Week 8 Assignment - Variable Selection

Omer Farooq (EDx ID: mfarooq4)

03/04/2020

Table of Contents

QUESTION 11.1

Using the crime data set uscrime.txt from Questions 8.2, 9.1, and 10.1, build a regression model using:

1. Stepwise regression

2. Lasso

3. Elastic net

For Parts 2 and 3, remember to scale the data first – otherwise, the regression coefficients will be on different scales and the constraint won't have the desired effect.For Parts 2 and 3, use the glmnet function in R.

STEPWISE REGRESSION

First, I loaded the required libraries.

```
library(corrplot) #for correlation plot
library (caret) #for cross-validation
library(MASS) #for stepwise regression
library (leaps)
library (glmnet)
```

Next, I loaded the Crimes data and printed a sample and summary of the data. The summary of Crime column is to be noted.

```
#setting the seed so that results are the same at every run
set.seed(101)

#loading data
crimedata <- read.delim("data_11.1/uscrime.txt")
```

```r
#quick glance at the data
head(crimedata)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

```r
#basic stats of the temps data
summary(crimedata)
```

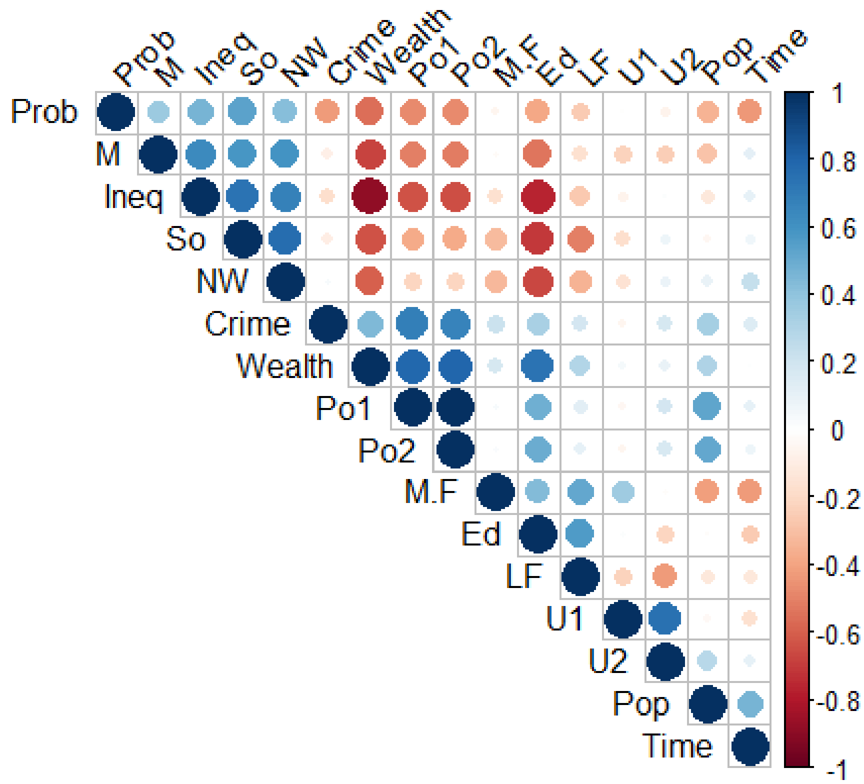
```
##        M               So               Ed             Po1       
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50  
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25  
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80  
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50  
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45  
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60  
##       Po2              LF              M.F              Pop        
##  Min.   : 4.100   Min.   :0.4800   Min.   : 93.40   Min.   :  3.00  
##  1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00  
##  Median : 7.300   Median :0.5600   Median : 97.70   Median : 25.00  
##  Mean   : 8.023   Mean   :0.5612   Mean   : 98.30   Mean   : 36.62  
##  3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50  
##  Max.   :15.700   Max.   :0.6410   Max.   :107.10   Max.   :168.00  
##        NW              U1                U2            Wealth    
##  Min.   : 0.20   Min.   :0.07000   Min.   :2.000   Min.   :2880  
##  1st Qu.: 2.40   1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595  
##  Median : 7.60   Median :0.09200   Median :3.400   Median :5370  
##  Mean   :10.11   Mean   :0.09547   Mean   :3.398   Mean   :5254  
##  3rd Qu.:13.25   3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915  
##  Max.   :42.30   Max.   :0.14200   Max.   :5.800   Max.   :6890  
##       Ineq            Prob              Time           Crime       
##  Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 342.0  
##  1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5  
##  Median :17.60   Median :0.04210   Median :25.80   Median : 831.0  
##  Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 905.1  
##  3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5  
##  Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :1993.0  
```

Before I jumped into models, I checked the pearson correlation matrix of the Crimes data. Value is 1 and -1 indicate positive and negative correlation where 0 indicates no correlation. The last column was of interest where correlation of Crime column with each predictor was given. Po1, Po2, Wealth and Prob showed some correlation to Crime column. The models built below tested these correlations further.

