, k)
```

```
cv.errors_manual = c(NA, k)
train.errors_manual = c(NA, k)

for(j in 1:k) {
   OLS.full = lm(log.crim ~ ., data=mydata3[folds!=j,])
   train.errors_full[j] = sum(OLS.full$residuals^2) / length(OLS.full$residuals)

   OLS.manual = lm(log.crim ~ zn + nox + rad + ptratio + exp.black + age + lstat,
                   data=mydata3[folds!=j,])
   train.errors_manual[j] = sum(OLS.manual$residuals^2) / length(OLS.manual$residuals)

   y.test = mydata3$log.crim[folds==j]

   pred_full = predict(OLS.full, newdata=mydata3[folds==j,])
   pred_manual = predict(OLS.manual, newdata=mydata3[folds==j,])

   cv.errors_full[j] = mean((pred_full - y.test)^2)
   cv.errors_manual[j] = mean((pred_manual - y.test)^2)
}

test.MSE_full = mean(cv.errors_full)
train.MSE_full = mean(train.errors_full)
test.MSE_manual = mean(cv.errors_manual)
train.MSE_manual = mean(train.errors_manual)

###################################
### Make summary table
###################################

# Table of model coefficients
coef.table = matrix(NA, 21, 7, dimnames=list(names(coef(full.lm1)), c("Full_model", "Manual", "Best_sub",
 "Ridge(min)", "Ridge(1se)", "Lasso(min)", "Lasso(1se)")))

coef.table[, 1] = full.lm1$coefficients
coef.table[names(manual.pick$coefficients),2 ] = manual.pick$coefficients
coef.table[names(coef(reg.best, index.best)), 3] = coef(reg.best, index.best)
coef.table[names(coef.bestlam_ridge), 4] = coef.bestlam_ridge
coef.table[names(coef.bestlam_ridge.1se), 5] = coef.bestlam_ridge.1se
coef.table[names(coef.bestlam_lasso), 6] = coef.bestlam_lasso
coef.table[names(coef.bestlam_lasso.1se), 7] = coef.bestlam_lasso.1se

coef.table[coef.table == 0] = NA

# MSE with all data
MSE_full = sum(full.lm1$residuals^2) / nrow(mydata3)
MSE_manual = sum(manual.pick$residuals^2) / nrow(mydata3)
MSE_bestsub = summary(reg.best)$rss[index.best]/reg.best$nn

MSE_all.data = c(MSE_full, MSE_manual, MSE_bestsub, MSE_ridge.min, MSE_ridge.1se,
                 MSE_lasso.min, MSE_lasso.1se)

# Test MSE
test.MSE = c(test.MSE_full, test.MSE_manual, test.MSE_bestsub, test.MSE_ridge.min,
             test.MSE_ridge.1se, test.MSE_lasso.min, test.MSE_lasso.1se)

# Join coefficient table and metrics
summary.table = rbind(coef.table, MSE_all.data, test.MSE)
rownames(summary.table)[22:23] = c("MSE(all_data)","Test_MSE")
```

```
# Table 8: Summary table of model metric and coefficients
summary.table = round(summary.table, digits=4)
kable(summary.table, caption="Table 8: Summary table of model metric and coefficients", format="html", ta
ble.attr = "style='width:90%;'")

#######################################
### DISCUSSION: Table for Best subsets

BS.table = matrix(0, 20, 21, dimnames=list(c(1:20), names(coef(full.lm1))))
BS.coef = coef(reg.best, c(1:npred))

for(i in c(1:20)) {
   BS.table[i,names(BS.coef[[i]])] = BS.coef[[i]]
}

BS.table_df = data.frame(BS.table)
BS.table_df$np = c(1:20)
colnames(BS.table_df)[1] = "intercept"
colnames(BS.table_df)[4] = "chasOth"

melt.BS.table = melt(BS.table_df, id.vars=c("np"))

# Figure 20: Coefficients vs. number of predictors for Best subsets

ggplot(data=melt.BS.table) +
   geom_line(aes(x=np, y=value, color=variable)) +
   scale_x_continuous(trans="reverse", breaks=c(1:20)) +
   labs(x="Number of predictors", y="Coefficients")

############################################################
### Top 5 pick by Best Subsets and Lasso Regression

# Best Subsets
BS.best5.coef = coef(reg.best, c(1:5))
BS.best5.MSE = summary.BS$rss[1:5] / reg.best$nn

# Lasso Regression
grid = 10^seq(0.3, -0.3, length=100)
lambda.pick = grid[c(15, 19, 83, 87, 88)] # Picked lambda for each predictor number

out_lasso5 = glmnet(x, y, alpha=1, lambda=grid)
out_lasso5.summary = summary(out_lasso5)

Lasso.best5 = list()
Lasso.best5.MSE = rep(NA, length(lambda.pick))

for(i in c(1:length(lambda.pick))) {
   Lasso.best5[[i]] = coef(out_lasso5, s=lambda.pick[i], exact=T, x=x, y=y)[,1]
   pred = predict(out_lasso5, s=lambda.pick[i], newx=x, exact=T, x=x, y=y)
   Lasso.best5.MSE[i] = mean((pred - y)^2)
}
# Table of model coefficients
feature.select = matrix(NA, 21, 10, dimnames=list(names(coef(full.lm1)), c("Best(n=1)","Lasso(n=1)","Best
(n=2)","Lasso(n=2)","Best(n=3)","Lasso(n=3)","Best(n=4)","Lasso(n=4)","Best(n=5)","Lasso(n=5)")))

for(i in c(1:length(lambda.pick))) {
   feature.select[names(BS.best5.coef[[i]]),2*i-1] = BS.best5.coef[[i]]
   feature.select[names(Lasso.best5[[i]]),2*i] = Lasso.best5[[i]]
}
```

```
feature.select[feature.select == 0] = NA

kable(feature.select, caption="Table 9: Selected coefficients for first 5 predictors from Best subsets an
d Lasso regression",
      format="html", table.attr = "style='width:100%;'")
```