AnalyticsModeling_HW8

Fall 2024

```r
# load data
crime_df <- read.table("uscrime.txt",  stringsAsFactors = FALSE, header = TRUE)
tail(crime_df, 5)
```

```
##       M So   Ed  Po1 Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 43 16.2  1  9.9  7.5 7.0 0.522  99.6  40 20.8 0.073 2.7   4960 22.4 0.054902
## 44 13.6  0 12.1  9.5 9.6 0.574 101.2  29  3.6 0.111 3.7   6220 16.2 0.028100
## 45 13.9  1  8.8  4.6 4.1 0.480  96.8  19  4.9 0.135 5.3   4570 24.9 0.056202
## 46 12.6  0 10.4 10.6 9.7 0.599  98.9  40  2.4 0.078 2.5   5930 17.1 0.046598
## 47 13.0  0 12.1  9.0 9.1 0.623 104.9   3  2.2 0.113 4.0   5880 16.0 0.052802
##       Time Crime
## 43 31.9989   823
## 44 30.0001  1030
## 45 32.5996   455
## 46 16.6999   508
## 47 16.0997   849
```

Question 11.1

Using the crime data set, build a regression model using Stepwise Regression

I started with a model that used all predictors and then used stepwise regression to reduce the number of variables. At each step, the stepwise regression removed the predictor with the lowest AIC until finally the dataset was reduced from 15 predictors to 8.

The initial model with all predictors resulted in an adjusted R-squared value of 0.7078, indicating that 70.78% of the model's variability is explained by the predictors.

The stepwise regression resulted in a model containing 8 predictors:

- `M`
- `Ed`
- `Po1`
- `M.F`
- `U1`
- `U2`
- `Ineq`
- `Prob`

And resulted in an adjusted R-squared value of 0.7444, indicating that 74.44% of the model's variability is explained by the predictors.

Stepwise Regression resulted in a model that improved in two ways:

- improved evaluation metrics
- increased model simplicity

```r
# set seed for reproducibility
set.seed(1)

# start with a model that has all predictors
initial_model <- lm(Crime~.
                    , data = crime_df
                    )
summary(initial_model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = crime_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

```r
# perform both direction stepwise regression
stepwise_model <- step(initial_model
                       , scope = list(lower = formula(lm(Crime~1, data = crime_df))
                                      , upper = formula(lm(Crime~., data = crime_df))
                                      )
                       , direction = "both"
                       )
```

```
## Start:  AIC=514.65
## Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + Time
##
##          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
```