```
#pearson correlation matrix
corrmat <- cor(crimedata)
round(corrmat, 2)

##            M    So    Ed   Po1   Po2    LF   M.F   Pop    NW    U1    U2 Wealth
## M       1.00  0.58 -0.53 -0.51 -0.51 -0.16 -0.03 -0.28  0.59 -0.22 -0.24  -0.67
## So      0.58  1.00 -0.70 -0.37 -0.38 -0.51 -0.31 -0.05  0.77 -0.17  0.07  -0.64
## Ed     -0.53 -0.70  1.00  0.48  0.50  0.56  0.44 -0.02 -0.66  0.02 -0.22   0.74
## Po1    -0.51 -0.37  0.48  1.00  0.99  0.12  0.03  0.53 -0.21 -0.04  0.19   0.79
## Po2    -0.51 -0.38  0.50  0.99  1.00  0.11  0.02  0.51 -0.22 -0.05  0.17   0.79
## LF     -0.16 -0.51  0.56  0.12  0.11  1.00  0.51 -0.12 -0.34 -0.23 -0.42   0.29
## M.F    -0.03 -0.31  0.44  0.03  0.02  0.51  1.00 -0.41 -0.33  0.35 -0.02   0.18
## Pop    -0.28 -0.05 -0.02  0.53  0.51 -0.12 -0.41  1.00  0.10 -0.04  0.27   0.31
## NW      0.59  0.77 -0.66 -0.21 -0.22 -0.34 -0.33  0.10  1.00 -0.16  0.08  -0.59
## U1     -0.22 -0.17  0.02 -0.04 -0.05 -0.23  0.35 -0.04 -0.16  1.00  0.75   0.04
## U2     -0.24  0.07 -0.22  0.19  0.17 -0.42 -0.02  0.27  0.08  0.75  1.00   0.09
## Wealth -0.67 -0.64  0.74  0.79  0.79  0.29  0.18  0.31 -0.59  0.04  0.09   1.00
## Ineq    0.64  0.74 -0.77 -0.63 -0.65 -0.27 -0.17 -0.13  0.68 -0.06  0.02  -0.88
## Prob    0.36  0.53 -0.39 -0.47 -0.47 -0.25 -0.05 -0.35  0.43 -0.01 -0.06  -0.56
## Time    0.11  0.07 -0.25  0.10  0.08 -0.12 -0.43  0.46  0.23 -0.17  0.10   0.00
## Crime  -0.09 -0.09  0.32  0.69  0.67  0.19  0.21  0.34  0.03 -0.05  0.18   0.44
##         Ineq  Prob  Time Crime
## M       0.64  0.36  0.11 -0.09
## So      0.74  0.53  0.07 -0.09
## Ed     -0.77 -0.39 -0.25  0.32
## Po1    -0.63 -0.47  0.10  0.69
## Po2    -0.65 -0.47  0.08  0.67
## LF     -0.27 -0.25 -0.12  0.19
## M.F    -0.17 -0.05 -0.43  0.21
## Pop    -0.13 -0.35  0.46  0.34
## NW      0.68  0.43  0.23  0.03
## U1     -0.06 -0.01 -0.17 -0.05
## U2      0.02 -0.06  0.10  0.18
## Wealth -0.88 -0.56  0.00  0.44
## Ineq    1.00  0.47  0.10 -0.18
## Prob    0.47  1.00 -0.44 -0.43
## Time    0.10 -0.44  1.00  0.15
## Crime  -0.18 -0.43  0.15  1.00

#plotting the correlation matrix
corrplot(corrmat, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



We should scale the data to ensure that the variables are in the same range and results are not biased by the scale.

```
out <- c("So", "Crime") #keeping the binary variable out of the scaling
newdata <- crimedata[,!(names(crimedata) %in% out)]
scaled_data <- scale(newdata)

#binding data back together
scaled_data <- cbind(scaled_data, crimedata[,out])
head(scaled_data)

##            M         Ed        Po1        Po2         LF         M.F
## 1  0.9886930 -1.3085099 -0.9085105 -0.8666988 -1.2667456 -1.12060499
## 2  0.3521372  0.6580587  0.6056737  0.5280852  0.5396568  0.98341752
## 3  0.2725678 -1.4872888 -1.3459415 -1.2958632 -0.6976051 -0.47582390
## 4 -0.2048491  1.3731746  2.1535064  2.1732150  0.3911854  0.37257228
## 5  0.1929983  1.3731746  0.8075649  0.7426673  0.7376187  0.06714965
## 6 -1.3983912  0.3898903  1.1104017  1.2433590 -0.3511718 -0.64550313
##           Pop           NW          U1         U2     Wealth       Ineq
## 1 -0.09500679  1.943738564  0.69510600  0.8313680 -1.3616094  1.6793638
## 2 -0.62033844  0.008483424  0.02950365  0.2393332  0.3276683  0.0000000
## 3 -0.48900552  1.146296747 -0.08143007 -0.1158877 -2.1492481  1.4036474
## 4  3.16204944 -0.205464381  0.36230482  0.5945541  1.5298536 -0.6767585
## 5 -0.48900552 -0.691709391 -0.24783066 -1.6551781  0.5453053 -0.5013026
## 6 -0.30513945 -0.555560788 -0.63609870 -0.5895155  1.6956723 -1.7044289
##         Prob        Time So Crime
## 1  1.6497631 -0.05599367  1   791
## 2 -0.7693365 -0.18315796  0  1635
```

```
## 3  1.5969416 -0.32416470  1   578
## 4 -1.3761895  0.46611085  0  1969
## 5 -0.2503580 -0.74759413  0  1234
## 6 -0.5669349 -0.78996812  0   682
```

