Ridge Regression and the Lasso

Maureen Renaud

6/15/2020

INTRODUCTION

Police departments and government officials seek to predict crime rates so that they are better equipped to combat that crime and more efficiently allocate personnel and resources.
In order to ensure accurate predictions, it is important to ensure that we have high quality data and that we are using appropriate models.
This study endeavors to predict per capita crime rate in the Boston area. To ensure we have the highest quality prediction, I considered a linear regression model with variables selected via best subset selection, a ridge regression model, and a lasso model. After creating the models with training sets and evaluating them on test sets, I assessed each model based on its Mean Square Error (MSE). The linear regression model had an MSE of 0.61, the ridge regression model had an MSE of 0.64, and the lasso model had a MSE of 0.67. Because the MSE values are nearly identical, I would be confident using any of these models to predict per capita crime rate.

In summary, the set of appropriate models includes the eight variable linear regression model, the ridge regression model, and the lasso model.

DATA

The data for this study was obtained from the Boston dataset in the MASS library. The variables are broken down as follows:

CRIM - per capita crime rate by town

ZN - proportion of residential land zoned for lots over 25,000 square feet

INDUS - proportion of non-retail business acres per town

CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)

NOX - nitric 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 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)ˆ2 where Bk is the proportion of blacks by town

LSTAT - % lower status of the population

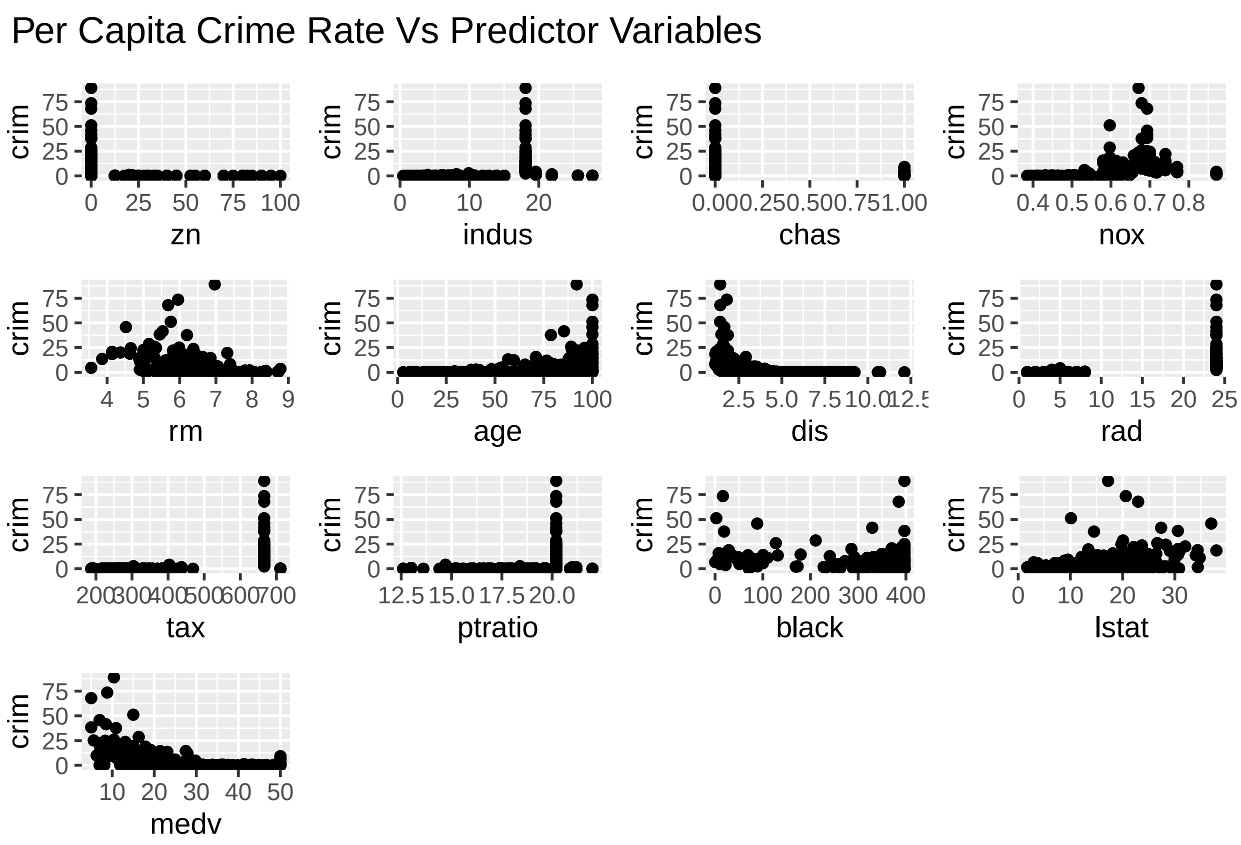
MEDV - Median value of owner-occupied homes in $1000's

To ensure an accurate prediction, it is essential that we are using complete, quality data. So, we will first examine the data. A check of the data reveals that we have a complete set. For further information, a complete summary of the variables can be found in Figure 1.

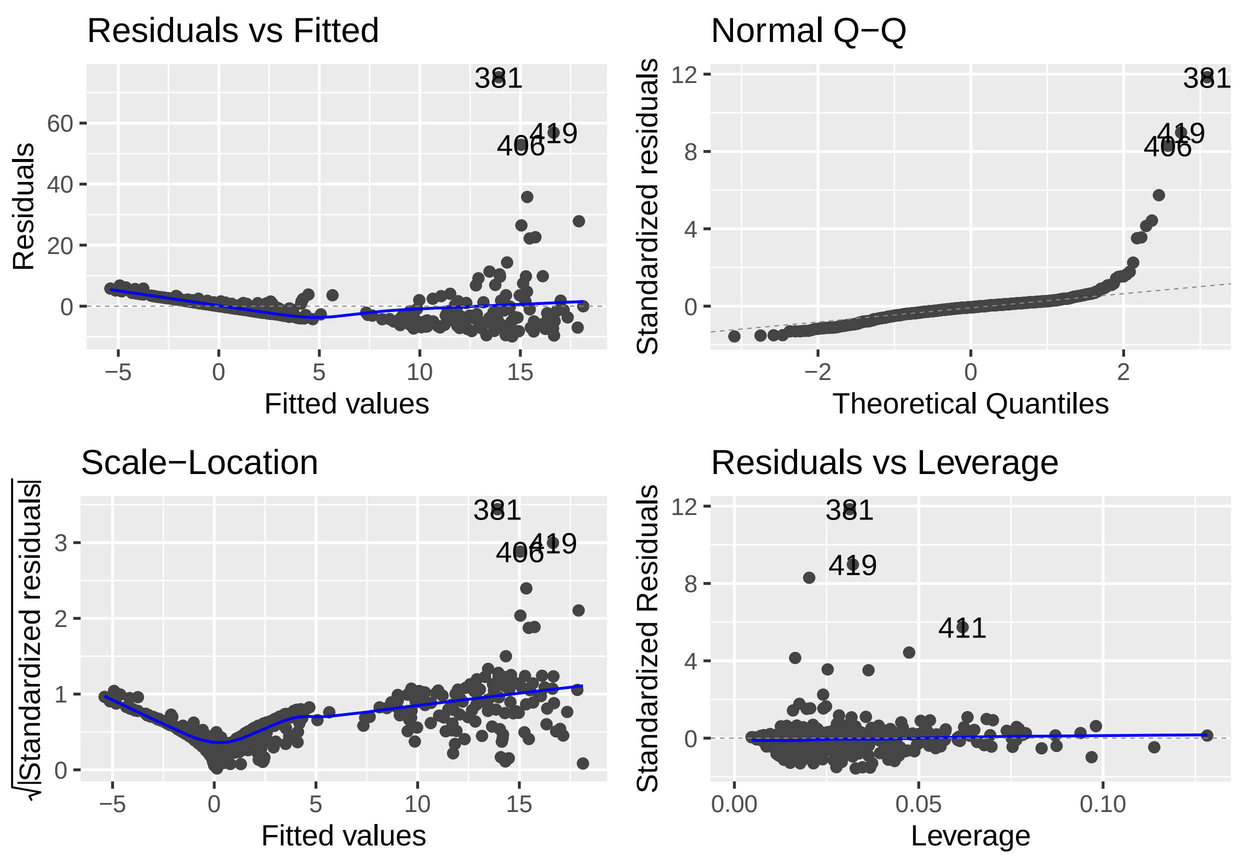
Now, I will take a closer look at the relationships between the data points. If I want to compare linear regression, ridge regression, and the lasso I will need to first set up a linear regression as seen below:

```
linear.boston <- lm(
  crim ~  zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black
  + lstat + medv, data = Boston
  )
```

