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.

Notes on R:
• For the elastic net model, what we called λ in the videos, glmnet calls "alpha"; you can get a range of results by varying alpha from 1 (lasso) to 0 (ridge regression) [and, of course, other values of alpha in between].
• In a function call like glmnet(x,y,family=”mgaussian”,alpha=1) the predictors x need to be in R's matrix format, rather than data frame format. You can convert a data frame to a matrix using as.matrix – for example, x <- as.matrix(data[,1:n-1]) • Rather than specifying a value of T, glmnet returns models for a variety of values of T.

1. Stepwise Regression

   Using stepwise regression, I came up with the regression model: {Crime = -6426.10 + 93.32M + 180.12Ed + 102.65Po1 + 22.34MF – 6086.63 + 187.35U2 + 61.33Ineq – 3796.03Prob}. I got this equation by first running a regression model on the crime data set and then looking at the summary of it.

```
set.seed(123)
data <- read.table('uscrime.txt', header = TRUE)
# Build the initial full model
full_model <- lm(Crime ~ ., data = data)
summary(full_model)

##
## Call:
## lm(formula = Crime ~ ., data = data)
```

```
##
## 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
```

Looking at the initial model, I could see that there were many variables that were not significant to predicting crime. This meant that they likely should not be used in the regression equation. To sort through which variables should not be included, I ran a stepwise regression on the regression equation shown above.

```
# Apply stepwise regression
stepwise_model <- step(full_model, 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

# View the summary of the final model
summary(stepwise_model)

##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = 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) -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
```

The stepwise regression ran a backwards elimination, meaning it started with all the variables in the original equation and eliminated them one at a time if having it in the equation resulted in a higher AIC value than without it. Above are all the variables that the stepwise regression deemed to be good enough to be in the regression equation. It has an adjusted R squared value of around 74%, which is better than the original equation's about 71% variation explained. What I found interesting about the variables it selected was that it included two that were not significant at the p-value < 0.05 level. So, to further explore this, I ran a normal regression using the variables that the stepwise recommended to use minus MF and U1 (the two non-significant ones) to see what the difference in variability explained would be.

```
#stepwise model without non-significant values
final <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = data)
summary(final)

##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = 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) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -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
```

It turned out that the adjusted R squared value for the equation with only significant variables was lower than the one with the non-significant ones, though not by much. This makes sense though because the non-significant variables can add more randomness into the model which the model will fit to, raising the variance. I would probably pick the equation without the non-significant values because the R squared value is about the same and all the variables are meaningful at a high degree. This equation would be: {Crime = -5040.50 + 105.02M + 196.47Ed + 115.02Po1 + 89.37U2 + 67.65Ineq – 3801.84}.

2. LASSO

Using LASSO, the regression equation I came up with was: {Crime = -5916.29 + 83.12M + 31.64So + 156.54Ed + 99.71Po1 + 17.76MF – 0.5Pop + 1.39NW – 3942.58U1 +137.64U2 +0.06Wealth + 62.71Ineq – 3923.13Prob}. I got this equation by first creating my predictors and my response. As stated in the directions, I made my predictors a matrix to have glmnet run.

```
data <- read.table('uscrime.txt', header = TRUE)

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
# 1. Prepare the predictors (X) and response (y)
x <- as.matrix(data[, -ncol(data)])  # Exclude the response variable (Crime)
y <- data$Crime
```

I then saw that it said to remember to scale the data, so I then did that.

```
# 2. Scale the predictors (X) and store scaling parameters
x_scaled <- scale(x)
x_means <- attr(x_scaled, "scaled:center")  # Mean of each column
x_sds <- attr(x_scaled, "scaled:scale")     # Std dev of each column
```

I saved the means of each column and the standard deviations of each column so that I could un-scale the data later to get the actual regression coefficient numbers. At this time, I was ready to run a LASSO regression on the data.

```
# 3. Fit the Lasso model using scaled data
lasso_model <- glmnet(x_scaled, y, family = "gaussian", alpha = 1)
# 4. Cross-validate to find the optimal lambda (penalty)
cv_lasso <- cv.glmnet(x_scaled, y, alpha = 1)
```

I used alpha = 1 to tell R that I wanted to run a LASSO regression. I then wanted to make sure that I was using the best lambda value, so I cross validated my lasso model to find just that.

```
# 5. Extract the best lambda value
best_lambda <- cv_lasso$lambda.min
```