Now, I was ready to jump into the stepwise regression modeling. I first used the traincontrol and train functions and then I tried it with the StepAIC function.

The results from the train function show 6 variables that are selected for the final model.

```
set.seed(101)
control <- trainControl(method = "repeatedcv", number = 5, repeats = 5)

step_reg <- train(Crime~., data = scaled_data, method = "leapSeq",  tuneGrid
= data.frame(nvmax = 1:15), trControl = control)

step_reg$results

##    nvmax     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      1 282.9833 0.5085407 223.5931 72.32898  0.2396306 64.14550
## 2      2 283.2710 0.4959722 218.1897 78.55153  0.3023001 51.55289
## 3      3 234.8246 0.6271130 178.8537 67.73357  0.2273468 51.06538
## 4      4 267.2173 0.5318271 212.6454 53.10430  0.2437186 42.12143
## 5      5 257.6952 0.5620831 210.4947 67.09845  0.2263403 55.02587
## 6      6 231.7022 0.6301640 179.9891 59.52632  0.1863682 49.15433
## 7      7 240.8166 0.6159335 187.0179 53.80846  0.1745623 44.07619
## 8      8 241.0274 0.6023218 190.6716 66.95977  0.2334157 49.90191
## 9      9 263.0857 0.5341761 208.2962 67.24961  0.1935754 56.58095
## 10    10 270.4671 0.5248692 214.2343 63.71456  0.2522389 51.37369
## 11    11 255.9617 0.5600713 201.0775 61.59172  0.2075029 51.87136
## 12    12 251.9888 0.5766704 197.6533 64.54606  0.2072474 54.10643
## 13    13 240.4799 0.6060154 189.0064 70.05284  0.2048077 53.49371
## 14    14 244.5541 0.6005229 190.3550 67.24362  0.1987283 51.81588
## 15    15 250.9416 0.5750821 195.9889 64.93200  0.2037569 51.14981

step_reg$bestTune

##   nvmax
## 6     6

summary(step_reg$finalModel)

## Subset selection object
## 15 Variables  (and intercept)
##        Forced in Forced out
## M          FALSE      FALSE
## Ed         FALSE      FALSE
## Po1        FALSE      FALSE
## Po2        FALSE      FALSE
## LF         FALSE      FALSE
## M.F        FALSE      FALSE
## Pop        FALSE      FALSE
```

```
## NW          FALSE      FALSE
## U1          FALSE      FALSE
## U2          FALSE      FALSE
## Wealth      FALSE      FALSE
## Ineq        FALSE      FALSE
## Prob        FALSE      FALSE
## Time        FALSE      FALSE
## So          FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: 'sequential replacement'
##          M   Ed  Po1 Po2 LF  M.F Pop NW  U1  U2  Wealth Ineq Prob Time So
## 1  ( 1 ) " " " " "*" " " " " " " " " " " " " " " " "    " "  " "  " "  " "
## 2  ( 1 ) " " " " "*" " " " " " " " " " " " " " " " "    "*"  " "  " "  " "
## 3  ( 1 ) " " "*" "*" " " " " " " " " " " " " " " " "    "*"  " "  " "  " "
## 4  ( 1 ) "*" "*" "*" "*" " " " " " " " " " " " " " "    " "  " "  " "  " "
## 5  ( 1 ) "*" "*" "*" " " " " " " " " " " " " " " " "    "*"  "*"  " "  " "
## 6  ( 1 ) "*" "*" "*" " " " " " " " " " " " " "*" " "    "*"  "*"  " "  " "
```

I then built a regression model using these 6 variables.

```
model1 <- lm(Crime~ M+Ed+Po1+U2+Ineq+Prob, data = scaled_data)
summary(model1)

##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = scaled_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      29.27  30.918  < 2e-16 ***
## M             131.98      41.85   3.154  0.00305 **
## Ed            219.79      50.07   4.390 8.07e-05 ***
## Po1           341.84      40.87   8.363 2.56e-10 ***
## U2             75.47      34.55   2.185  0.03483 *
## Ineq          269.91      55.60   4.855 1.88e-05 ***
## Prob          -86.44      34.74  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11

AIC(model1)

## [1] 640.1661
```

```r
BIC(model1)

## [1] 654.9673
```

The model clearly showed thay all 6 variables were significant indicating that the stepwise process worked well. The R-sq value of the model was 76.5% which was good (not too high, not too low).