```r
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = crime_df)
##
## 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) -6426.10    1194.61  -5.379 4.04e-06 ***
## M              93.32      33.50   2.786  0.00828 **
## Ed            180.12      52.75   3.414  0.00153 **
## Po1           102.65      15.52   6.613 8.26e-08 ***
## M.F            22.34      13.60   1.642  0.10874
## U1          -6086.63    3339.27  -1.823  0.07622 .
## U2            187.35      72.48   2.585  0.01371 *
## Ineq           61.33      13.96   4.394 8.63e-05 ***
## Prob        -3796.03    1490.65  -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
```

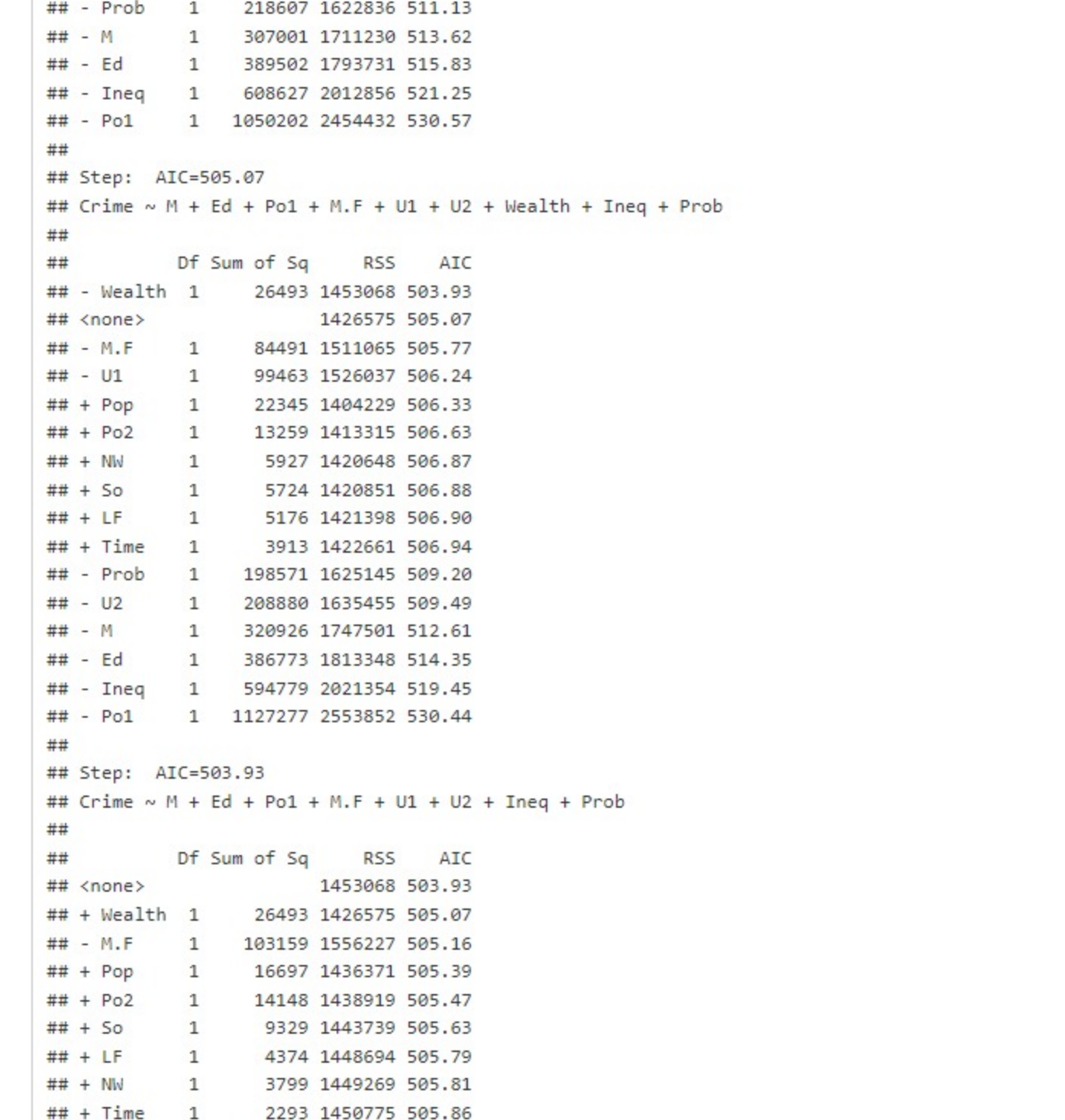
Using the crime data set, build a regression model using Lasso Regression

I created a lasso model using `cv.glmnet`, which automatically scaled the data for me. The tau / lambda threshold that worked best for the model was 8.83952725, which best minimized the MSE, and resulted in a regression with 11 non-zero variables. In other words, the variable selection process removed 4 predictors.

```r
# set seed for reproducibility
set.seed(1)

# do k-fold cross validation for lasso model
lasso_model <- cv.glmnet(x = as.matrix(crime_df[, -16])
                         , y = as.matrix(crime_df[,16])
                         , alpha = 1                        # lasso regression alpha = 1
                         , nfolds = 8                       # number of folds
                         , nlambda = 20                     # tau thresholds randomly generated
                         , type.measure = "mse"             # squared error for gaussian models
                         , family = "gaussian"
                         , standardize = TRUE               # use automatically scaled data
                         )

# plot MSE of lasso model
plot(lasso_model)
```




```r
# use the tau / lambda that corresponds to the lowest MSE
lasso_model$lambda.min
```

```
## [1] 8.839527
```

```r
# get a list of tau/lambdas, cross-validation error, and number of non-zero coefficients for each lambda.
cbind(lasso_model$lambda, lasso_model$cvm, lasso_model$nzero)
```