We can examine scatterplots of the relationship between per capita crime rate and each of the predictors:



There does not appear to be evidence of strong linear relationships between per capita crime and many of the predictors, so a transformation will be in order. Before moving on, we will examine the residual plots:



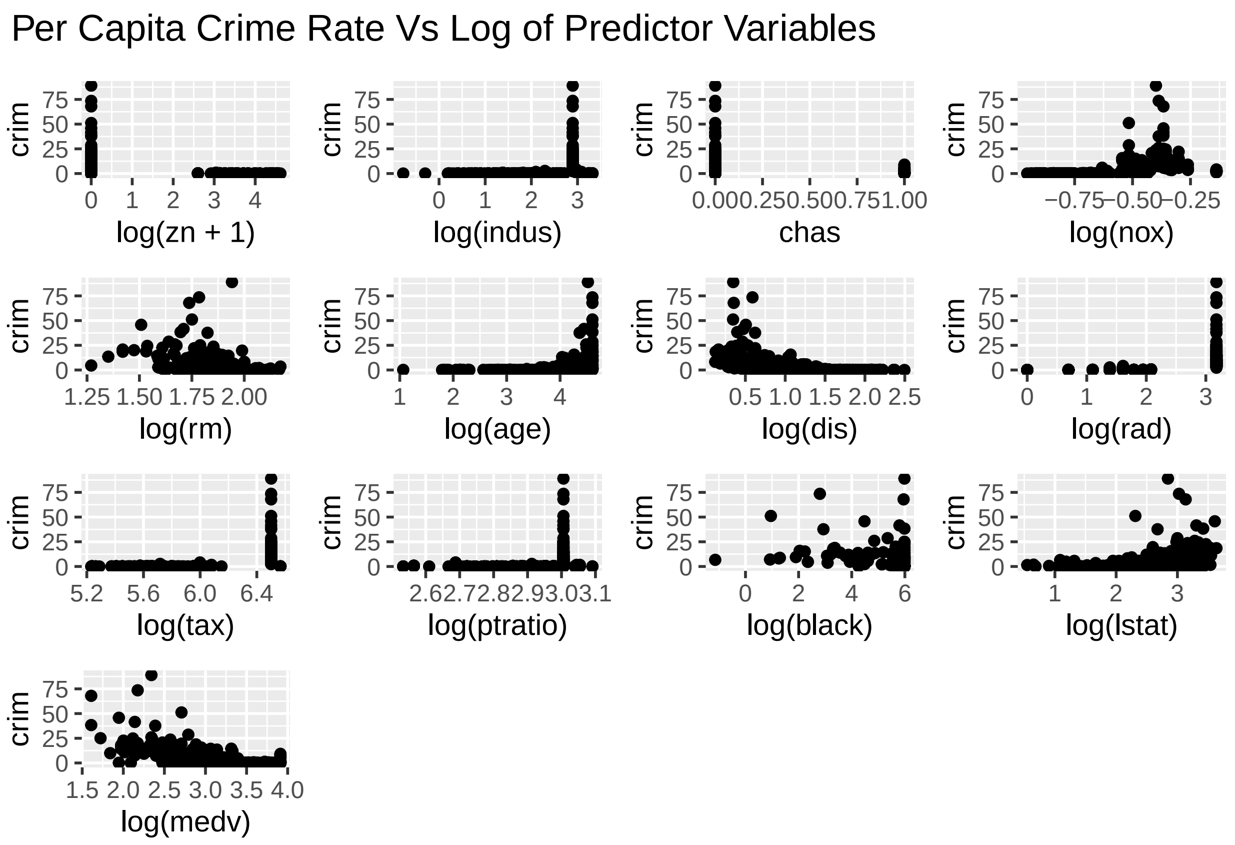
These residual plots show clear violations of the assumptions of linear regression. In the Residuals vs. Fitted plot, you can see that the points do not bounce and down nicely around the zero value indicating that the relationship is not linear. Additionally, the points spread out more as you move on down the line indicating that they fail to meet the assumption of homoscedasticity. Looking at the Normal Q-Q plot, the points fail to follow the dashed line, indicating that the residuals fail to meet the assumption of normality.

To rectify this, I will first explore taking a log transformation of the predictors. I took the log of each predictor with the except of zn where I took the log of ($zn + 1$) because of the zeroes in this variable. I chose not to take a log transformation of *chas* because it is a dummy variable coded with 0 and 1.