Next, I built the model with 80% training and 20% testing to confirm the results. The R-sq went down a little but overall the model performed the same as previous one except that it has lower AIC and BIC values.

```r
#splitting data to training and validation
set.seed(101)
sample <- sample.int(n = nrow(scaled_data), size =
floor(.80*nrow(scaled_data)), replace = F)
train_data <- scaled_data[sample,]
test_data  <- scaled_data[-sample,]
nrow(train_data)

## [1] 37

nrow(test_data)

## [1] 10

#building model 2 on training data
model2 <- lm(Crime~ M+Ed+Po1+U2+Ineq+Prob , data=train_data)
summary(model2)

##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -432.18 -124.12  -21.34   96.59  573.68
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   887.60      32.16  27.595  < 2e-16 ***
## M             138.71      44.44   3.121 0.003965 **
## Ed            197.28      53.04   3.719 0.000821 ***
## Po1           327.12      44.64   7.328 3.67e-08 ***
## U2             79.85      39.86   2.003 0.054232 .
## Ineq          212.30      58.71   3.616 0.001084 **
## Prob          -78.49      35.40  -2.217 0.034341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 192.5 on 30 degrees of freedom
```

```
## Multiple R-squared:  0.7586, Adjusted R-squared:  0.7103
## F-statistic: 15.71 on 6 and 30 DF,  p-value: 4.484e-08
```

```r
AIC(model2)
```

```
## [1] 502.5064
```

```r
BIC(model2)
```

```
## [1] 515.3937
```

```r
#checking model performance on testing data
eval <- predict(model2, test_data)
pred <- data.frame(cbind(actuals=test_data$Crime, predicteds=eval))
cor(pred)
```

```
##              actuals predicteds
## actuals    1.0000000  0.8092192
## predicteds 0.8092192  1.0000000
```

```r
head(pred)
```

```
##    actuals predicteds
## 2     1635  1343.8833
## 5     1234  1230.4937
## 15     798   780.7135
## 19     750  1231.8635
## 20    1225  1208.1651
## 23    1216   880.5614
```