At this time, I was ready to get the coefficients of the scaled data, un-scale them, and find the regression equation running this LASSO regression came out with.

```
# 6. Get the coefficients from the model at the optimal lambda
scaled_coefs <- coef(cv_lasso, s = best_lambda)

# 7. Unscale the coefficients to get them in the original units
unscaled_coefs <- scaled_coefs[-1] / x_sds  # Exclude intercept, divide by SD
intercept <- as.numeric(scaled_coefs[1]) - sum(unscaled_coefs * x_means)

# 8. Display the final unscaled coefficients and intercept
print(intercept)

## [1] -5916.287

for (i in 1:length(unscaled_coefs)) {
  cat(" +", round(unscaled_coefs[i], 2), "*", colnames(data)[i], "\n")
}

##  + 83.12 * M
##  + 31.64 * So
##  + 156.54 * Ed
##  + 99.71 * Po1
##  + 0 * Po2
##  + 0 * LF
##  + 17.76 * M.F
##  + -0.5 * Pop
##  + 1.39 * NW
##  + -3942.58 * U1
##  + 137.64 * U2
##  + 0.06 * Wealth
##  + 62.71 * Ineq
##  + -3923.13 * Prob
##  + 0 * Time
```

Above shows how that was done. Given the penalty term alpha = 1, the LASSO regression deemed that the variables P02, LF, and Time was not important in predicting Crime. I would also say that Pop and Wealth are also not important as they are both very close to 0, but I still put them in my equation because they were not exactly 0, so they still describe something even if it is very small. If they were not added, I am sure the quality of the model would be a little better.

3. Elastic Net

Using Elastic Net, the regression equation I came up with was: {Crime = -5988.94 + 82.4M + 35.29So + 156.21Ed + 97.14Po1 + 18.83MF – 0.5Pop + 1.85NW – 4202.44U1 + 142.4U2 +0.06Wealth + 61.75Ineq – 4008.13Prob}. The way I went about finding this equation is pretty much exactly how I got the LASSO regression equation. The only difference with the Elastic Net was when using the glmnet function, I used alpha = 0.05. Using alpha = 0 means you run a Ridge regression and using alpha = 1 means you run a LASSO regression. Elastic Net being a mix of LASSO and Ridge, I used 0.5 as half and half of each. I'm not sure if using different mixes of each would significantly change the final output, but 0.5 made the most sense to me intuitively. The explanation of everything else is exactly as it was in the LASSO regression.

```
set.seed(123)
data <- read.table('uscrime.txt', header = TRUE)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8

# 2. Prepare predictors (X) and response (y)
x <- as.matrix(data[, -ncol(data)])  # Exclude the response variable (Crime)
y <- data$Crime

# 3. Scale the predictors and store scaling parameters
x_scaled <- scale(x)
x_means <- attr(x_scaled, "scaled:center")  # Store column means
x_sds <- attr(x_scaled, "scaled:scale")     # Store column standard deviation
s

# 4. Fit the Elastic Net model with alpha = 0.5 (50% Lasso, 50% Ridge)
elastic_net_model <- glmnet(x_scaled, y, family = "gaussian", alpha = 0.5)

# 5. Cross-validate to find the optimal lambda (penalty)
cv_elastic_net <- cv.glmnet(x_scaled, y, alpha = 0.5)

# 6. Extract the best lambda value
best_lambda_elastic <- cv_elastic_net$lambda.min

# 7. Get the coefficients from the model at the optimal lambda
scaled_coefs <- coef(cv_elastic_net, s = best_lambda_elastic)

# 8. Unscale the coefficients
unscaled_coefs <- scaled_coefs[-1] / x_sds  # Exclude intercept, divide by SD
intercept <- as.numeric(scaled_coefs[1]) - sum(unscaled_coefs * x_means)

# 9. Display the final unscaled coefficients and intercept
print(intercept)

## [1] -5988.94

for (i in 1:length(unscaled_coefs)) {
  cat(" +", round(unscaled_coefs[i], 2), "*", colnames(data)[i], "\n")
}
```

```
##  + 82.4 * M
##  + 35.29 * So
##  + 156.21 * Ed
##  + 97.14 * Po1
##  + 0 * Po2
##  + 0 * LF
##  + 18.83 * M.F
##  + -0.5 * Pop
##  + 1.85 * NW
##  + -4202.44 * U1
##  + 142.4 * U2
##  + 0.06 * Wealth
##  + 61.75 * Ineq
##  + -4008.13 * Prob
##  + 0 * Time
```

I found it interesting that both LASSO and Elastic Net deemed Po2, LF, and Time to be non-important, and Pop and Wealth to be of very minimal importance. In comparing the equations, all the coefficients of the variables are very similar which I also found interesting. I tested out a couple different alpha values closer to ridge regression (alpha = 0) like 0.3 and 0.2 to see if it would be much different from what both alpha = 0.5 and 1 came out with, and after alpha < 0.3 it was obvious that it was more Ridge regression than LASSO because less variables were being eliminated. Using alpha = 0 gave the most difference which makes sense because it is an entirely different type of regression at that point which does not eliminate any variables.