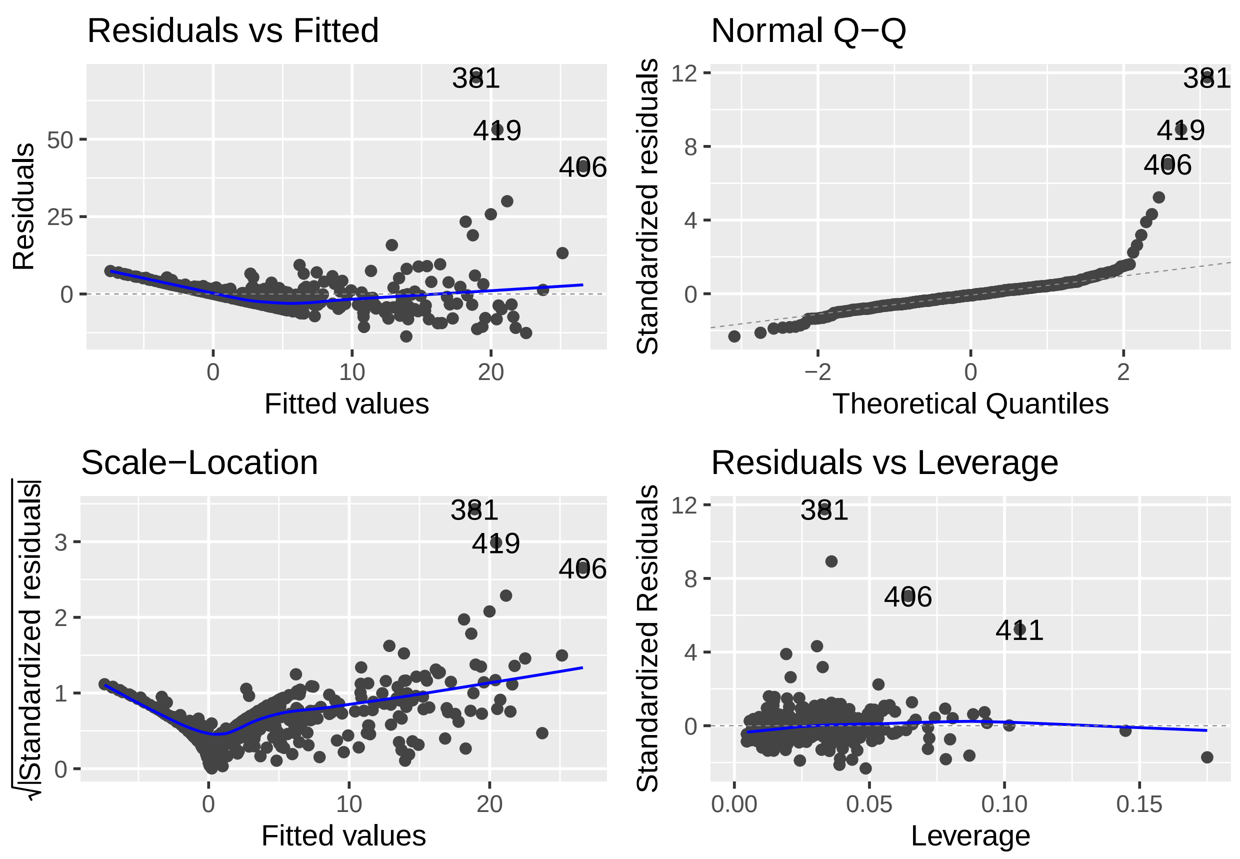
Here is the model:

```r
linear.boston.1 <- lm(
  crim ~  log(zn + 1) + log(indus) + chas + log(nox) + log(rm) + log(age) + log(dis)
  + log(rad) + log(tax) + log(ptratio) + log(black) + log(lstat) + log(medv),
  data = Boston
  )
```

And here is a plot of the relationships between per capita crime and the predictors:



These linear relationships do not appear significantly stronger than the relationships we saw in the previous model. Before trying another transformation, I will examine the residual plots:

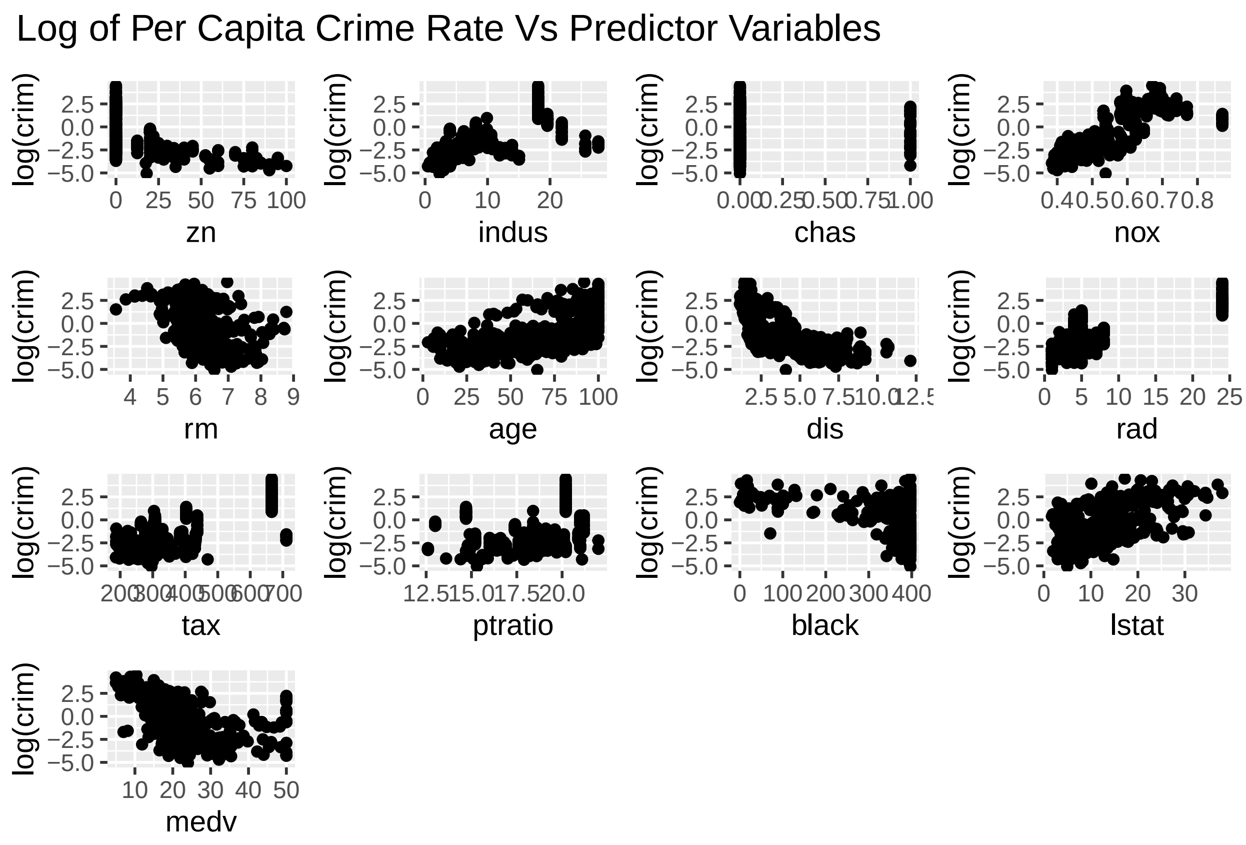


It appears that this model still violates the assumptions of linear models. In the Residuals vs. Fitted plot, you can see that the points do not bounce and down nicely around the zero value indicating that the relationship is not linear. Additionally, the points spread out more as you move on down the line indicating that they fail to meet the assumption of homoscedasticity. Looking at the Normal Q-Q plot, the points fail to follow the dashed line, indicating that the residuals fail to meet the assumption of normality.