Lastly, I used the stepAIC function from MASS package to perform the both ways stepwise regression.

```r
#building model 3
model <- lm(Crime~. , data=scaled_data)
model3 <- stepAIC(model, direction="both")
```

```
## Start:  AIC=514.65
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob + Time + So
##
##          Df Sum of Sq     RSS    AIC
## - So      1        29 1354974 512.65
## - LF      1      8917 1363862 512.96
## - Time    1     10304 1365250 513.00
## - Pop     1     14122 1369068 513.14
## - NW      1     18395 1373341 513.28
## - M.F     1     31967 1386913 513.74
## - Wealth  1     37613 1392558 513.94
## - Po2     1     37919 1392865 513.95
## <none>                1354946 514.65
## - U1      1     83722 1438668 515.47
```

```
## - Po1     1    144306 1499252 517.41
## - U2      1    181536 1536482 518.56
## - M       1    193770 1548716 518.93
## - Prob    1    199538 1554484 519.11
## - Ed      1    402117 1757063 524.86
## - Ineq    1    423031 1777977 525.42
##
## Step:  AIC=512.65
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - Time    1     10341 1365315 511.01
## - LF      1     10878 1365852 511.03
## - Pop     1     14127 1369101 511.14
## - NW      1     21626 1376600 511.39
## - M.F     1     32449 1387423 511.76
## - Po2     1     37954 1392929 511.95
## - Wealth  1     39223 1394197 511.99
## <none>                1354974 512.65
## - U1      1     96420 1451395 513.88
## + So      1        29 1354946 514.65
## - Po1     1    144302 1499277 515.41
## - U2      1    189859 1544834 516.81
## - M       1    195084 1550059 516.97
## - Prob    1    204463 1559437 517.26
## - Ed      1    403140 1758114 522.89
## - Ineq    1    488834 1843808 525.13
##
## Step:  AIC=511.01
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - LF      1     10533 1375848 509.37
## - NW      1     15482 1380797 509.54
## - Pop     1     21846 1387161 509.75
## - Po2     1     28932 1394247 509.99
## - Wealth  1     36070 1401385 510.23
## - M.F     1     41784 1407099 510.42
## <none>                1365315 511.01
## - U1      1     91420 1456735 512.05
## + Time    1     10341 1354974 512.65
## + So      1        65 1365250 513.00
## - Po1     1    134137 1499452 513.41
## - U2      1    184143 1549458 514.95
## - M       1    186110 1551425 515.01
## - Prob    1    237493 1602808 516.54
## - Ed      1    409448 1774763 521.33
## - Ineq    1    502909 1868224 523.75
```

```
##
## Step:  AIC=509.37
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + U2 + Wealth +
##     Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - NW      1     11675 1387523 507.77
## - Po2     1     21418 1397266 508.09
## - Pop     1     27803 1403651 508.31
## - M.F     1     31252 1407100 508.42
## - Wealth  1     35035 1410883 508.55
## <none>                1375848 509.37
## - U1      1     80954 1456802 510.06
## + LF      1     10533 1365315 511.01
## + Time    1      9996 1365852 511.03
## + So      1      3046 1372802 511.26
## - Po1     1    123896 1499744 511.42
## - U2      1    190746 1566594 513.47
## - M       1    217716 1593564 514.27
## - Prob    1    226971 1602819 514.54
## - Ed      1    413254 1789103 519.71
## - Ineq    1    500944 1876792 521.96
##
## Step:  AIC=507.77
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Po2     1     16706 1404229 506.33
## - Pop     1     25793 1413315 506.63
## - M.F     1     26785 1414308 506.66
## - Wealth  1     31551 1419073 506.82
## <none>                1387523 507.77
## - U1      1     83881 1471404 508.52
## + NW      1     11675 1375848 509.37
## + So      1      7207 1380316 509.52
## + LF      1      6726 1380797 509.54
## + Time    1      4534 1382989 509.61
## - Po1     1    118348 1505871 509.61
## - U2      1    201453 1588976 512.14
## - Prob    1    216760 1604282 512.59
## - M       1    309214 1696737 515.22
## - Ed      1    402754 1790276 517.74
## - Ineq    1    589736 1977259 522.41
##
## Step:  AIC=506.33
## Crime ~ M + Ed + Po1 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
```

```
## - Pop     1     22345 1426575 505.07
## - Wealth  1     32142 1436371 505.39
## - M.F     1     36808 1441037 505.54
## <none>                1404229 506.33
## - U1      1     86373 1490602 507.13
## + Po2     1     16706 1387523 507.77
## + NW      1      6963 1397266 508.09
## + So      1      3807 1400422 508.20
## + LF      1      1986 1402243 508.26
## + Time    1       575 1403654 508.31
## - U2      1    205814 1610043 510.76
## - Prob    1    218607 1622836 511.13
## - M       1    307001 1711230 513.62
## - Ed      1    389502 1793731 515.83
## - Ineq    1    608627 2012856 521.25
## - Po1     1   1050202 2454432 530.57
##
## Step:  AIC=505.07
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - Wealth  1     26493 1453068 503.93
## <none>                1426575 505.07
## - M.F     1     84491 1511065 505.77
## - U1      1     99463 1526037 506.24
## + Pop     1     22345 1404229 506.33
## + Po2     1     13259 1413315 506.63
## + NW      1      5927 1420648 506.87
## + So      1      5724 1420851 506.88
## + LF      1      5176 1421398 506.90
## + Time    1      3913 1422661 506.94
## - Prob    1    198571 1625145 509.20
## - U2      1    208880 1635455 509.49
## - M       1    320926 1747501 512.61
## - Ed      1    386773 1813348 514.35
## - Ineq    1    594779 2021354 519.45
## - Po1     1   1127277 2553852 530.44
##
## Step:  AIC=503.93
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## <none>                1453068 503.93
## + Wealth  1     26493 1426575 505.07
## - M.F     1    103159 1556227 505.16
## + Pop     1     16697 1436371 505.39
## + Po2     1     14148 1438919 505.47
## + So      1      9329 1443739 505.63
## + LF      1      4374 1448694 505.79
## + NW      1      3799 1449269 505.81
```

```
## + Time    1      2293 1450775 505.86
## - U1      1    127044 1580112 505.87
## - Prob    1    247978 1701046 509.34
## - U2      1    255443 1708511 509.55
## - M       1    296790 1749858 510.67
## - Ed      1    445788 1898855 514.51
## - Ineq    1    738244 2191312 521.24
## - Po1     1   1672038 3125105 537.93
```

```
model3$anova # display results
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 + 
##     Wealth + Ineq + Prob + Time + So
## 
## Final Model:
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob
## 
## 
##       Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1                                31    1354946 514.6488
## 2     - So  1    28.57405        32    1354974 512.6498
## 3   - Time  1 10340.66984        33    1365315 511.0072
## 4     - LF  1 10533.15902        34    1375848 509.3684
## 5     - NW  1 11674.63991        35    1387523 507.7655
## 6    - Po2  1 16706.34095        36    1404229 506.3280
## 7    - Pop  1 22345.36638        37    1426575 505.0700
## 8 - Wealth  1 26493.24677        38    1453068 503.9349
```

```
summary(model3)
```

```
## 
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob, 
##     data = scaled_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -444.70 -111.07    3.03  122.15  483.30 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   905.09      28.52  31.731  < 2e-16 ***
## M             117.28      42.10   2.786  0.00828 ** 
## Ed            201.50      59.02   3.414  0.00153 ** 
## Po1           305.07      46.14   6.613 8.26e-08 ***
## M.F            65.83      40.08   1.642  0.10874    
## U1           -109.73      60.20  -1.823  0.07622 .
```

```
## U2            158.22      61.22   2.585  0.01371 *
## Ineq          244.70      55.69   4.394 8.63e-05 ***
## Prob          -86.31      33.89  -2.547  0.01505 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.5 on 38 degrees of freedom
## Multiple R-squared:  0.7888, Adjusted R-squared:  0.7444
## F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