```
##             [,1]      [,2] [,3]
## s0  263.09539664 151480.86    0
## s1  162.02682936 121743.71    1
## s2   99.78393301 103954.55    1
## s3   61.45175663  93168.04    4
## s4   37.84495439  76653.63    5
## s5   23.30674746  71555.18    9
## s6   14.35341873  68765.92   10
## s7    8.83952725  67598.55   11
## s8    5.44380704  67649.74   12
## s9    3.35255883  68290.67   12
## s10   2.06466736  70005.68   13
## s11   1.27152170  71798.41   14
## s12   0.78306436  73344.38   15
## s13   0.48224879  73634.73   15
## s14   0.29699205  73757.71   15
## s15   0.18290202  73917.56   15
## s16   0.11263988  74034.73   14
## s17   0.06936907  74115.66   15
## s18   0.04272082  74164.91   15
## s19   0.02630954  74223.75   15
```

```r
# get the coefficients of the model with the best tau / lambda
coef(lasso_model, s = lasso_model$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -5.072255e+03
## M            7.184295e+01
## So           4.466407e+01
## Ed           1.253875e+02
## Po1          1.023402e+02
## Po2          .
## LF           .
## M.F          1.888147e+01
## Pop          .
## NW           6.315089e-01
## U1          -2.143645e+03
## U2           8.835503e+01
## Wealth       7.715072e-03
## Ineq         4.882548e+01
## Prob        -3.688177e+03
## Time         .
```

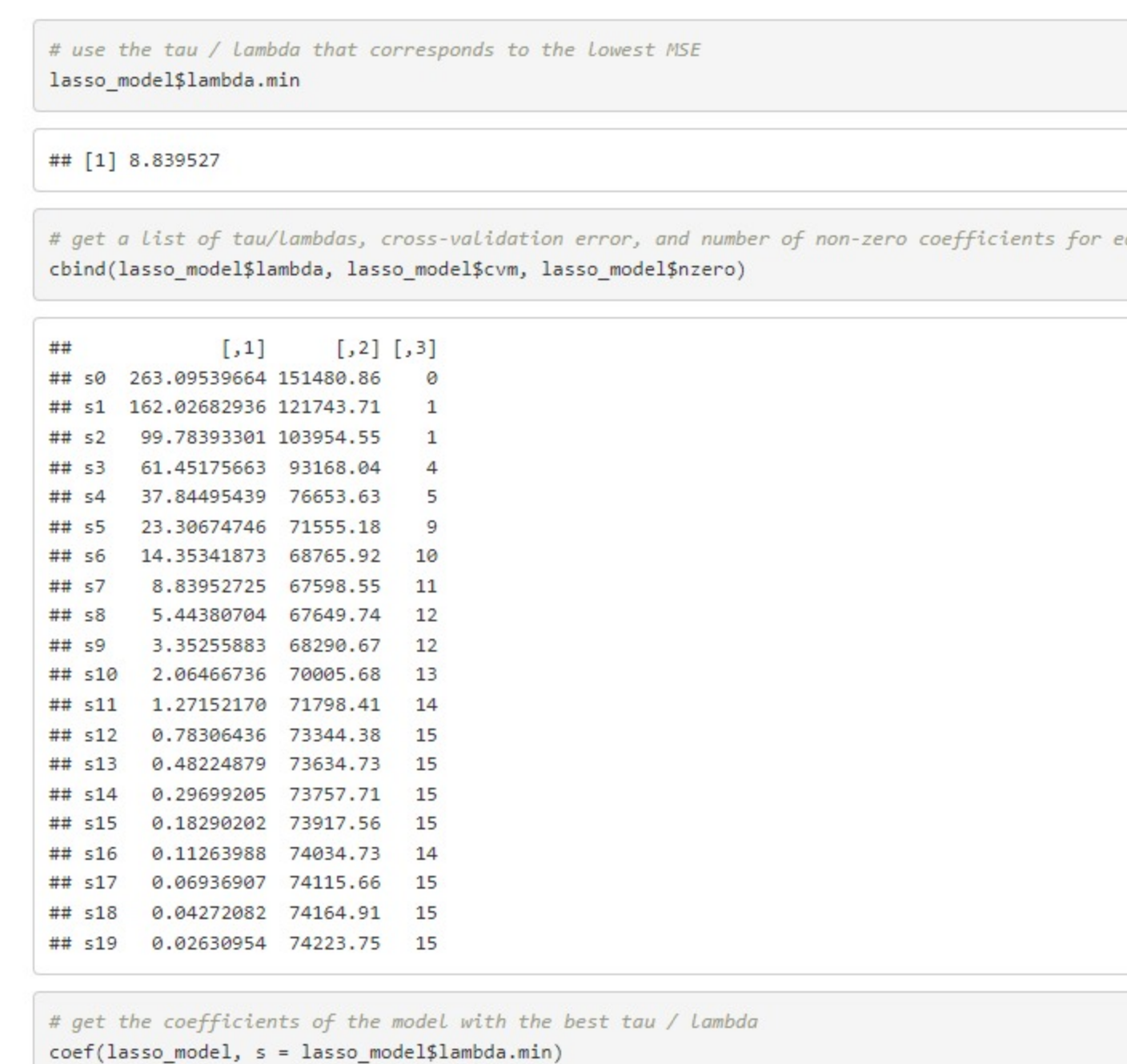
Using the crime data set, build a regression model using Elastic Net Regression

Comparatively, I created an elastic net model using `cv.glmnet`, which automatically scaled the data for me. The tau / lambda threshold that worked best for the model was 13.4102353, which best minimized the MSE, and resulted in a regression with 14 non-zero variables. In other words, the variable selection process removed 1 predictor.