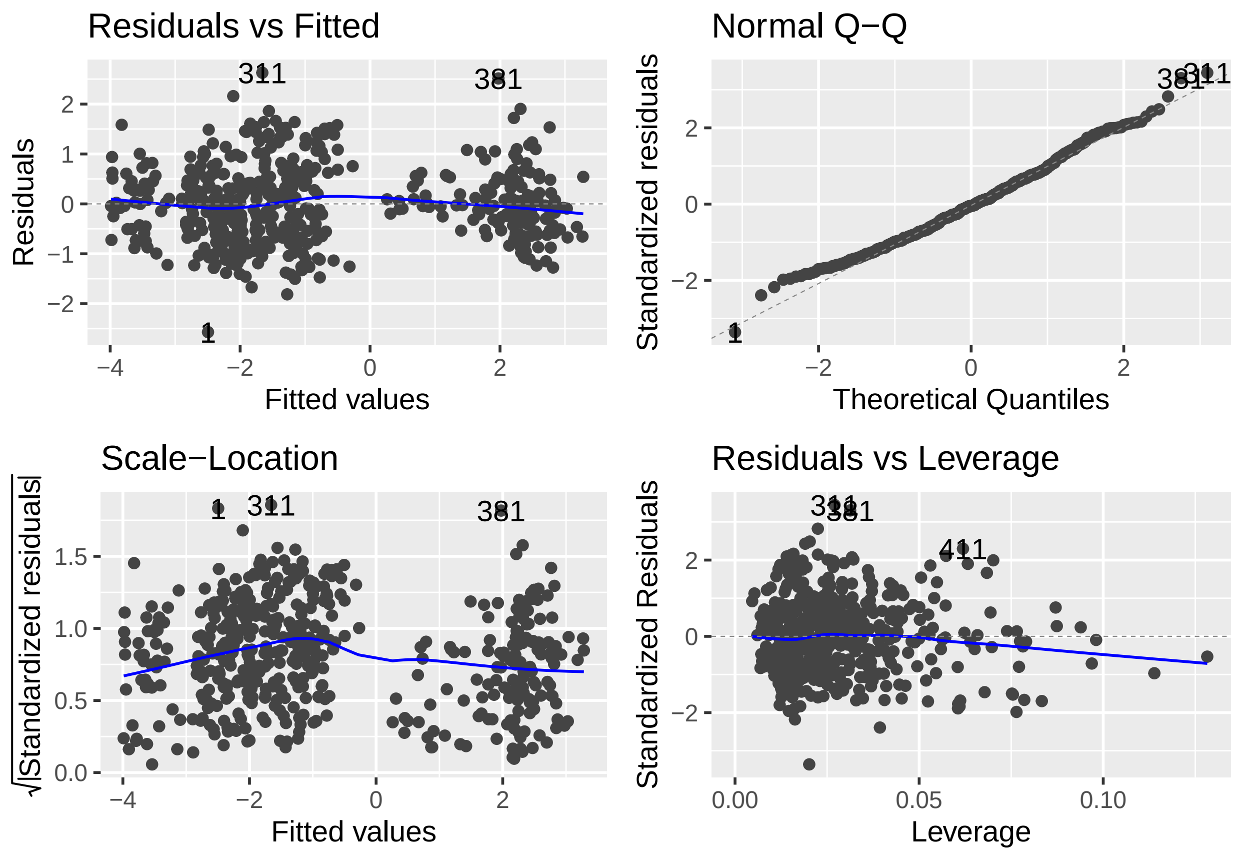
I will now transform per capita crime by taking the natural log of the predictor *crim* and leaving the predictors as they are:

```
linear.log.boston <- lm(
  log(crim) ~  zn + indus + chas + nox + rm + age + dis + rad + tax
                        + ptratio + black + lstat + medv, data = Boston
  )
```

We first examine the scatterplots showing the relationships between *crim* and the various predictors:

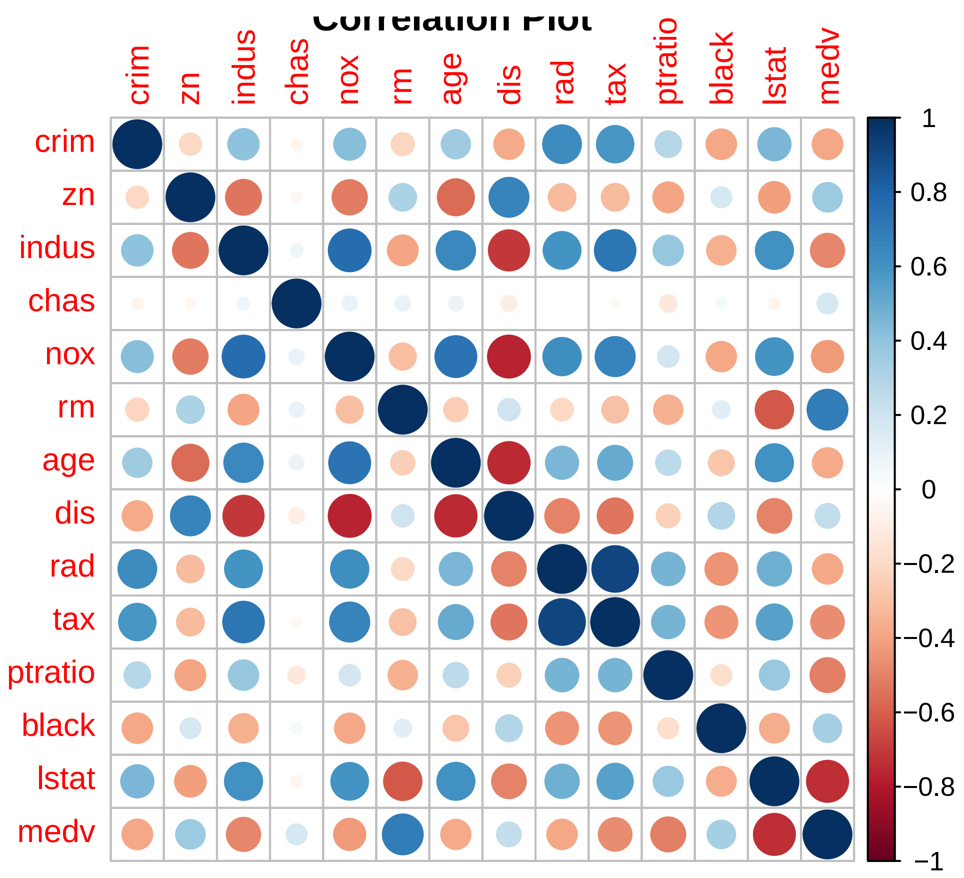


The linear relationships in these plots look significantly stronger than the relationships we saw in our previous transformations. They all appear roughly linear, with the exception of *chas*. However, this is a dummy variable, so this is not a surprise. To confirm this transformation, we will take a look at our resiudal plots:



The residual plots also look significiantly better. In the Residuals vs Fitted plot,the points now appear to center around the zero line indicating a linear relationship. The points also appear to maintain a roughly equal distance from that zero line, indicating homoscedasticity. Looking at the Normal Q-Q plot, the points roughly follow the dashed line indicating that the residuals are normally distributed.