```
AIC(model3)
```

```
## [1] 639.3151
```

```
BIC(model3)
```

```
## [1] 657.8166
```

The AIC and BIC values of this model were very close to the first model we built based on 6 variables. This stepwise process selected 8 variables out of 15 (M.F and U1 were the two new addeed). TheR-sq is higher than the previous model but in the same range. AIC and BIC make this model comparable to the first one indicating the two new variables in this model didn't improve things much and we could use the first model with 6 variables.

LASSO REGRESSION

Kicking off the Lasso regression with glmnet function using Alpha = 1.

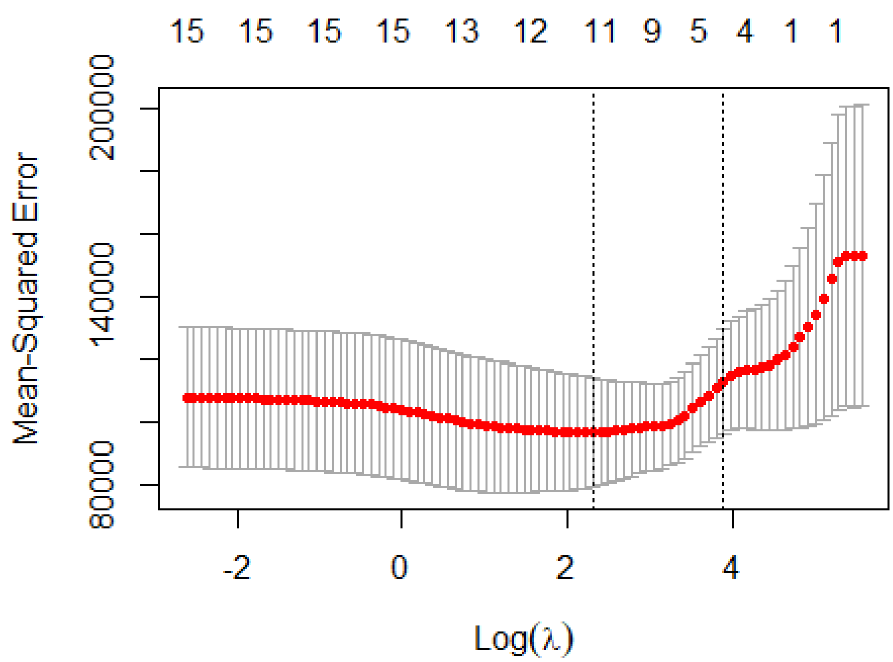
```
set.seed(101)
lasso_reg = cv.glmnet(x=as.matrix(scaled_data[,-16]),
                      y=as.matrix(scaled_data$Crime),
                      alpha=1,
                      nfolds = 5,
                      type.measure="mse",
                      family="gaussian")
lasso_reg
```

```
##
## Call:  cv.glmnet(x = as.matrix(scaled_data[, -16]), y =
as.matrix(scaled_data$Crime),      type.measure = "mse", nfolds = 5, alpha =
1, family = "gaussian")
##
## Measure: Mean-Squared Error
##
##     Lambda Measure    SE Nonzero
## min  10.14   96373 17488      11
## 1se  49.30  112838 16743       5
```

```
coef(lasso_reg, s=lasso_reg$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) 889.384205
## M            86.916102
## Ed          131.801907
## Po1         307.727030
## Po2           .
## LF            0.095438
## M.F          54.059130
## Pop           .
## NW            5.188570
## U1          -29.893902
## U2           64.403807
## Wealth        .
## Ineq        185.202923
## Prob        -83.088331
## Time          .
## So           46.121397
```

```
plot(lasso_reg)
```



Lasso method suggested 11 variables with alpha = 1. I built the model using these 11 variables (first using all data and then using training & testing)

```
model4 <- lm(Crime~ M+Ed+Po1+LF+M.F+NW+U1+U2+Ineq+Prob+So, data =
scaled_data)
summary(model4)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + LF + M.F + NW + U1 + U2 +
##     Ineq + Prob + So, data = scaled_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -443.2 -101.4    4.1  120.5  486.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   892.63      55.99  15.943  < 2e-16 ***
## M             106.61      49.29   2.163  0.03747 *
## Ed            209.15      65.00   3.218  0.00278 **
## Po1           295.60      54.50   5.424 4.44e-06 ***
## LF            -10.69      54.11  -0.198  0.84447
## M.F            74.96      51.13   1.466  0.15159
## NW             13.01      59.46   0.219  0.82814
## U1           -109.08      71.71  -1.521  0.13725
## U2            151.47      65.99   2.295  0.02783 *
## Ineq          233.00      67.67   3.443  0.00151 **
## Prob          -96.00      39.58  -2.425  0.02059 *
## So             36.57     139.62   0.262  0.79489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202.9 on 35 degrees of freedom
## Multiple R-squared:  0.7906, Adjusted R-squared:  0.7248
## F-statistic: 12.01 on 11 and 35 DF,  p-value: 6.965e-09
```