```r
# set seed for reproducibility
set.seed(1)

# do k-fold cross validation for Elastic Net Regression model
enet_model <- cv.glmnet(x = as.matrix(crime_df[, -16])
                        , y = as.matrix(crime_df[,16])
                        , alpha = .25                      # elastic regression alpha between 0,1
                        , nfolds = 8                       # number of folds
                        , nlambda = 20                     # tau thresholds randomly generated
                        , type.measure = "mse"             # squared error for gaussian models
                        , family = "gaussian"
                        , standardize = TRUE               # use automatically scaled data
                        )

# plot MSE of elastic net model
plot(enet_model)
```




```r
# use the tau / lambda that corresponds to the lowest MSE
enet_model$lambda.min
```

```
## [1] 13.41024
```

```r
# get a list of tau/lambdas, cross-validation error, and number of non-zero coefficients for each lambda.
cbind(enet_model$lambda, enet_model$cvm, enet_model$nzero)
```

```
##            [,1]      [,2] [,3]
## s0  1052.3815866 151637.08    0
## s1   648.1073174 131704.50    2
## s2   399.1357320 114386.29    3
## s3   245.8070265 103499.42    4
## s4   151.3798176  90277.38    7
## s5    93.2269898  80318.86   11
## s6    57.4136749  74902.13   12
## s7    35.3581090  71932.13   13
## s8    21.7752282  70199.96   13
## s9    13.4102353  68930.95   14
## s10    8.2586694  69312.54   14
## s11    5.0860868  70458.52   14
## s12    3.1322575  71322.94   14
## s13    1.9289951  72575.09   15
## s14    1.1879682  73146.45   15
## s15    0.7316081  73414.66   15
## s16    0.4505595  73585.09   15
## s17    0.2774763  73778.99   15
## s18    0.1708833  73938.90   15
## s19    0.1052382  74048.81   15
```

```r
# get the coefficients of the model with the best tau / lambda
coef(enet_model, s = enet_model$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -5.641666e+03
## M            7.479604e+01
## So           5.460175e+01
## Ed           1.301457e+02
## Po1          7.004305e+01
## Po2          2.589845e+01
## LF           1.364334e+02
## M.F          2.160439e+01
## Pop         -1.499301e-01
## NW           2.015364e+00
## U1          -3.450636e+03
## U2           1.203124e+02
## Wealth       4.213319e-02
## Ineq         5.071427e+01
## Prob        -3.952460e+03
## Time         .
```

It is interesting that all three methods removed the variable `Time`.

Out of the two global optimization variable selection methods, the Lasso model resulted in a model with a lower error at `67598.55` while the Elastic Net model had a slightly higher error of `68930.95`.