The next issue to consider is multicollinearity. I will begin with examining a visual representation of the correlation between the variables:



It definitely appears that there are strong correlations between some of the variables. To get a more specific idea of what we are dealing with, I will examine the Variance Inflation Factors (VIF) from the model:

Predictor	VIF
zn	2.325094
indus	3.987753
chas	1.094326
nox	4.551563
rm	2.258113
age	3.100801
dis	4.289041
rad	7.158834
tax	9.195495
ptratio	1.984489
black	1.369741
lstat	3.561476
medv	3.772856

A VIF of 1 indicates no correlation. A VIF between 1 and 5 indicates moderate correlation and a VIF greater than 5 indicates significant correlation. With values ranging from 1.09 to 9.19, it appears that most of our predictors are moderately to significantly correlated with another variable.

However this is not necessarily a problem. Just because the predictor variables are correlated does not mean that we can not fit an accurate model or use it to predict new observations. We would struggle to make accurate inferences about the impact of any particular predictor. Since the goal of this study is to predict per capita crime rate and not to make inferences about the impact of any given predictor, I will not try to correct for the multicollinearity.

Now that the necessary transformations have been made, we can move on to comparing the linear model selected through best subset selection, the ridge regression model, and the lasso model.

ANALYSES

We now move on to generating the models for comparison. First, I will determine the best linear regression from this data set based on best subset selection.
Below is the summary with the best model at each level of i predictors:

```r
library (leaps)
regfit.full=regsubsets(log(crim) ~ ., Boston, nvmax = 13)
reg.summary <- summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(log(crim) ~ ., Boston, nvmax = 13)
## 13 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: exhaustive
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " " 
## 2  ( 1 )  " " " "   " "  "*" " " " " " " "*" " " " "     " "   " "   " " 
## 3  ( 1 )  "*" " "   " "  "*" " " " " " " "*" " " " "     " "   " "   " " 
## 4  ( 1 )  "*" " "   " "  "*" " " " " " " "*" " " " "     " "   "*"   " " 
## 5  ( 1 )  "*" " "   " "  "*" " " " " " " "*" " " " "     "*"   "*"   " " 
## 6  ( 1 )  "*" " "   " "  "*" " " "*" " " "*" " " " "     "*"   "*"   " " 
## 7  ( 1 )  "*" " "   " "  "*" " " "*" " " "*" " " "*"     "*"   "*"   " " 
## 8  ( 1 )  "*" "*"   " "  "*" " " "*" " " "*" " " "*"     "*"   "*"   " " 
## 9  ( 1 )  "*" "*"   " "  "*" " " "*" " " "*" " " "*"     "*"   "*"   "*" 
## 10  ( 1 ) "*" "*"   " "  "*" "*" "*" " " "*" " " "*"     "*"   "*"   "*" 
## 11  ( 1 ) "*" "*"   "*"  "*" "*" "*" " " "*" " " "*"     "*"   "*"   "*" 
## 12  ( 1 ) "*" "*"   "*"  "*" "*" "*" " " "*" "*" "*"     "*"   "*"   "*" 
## 13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
```

To find the best model from each of these best models, I will now set up my training and test sets and use cross-validation to find the best model.

Computing the test MSE for each i variable model gives us:

```
val.errors
```

```
##  [1] 1.1889529 0.7383462 0.6914392 0.6466656 0.6472687 0.6239908 0.6239356
##  [8] 0.6122171 0.6158331 0.6171000 0.6184207 0.6146669 0.6157659
```

```
coef(regfit.best, which.min(val.errors))
```

```
##  (Intercept)           zn        indus          nox          age          rad
## -3.757598429 -0.012178115  0.022427377  3.845516663  0.007502586  0.135911611
##      ptratio        black        lstat
## -0.043533111 -0.001446271  0.019883695
```

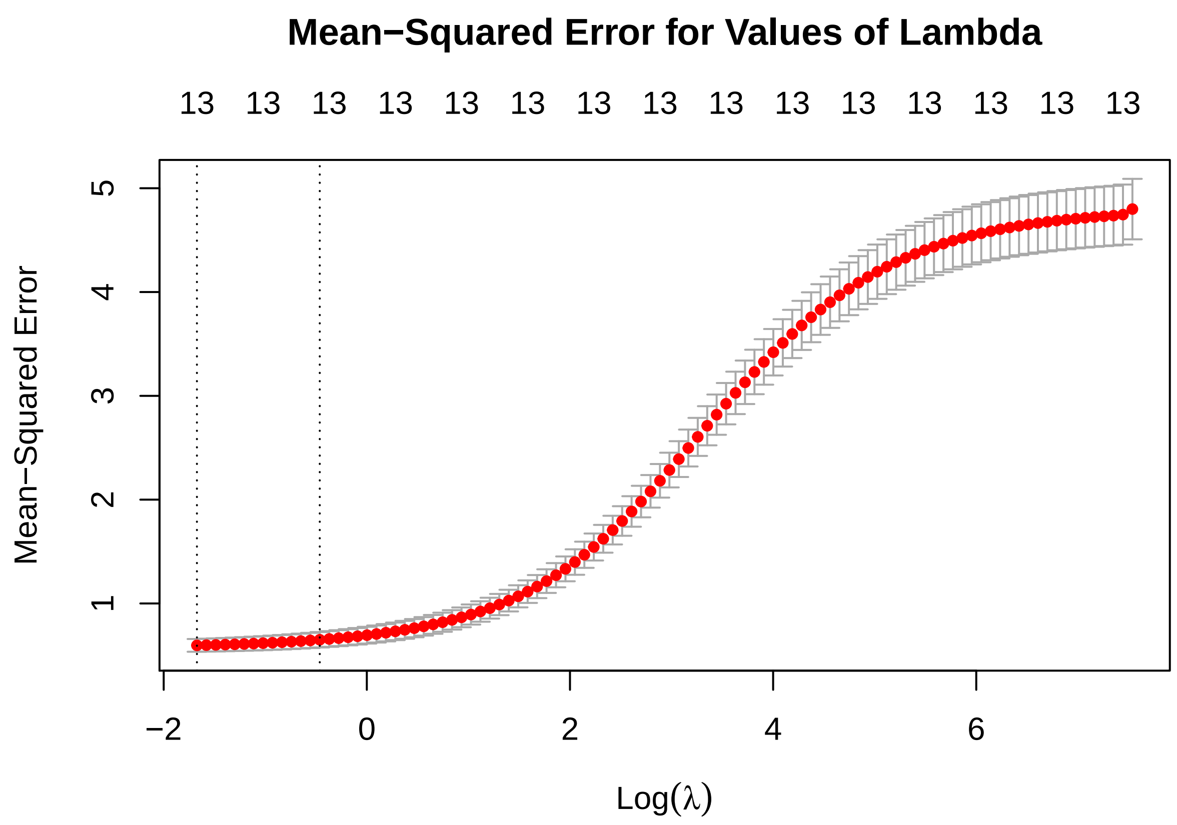
The best subset model picked a model with 8 predictors and an MSE of 0.612. The variables that did not make the cut were *chas*, *rn*, *dis*, *tax*, and *medv*. It's interesting to note that *dis*, *tax*, and *medv* had high VIF values when we previously calculated them. They were likely eliminated due to that multicollinearity. If I check out VIF values now, I find:

Predictor	VIF
zn	1.782798
indus	3.012089
nox	3.999845
age	2.631361
rad	2.261142
ptratio	1.689929
black	1.311225
lstat	1.967032

The VIFs have all decreased and we no longer have any values above five. So, choosing an appropriate model eliminated the most severe multicollinearity in the model.