```
AIC(model4)
```

```
## [1] 644.9212
```

```
BIC(model4)
```

```
## [1] 668.9731
```

```
#building model on training data
model5 <- lm(Crime~ M+Ed+Po1+LF+M.F+NW+U1+U2+Ineq+Prob+So , data=train_data)
summary(model5)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + LF + M.F + NW + U1 + U2 +
##     Ineq + Prob + So, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -298.27 -119.80    0.93  109.10  487.05
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  870.135     63.396  13.725 3.83e-13 ***
## M            119.904     50.889   2.356 0.026611 *
## Ed           215.728     66.820   3.229 0.003465 **
## Po1          264.986     59.518   4.452 0.000154 ***
## LF           -15.781     53.713  -0.294 0.771331
## M.F           64.298     63.833   1.007 0.323451
## NW             6.287     70.135   0.090 0.929289
## U1          -150.907     91.794  -1.644 0.112700
## U2           216.008     94.914   2.276 0.031687 *
## Ineq         177.786     76.732   2.317 0.028987 *
## Prob         -86.573     41.575  -2.082 0.047701 *
## So            64.122    157.568   0.407 0.687510
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 192.8 on 25 degrees of freedom
## Multiple R-squared:  0.7984, Adjusted R-squared:  0.7096
## F-statistic: 8.998 on 11 and 25 DF,  p-value: 3.188e-06
```

```
AIC(model5)
```

```
## [1] 505.849
```

```
BIC(model5)
```

```
## [1] 526.7909
```

```
#checking model performance on testing data
eval <- predict(model5, test_data)
pred <- data.frame(cbind(actuals=test_data$Crime, predicteds=eval))
cor(pred)
```

```
##              actuals predicteds
## actuals    1.0000000  0.7733889
## predicteds 0.7733889  1.0000000
```

```
head(pred)
```

```
##    actuals predicteds
## 2     1635  1383.4323
## 5     1234  1004.2471
## 15     798   958.0695
## 19     750  1260.1291
## 20    1225  1284.8101
## 23    1216   835.4384
```

Model with full data and training data (80%) had very similar R-sq but AIC value for the model trained on 80% of the data was lower indicating that model5 was better.

Elastic Net Regression

For elastic net, I used the same glmnet function but varied the alpha value b/w 1 (lasso) and 0 (ridge) to get the best variables combination.

```
set.seed(101)
list <- numeric()

#function to run the Loop on
best_alpha <- function(num, scaled_date){
  alpha = num
  elastic <- cv.glmnet(x=as.matrix(scaled_data[,-16]),
                      y=as.matrix(scaled_data$Crime),
                      alpha=alpha,
                      nfolds = 5,
                      type.measure="mse",
                      family="gaussian")
  list <<- cbind(list, c(alpha, min(elastic$cvm),elastic$lambda.min))
}

for (i in seq(0.01, 1, by=0.01)){
  best_alpha(i, scaled_data)
}

#minimum MSE in the Loop
list[2,which.min(list[2,])]

## [1] 54508.37

#which alpha value lowest MSE was at
list[1, which.min(list[2,])]

## [1] 0.49
```

The results of the loop from 0.01 to 1 in 0.01 intervals of alpha showed that the best alpha with minumum MSE was 0.49. I built the Elasti net with this alpha to get the variables list.

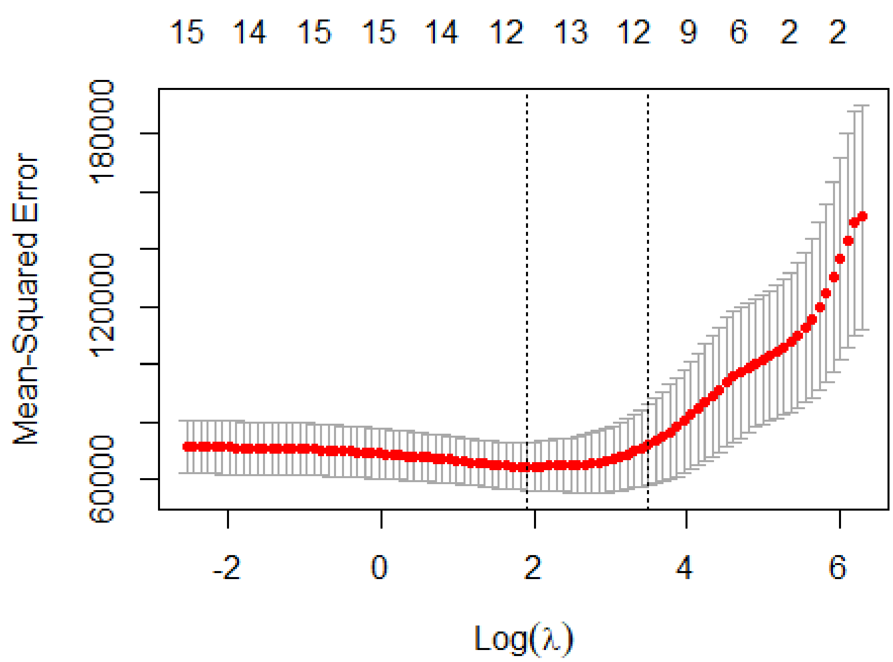
```
elastic_final <- cv.glmnet(x=as.matrix(scaled_data[,-16]),
                      y=as.matrix(scaled_data$Crime),
                      alpha=0.49,
                      nfolds = 5,
                      type.measure="mse",
                      family="gaussian")