Ridge Regression I will now evaluate the ridge regression model. I retain the log transformed per capita crime rate, fit a ridge regression model, and split my data into a training set and test set.

I then use cross-validation to choose the tuning parameter, lambda. The plot below displays the MSE for the various possibilities of lambda.



From this, the best lambda is calculated as:

```
bestlam =cv.out$lambda.min
bestlam
```

```
## [1] 0.1877345
```

Now I can fit the ridge regression model with a lambda of 0.1877345 and calculate the MSE for the ridge regression model:

```
ridge.pred = predict(ridge.mod, s=bestlam, newx = Boston.pred[test.ridge ,])
mean((ridge.pred -Boston.resp.test.ridge)^2)
```

```
## [1] 0.6417752
```

Finally, here are the coefficients for the ridge regression model:

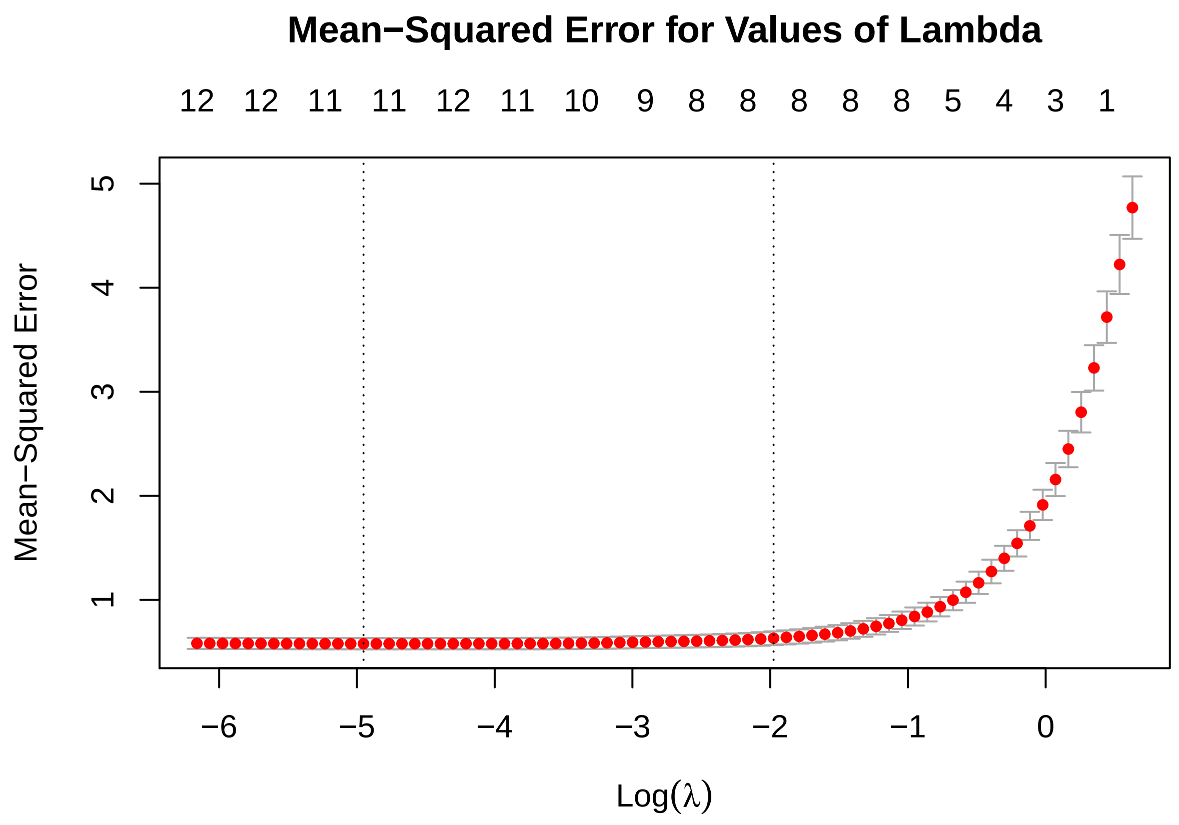
```
out = glmnet(Boston.pred, Boston.resp, alpha =0)
predict (out ,type = "coefficients", s=bestlam )[1:14 ,]
```

```
##  (Intercept)           zn        indus         chas          nox           rm 
## -4.046090714 -0.010301373  0.014428303  0.018484950  3.451887913 -0.019762582 
##          age          dis          rad          tax      ptratio        black 
##  0.005848858 -0.043923384  0.102154175  0.001622608 -0.020186289 -0.001689756 
##        lstat         medv 
##  0.029637920  0.007017473
```

The ridge regression picked all variables because the model doesn't allow for the elimination of variables, with an MSE of 0.64. This MSE is comparable than the model selected by best subset selection. The main difference between this model and the model developed with bestsubset selection, is that this model contains all variables instead of eight. However, as was stated, the ridge regression does not eliminate variables.

Next, we will develop a model with the lasso.

Lasso First, I will fit the lasso model and use the training and test sets to choose the best tuning parameter, lambda:



The best lambda for the lasso model is:

```
bestlam.lasso = lasso.cv.out$lambda.min
bestlam.lasso
```

```
## [1] 0.007068084
```

Inserting the best lambda of 0.007 into lasso model, gives us an MSE of 0.67.

```
lasso.pred = predict(lasso.mod ,s=bestlam.lasso ,newx=Boston.pred[test.ridge ,])
mean((lasso.pred -Boston.resp.test.ridge)^2)
```

```
## [1] 0.6722014
```

And the lasso model shown below:

```
out.lasso=glmnet(Boston.pred, Boston.resp,alpha =1, lambda =grid)
lasso.coef=predict(out.lasso, type = "coefficients", s=bestlam.lasso )[1:14 ,]
lasso.coef
```

```
##  (Intercept)           zn        indus         chas          nox           rm
## -3.990634216 -0.010967887  0.017559791  0.000000000  3.863140867 -0.002457205
##          age          dis          rad          tax      ptratio        black
##  0.005652659 -0.015930717  0.139254029  0.000000000 -0.033208368 -0.001368762
##        lstat         medv
##  0.028672802  0.004134684
```

You can see in this model, that all variables are maintained except for *tax* and *chas*. It's interesting to note that *tax* had the highest VIF in the original model, so it likely added nothing new to the model that wasn't already supplied by the other variables. By eliminating such a highly correlated variable, we reduce multicollinearity. *chas* did not have a high VIF, but it was also eliminated from the eight-variable linear regression model. So it is likely that it is just not a significant predictor of per capita crime rate.

PLOTS AND TABLES

```
library(MASS)
library(stargazer)
cols <- c("crim", "zn", "indus", "chas",  "nox",  "rm", "age",
          "dis", "rad", "tax", "ptratio", "black", "lstat",   "medv")
stargazer(
    Boston[, cols], type = "text",
    summary.stat = c("min", "p25", "median", "p75", "max", "median", "sd"),
    title = "Figure 1: Summary Statistics for the Boston Data Set"
)
```

```
##
## Figure 1: Summary Statistics for the Boston Data Set
## ===================================================================
## Statistic  Min   Pctl(25) Median  Pctl(75)   Max   Median  St. Dev.
## -------------------------------------------------------------------
## crim      0.006   0.082    0.257   3.677   88.976  0.257    8.602
## zn          0       0        0     12.5     100      0     23.322
## indus     0.460   5.190    9.690   18.100  27.740  9.690    6.860
## chas        0       0        0       0        1      0      0.254
## nox       0.385   0.449    0.538   0.624    0.871  0.538    0.116
## rm        3.561   5.886    6.208   6.624    8.780  6.208    0.703
## age       2.900   45.025  77.500   94.075  100.000 77.500  28.149
## dis       1.130   2.100    3.207   5.188   12.127  3.207    2.106
## rad         1       4        5      24       24      5      8.707
## tax        187     279      330     666      711     330   168.537
## ptratio   12.600  17.400  19.050   20.200  22.000  19.050   2.165
## black     0.320  375.377  391.440 396.225  396.900 391.440  91.295
## lstat     1.730   6.950   11.360   16.955  37.970  11.360   7.141
## medv        5      17.0    21.2     25       50     21.2    9.197
## -------------------------------------------------------------------
```

Figure 1

CONCLUSION

In order to predict per capita crime rate, we require a reliable model. This study compared a linear model selected with best subset selection, a ridge regression model, and a lasso model based on Mean Standard Error. With MSEs of 0.61, 0.64, and 0.67 I can not state that one model is unequivocally better than another.
Thus, to predict per capita crime rate, I propose the set of models including the eight-variable linear regression model, the ridge regression model, and the lasso model.

SOURCES

1) James, G et al. (2017). *An Introduction To Statistical Learning.* New York, NY.