coef(elastic_final, s=elastic_final$lambda.min)

## 16 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 891.477406
## M            99.544684
## Ed          165.197504
```

```
## Po1         284.795680
## Po2           1.154623
## LF             .
## M.F          58.125398
## Pop         -13.079211
## NW           19.011334
## U1          -69.398589
## U2          110.997688
## Wealth       50.338572
## Ineq        230.336499
## Prob        -90.138521
## Time           .
## So           39.972621
```

```
plot(elastic_final)
```



The elastic net revealed 13 variables. I built the regression model on these variables.

```
model6 <- lm(Crime~ M+Ed+Po1+Po2+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+So, data =
scaled_data)
summary(model6)

##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + So, data = scaled_data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -389.63  -94.25    7.83  109.20  491.62 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   893.38      52.51  17.012  < 2e-16 ***
## M             109.87      49.82   2.205  0.03451 *  
## Ed            202.41      64.00   3.163  0.00335 ** 
## Po1           501.63     287.30   1.746  0.09012 .  
## Po2          -215.08     288.65  -0.745  0.46148    
## M.F            43.45      48.99   0.887  0.38162    
## Pop           -36.21      46.10  -0.785  0.43784    
## NW             24.91      58.61   0.425  0.67360    
## U1            -86.62      66.24  -1.308  0.20002    
## U2            136.97      67.41   2.032  0.05027 .  
## Wealth         82.03      96.17   0.853  0.39983    
## Ineq          275.77      86.79   3.177  0.00322 ** 
## Prob          -95.16      41.52  -2.292  0.02843 *  
## So             34.40     127.12   0.271  0.78840    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 204 on 33 degrees of freedom
## Multiple R-squared:  0.8005, Adjusted R-squared:  0.7219 
## F-statistic: 10.19 on 13 and 33 DF,  p-value: 4.088e-08
```

```r
AIC(model6)
```

```
## [1] 646.6444
```

```r
BIC(model6)
```

```
## [1] 674.3966
```

```r
#building model on training data
model7 <- lm(Crime~ M+Ed+Po1+Po2+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+So,
data=train_data)
summary(model7)
```

```
## 
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + 
##     U2 + Wealth + Ineq + Prob + So, data = train_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -293.14 -136.58    3.07  115.19  485.78 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  867.579     64.922  13.363 2.51e-12 ***
## M            123.002     52.858   2.327  0.02913 *
## Ed           209.126     70.149   2.981  0.00668 **
## Po1          217.839    401.641   0.542  0.59278
## Po2           61.013    397.488   0.153  0.87935
## M.F           50.375     65.116   0.774  0.44703
## Pop          -10.201     54.550  -0.187  0.85329
## NW            -2.456     78.095  -0.031  0.97519
## U1          -141.630     90.596  -1.563  0.13163
## U2           216.081    100.800   2.144  0.04286 *
## Wealth         2.259    100.103   0.023  0.98219
## Ineq         185.581    107.348   1.729  0.09725 .
## Prob         -86.198     44.747  -1.926  0.06651 .
## So            76.137    154.037   0.494  0.62580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201 on 23 degrees of freedom
## Multiple R-squared:  0.7983, Adjusted R-squared:  0.6843
## F-statistic: 7.002 on 13 and 23 DF,  p-value: 2.962e-05
```

```
AIC(model7)
```

```
## [1] 509.8619
```

```
BIC(model7)
```

```
## [1] 534.0257
```

```
#checking model performance on testing data
eval <- predict(model7, test_data)
pred <- data.frame(cbind(actuals=test_data$Crime, predicteds=eval))
cor(pred)
```

```
##              actuals predicteds
## actuals    1.0000000  0.7684252
## predicteds 0.7684252  1.0000000
```

```
head(pred)
```

```
##    actuals predicteds
## 2     1635  1383.1095
## 5     1234  1017.0148
## 15     798   967.0064
## 19     750  1282.7850
## 20    1225  1292.6434
## 23    1216   853.6128
```

Unsurprisingly, models based on 13 variables have higher R-sq because there could be overfitting here due to small amount of data. I would go with models 2 or 3 instead due to simplicity because they use less variables(6 and 8) and offer similar R-sq (75% to 78%).