APPENDIX

The below code is a complete record of all R code used in this study:

```r
RNGkind(sample.kind="Rounding")

#Load the MASS library in order to access the Boston data set

library(MASS)

#Calculate the NAs in the Boston data set

sum(is.na(Boston))

#Set up the initial linear model with response variable crim

linear.boston <- lm(
  crim ~  zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black
  + lstat + medv, data = Boston
  )

# Set up the scatterplots of the relationship between crim and the predictor variables
library(ggplot2)
zn.scatter <- ggplot(Boston, aes(zn, crim)) + geom_point()
indus.scatter <- ggplot(Boston, aes(indus, crim)) + geom_point()
chas.scatter <- ggplot(Boston, aes(chas, crim)) + geom_point()
nox.scatter <- ggplot(Boston, aes(nox, crim)) + geom_point()
rm.scatter <- ggplot(Boston, aes(rm, crim)) + geom_point()
age.scatter <- ggplot(Boston, aes(age, crim)) + geom_point()
dis.scatter <- ggplot(Boston, aes(dis, crim)) + geom_point()
rad.scatter <- ggplot(Boston, aes(rad, crim)) + geom_point()
tax.scatter <- ggplot(Boston, aes(tax, crim)) + geom_point()
ptratio.scatter <- ggplot(Boston, aes(ptratio, crim)) + geom_point()
black.scatter <- ggplot(Boston, aes(black, crim)) + geom_point()
lstat.scatter <- ggplot(Boston, aes(lstat, crim)) + geom_point()
medv.scatter <- ggplot(Boston, aes(medv, crim)) + geom_point()

#Display all scatterplots in the same image
```

```r
library(cowplot)
subset.plot.row <- plot_grid(
  zn.scatter, indus.scatter, chas.scatter, nox.scatter, rm.scatter,
  age.scatter, dis.scatter, rad.scatter, tax.scatter, ptratio.scatter,
  black.scatter, lstat.scatter, medv.scatter, align = 'h'
  )

title <- ggdraw() +
  draw_label(
    "Per Capita Crime Rate Vs Predictor Variables",
    x = 0,
    hjust = 0
  ) +
  theme(
    plot.margin = margin(0, 0, 0, 7)
  )
plot_grid(
  title, subset.plot.row,
   ncol = 1,
  rel_heights = c(0.1, 1)
)

#Use ggplot to display the residual plots

library(ggfortify)
autoplot(linear.boston)

#Examine a model where the predictors are transformed

linear.boston.1 <- lm(
  crim ~  log(zn + 1) + log(indus) + chas + log(nox) + log(rm) + log(age) + log(dis)
  + log(rad) + log(tax) + log(ptratio) + log(black) + log(lstat) + log(medv),
  data = Boston
  )

#Display the scatterplots of the relationship between crim and the log of
#the predictors

library(ggplot2)
zn.scatter.1 <- ggplot(Boston, aes(log(zn + 1), crim)) + geom_point()
indus.scatter.1 <- ggplot(Boston, aes(log(indus), crim)) + geom_point()
chas.scatter.1 <- ggplot(Boston, aes(chas, crim)) + geom_point()
nox.scatter.1 <- ggplot(Boston, aes(log(nox), crim)) + geom_point()
rm.scatter.1 <- ggplot(Boston, aes(log(rm), crim)) + geom_point()
age.scatter.1 <- ggplot(Boston, aes(log(age), crim)) + geom_point()
dis.scatter.1 <- ggplot(Boston, aes(log(dis), crim)) + geom_point()
rad.scatter.1 <- ggplot(Boston, aes(log(rad), crim)) + geom_point()
tax.scatter.1 <- ggplot(Boston, aes(log(tax), crim)) + geom_point()
ptratio.scatter.1 <- ggplot(Boston, aes(log(ptratio), crim)) + geom_point()
black.scatter.1 <- ggplot(Boston, aes(log(black), crim)) + geom_point()
lstat.scatter.1 <- ggplot(Boston, aes(log(lstat), crim)) + geom_point()
medv.scatter.1 <- ggplot(Boston, aes(log(medv), crim)) + geom_point()
```

```r
#Display all scatterplots in one image

library(cowplot)
subset.plot.row.1 <- plot_grid(
  zn.scatter.1, indus.scatter.1, chas.scatter.1, nox.scatter.1, rm.scatter.1,
  age.scatter.1, dis.scatter.1, rad.scatter.1, tax.scatter.1, ptratio.scatter.1,
  black.scatter.1, lstat.scatter.1, medv.scatter.1, align = 'h'
  )

title <- ggdraw() +
  draw_label(
    "Per Capita Crime Rate Vs Log of Predictor Variables",
    x = 0,
    hjust = 0
  ) +
  theme(
    plot.margin = margin(0, 0, 0, 7)
  )
plot_grid(
  title, subset.plot.row.1,
   ncol = 1,
  rel_heights = c(0.1, 1)
)

#Use ggplot to display the residual plots

autoplot(linear.boston.1)

#Create the model with the log of crim and the untransformed predictors

linear.log.boston <- lm(
  log(crim) ~  zn + indus + chas + nox + rm + age + dis + rad + tax
                        + ptratio + black + lstat + medv, data = Boston
  )

#Create the scatterplots showing the relationship between the log of crim and
#the predictors

zn.scatter.log <- ggplot(Boston, aes(zn, log(crim))) + geom_point()
indus.scatter.log <- ggplot(Boston, aes(indus, log(crim))) + geom_point()
chas.scatter.log <- ggplot(Boston, aes(chas, log(crim))) + geom_point()
nox.scatter.log <- ggplot(Boston, aes(nox, log(crim))) + geom_point()
rm.scatter.log <- ggplot(Boston, aes(rm, log(crim))) + geom_point()
age.scatter.log <- ggplot(Boston, aes(age, log(crim))) + geom_point()
dis.scatter.log <- ggplot(Boston, aes(dis, log(crim))) + geom_point()
rad.scatter.log <- ggplot(Boston, aes(rad, log(crim))) + geom_point()
tax.scatter.log <- ggplot(Boston, aes(tax, log(crim))) + geom_point()
ptratio.scatter.log <- ggplot(Boston, aes(ptratio, log(crim))) + geom_point()
black.scatter.log <- ggplot(Boston, aes(black, log(crim))) + geom_point()
lstat.scatter.log <- ggplot(Boston, aes(lstat, log(crim))) + geom_point()
medv.scatter.log <- ggplot(Boston, aes(medv, log(crim))) + geom_point()

#Combine all scatterplots into one image
```

```r
library(cowplot)
subset.plot.row.log <- plot_grid(
  zn.scatter.log, indus.scatter.log, chas.scatter.log, nox.scatter.log, rm.scatter.log,
  age.scatter.log, dis.scatter.log, rad.scatter.log, tax.scatter.log, ptratio.scatter.log,
  black.scatter.log, lstat.scatter.log, medv.scatter.log, align = 'h'
  )

title <- ggdraw() +
  draw_label(
    "Log of Per Capita Crime Rate Vs Predictor Variables",
    x = 0,
    hjust = 0
  ) +
  theme(
    plot.margin = margin(0, 0, 0, 7)
  )
plot_grid(
  title, subset.plot.row.log,
   ncol = 1,
  rel_heights = c(0.1, 1)
)

#Use ggplot to display all scatterplots in one image

autoplot(linear.log.boston)

#Create the correlation plot between the variables in the Boston data set

library(corrplot)
corr.Boston <- cor(Boston)
corrplot(corr.Boston, title = "Correlation Plot")

#Display the VIF values for the predictors in the chosen model

library(car)
vif(linear.log.boston)

# Passing in the X matrix and Y vector to use in ridge regression and the lasso

Boston.pred = model.matrix(crim ~ ., Boston )[,-1]
Boston.resp = log(Boston$crim)

#Fit the model for best subset selection

library (leaps)
regfit.full=regsubsets(log(crim) ~ ., Boston, nvmax = 13)
reg.summary <- summary(regfit.full)
reg.summary

# Set up training and test sets

set.seed (1)
train=sample (c(TRUE ,FALSE), nrow(Boston),rep=TRUE)
```

```r
test =(!train )

#Run the model on the training set

regfit.best=regsubsets (log(crim) ~.,data = Boston[train ,],
nvmax =13)

#Test the model with the test set

test.mat = model.matrix(log(crim) ~ .,data = Boston[test ,])


#Computing the test MSE for each _i_ variable model gives us:

val.errors =rep(NA ,13)
for(i in 1:13){
coefi=coef(regfit.best ,id=i)
pred=test.mat[,names(coefi)]%*% coefi
val.errors[i]= mean(( log(Boston$crim)[test]-pred)^2)
}

# Calculate the MSEs for each model

val.errors

#Find the coefficients for the best model

coef(regfit.best, which.min(val.errors))

#Examine the VIF values for the selected model

vif(lm(log(crim)~ zn + indus + nox + age + rad + ptratio + black + lstat, data = Boston))


#Fit a ridge regression model

library (glmnet)
grid = 10^seq(10,-2, length =100)
ridge.mod = glmnet(Boston.pred, Boston.resp ,alpha = 0, lambda = grid)


# Split data into a training set and test set.

set.seed(1)
train.ridge = sample (1: nrow(Boston.pred), nrow(Boston.pred)/2)
test.ridge =(-train.ridge)
Boston.resp.test.ridge = Boston.resp[test.ridge]

#Displays the MSE for the various possibilities of lambda.

set.seed (1)
cv.out =cv.glmnet (Boston.pred[train.ridge ,],Boston.resp[train.ridge],alpha =0)
plot(cv.out)
```

```r
title("Mean-Squared Error for Values of Lambda", line = 3)

#Calculate the best lambda

bestlam =cv.out$lambda.min
bestlam

# Fit the ridge regression model with the best lambda
#and calculate the MSE for the ridge regression model

ridge.pred = predict(ridge.mod, s=bestlam, newx = Boston.pred[test.ridge ,])
mean((ridge.pred -Boston.resp.test.ridge)^2)

#Calculate the coefficients for the ridge regression model

out = glmnet(Boston.pred, Boston.resp, alpha =0)
predict (out ,type = "coefficients", s=bestlam )[1:14 ,]

# Fit the lasso model

lasso.mod = glmnet(Boston.pred[train.ridge ,], Boston.resp[train.ridge],alpha =1, lambda =grid)

#Show a plot of the MSE values for various lambdas

set.seed (1)
lasso.cv.out = cv.glmnet (Boston.pred[train.ridge ,], Boston.resp[train.ridge],alpha =1)
plot(lasso.cv.out)
title("Mean-Squared Error for Values of Lambda", line = 3)

#Find best lambda for the lasso model

bestlam.lasso = lasso.cv.out$lambda.min
bestlam.lasso

# Inserting the best lambda of 0.007 into lasso model and find the MSE

lasso.pred = predict(lasso.mod ,s=bestlam.lasso ,newx=Boston.pred[test.ridge ,])
mean((lasso.pred -Boston.resp.test.ridge)^2)


#Find the coefficients for the lasso model

out.lasso=glmnet(Boston.pred, Boston.resp,alpha =1, lambda =grid)
lasso.coef=predict(out.lasso, type = "coefficients", s=bestlam.lasso )[1:14 ,]
lasso.coef

#Show summary of the Boston data set

library(MASS)
library(stargazer)
cols <- c("crim", "zn", "indus", "chas",  "nox",  "rm", "age",
          "dis", "rad", "tax", "ptratio", "black", "lstat",   "medv")
stargazer(
    Boston[, cols], type = "text",
    summary.stat = c("min", "p25", "median", "p75", "max", "median", "sd"),
    title = "Figure 1: Summary Statistics for the Boston Data Set"
)
```