R CODE AND OUTPUT

STEPWISE:

```
set.seed(123)
data <- read.table('uscrime.txt', header = TRUE)
# Build the initial full model
full_model <- lm(Crime ~ ., data = data)
summary(full_model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = data)
##
## 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
```

```
# Apply stepwise regression
stepwise_model <- step(full_model, 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
```

```
# View the summary of the final model
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = 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) -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
```

```
#stepwise model without non significant values
final <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = data)
summary(final)

##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = 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) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -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
```

LASSO:

```
set.seed(123)
data <- read.table('uscrime.txt', header = TRUE)

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8

# 1. Prepare the predictors (X) and response (y)
x <- as.matrix(data[, -ncol(data)])  # Exclude the response variable (Crime)
y <- data$Crime

# 2. Scale the predictors (X) and store scaling parameters
x_scaled <- scale(x)
x_means <- attr(x_scaled, "scaled:center")  # Mean of each column
x_sds <- attr(x_scaled, "scaled:scale")     # Std dev of each column

# 3. Fit the Lasso model using scaled data
```

```
lasso_model <- glmnet(x_scaled, y, family = "gaussian", alpha = 1)

# 4. Cross-validate to find the optimal lambda (penalty)
cv_lasso <- cv.glmnet(x_scaled, y, alpha = 1)

# 5. Extract the best lambda value
best_lambda <- cv_lasso$lambda.min

# 6. Get the coefficients from the model at the optimal lambda
scaled_coefs <- coef(cv_lasso, s = best_lambda)

# 7. Unscale the coefficients to get them in the original units
unscaled_coefs <- scaled_coefs[-1] / x_sds  # Exclude intercept, divide by SD
intercept <- as.numeric(scaled_coefs[1]) - sum(unscaled_coefs * x_means)

# 8. Display the final unscaled coefficients and intercept
print(intercept)

## [1] -5916.287

for (i in 1:length(unscaled_coefs)) {
  cat(" +", round(unscaled_coefs[i], 2), "*", colnames(data)[i], "\n")
}

##  + 83.12 * M
##  + 31.64 * So
##  + 156.54 * Ed
##  + 99.71 * Po1
##  + 0 * Po2
##  + 0 * LF
##  + 17.76 * M.F
##  + -0.5 * Pop
##  + 1.39 * NW
##  + -3942.58 * U1
##  + 137.64 * U2
##  + 0.06 * Wealth
##  + 62.71 * Ineq
##  + -3923.13 * Prob
##  + 0 * Time
```

ELASTIC NET:

```
set.seed(123)
data <- read.table('uscrime.txt', header = TRUE)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
# 2. Prepare predictors (X) and response (y)
x <- as.matrix(data[, -ncol(data)])  # Exclude the response variable (Crime)
y <- data$Crime

# 3. Scale the predictors and store scaling parameters
x_scaled <- scale(x)
x_means <- attr(x_scaled, "scaled:center")  # Store column means
x_sds <- attr(x_scaled, "scaled:scale")     # Store column standard deviation
s

# 4. Fit the Elastic Net model with alpha = 0.5 (50% Lasso, 50% Ridge)
elastic_net_model <- glmnet(x_scaled, y, family = "gaussian", alpha = 0.5)

# 5. Cross-validate to find the optimal lambda (penalty)
cv_elastic_net <- cv.glmnet(x_scaled, y, alpha = 0.5)

# 6. Extract the best lambda value
best_lambda_elastic <- cv_elastic_net$lambda.min

# 7. Get the coefficients from the model at the optimal lambda
scaled_coefs <- coef(cv_elastic_net, s = best_lambda_elastic)

# 8. Unscale the coefficients
unscaled_coefs <- scaled_coefs[-1] / x_sds  # Exclude intercept, divide by SD
intercept <- as.numeric(scaled_coefs[1]) - sum(unscaled_coefs * x_means)

# 9. Display the final unscaled coefficients and intercept
print(intercept)
```

```
## [1] -5988.94
```

```
for (i in 1:length(unscaled_coefs)) {
  cat(" +", round(unscaled_coefs[i], 2), "*", colnames(data)[i], "\n")
}
```

```
##  + 82.4 * M
##  + 35.29 * So
##  + 156.21 * Ed
##  + 97.14 * Po1
##  + 0 * Po2
##  + 0 * LF
##  + 18.83 * M.F
##  + -0.5 * Pop
##  + 1.85 * NW
##  + -4202.44 * U1
##  + 142.4 * U2
##  + 0.06 * Wealth
##  + 61.75 * Ineq
##  + -4008.13 * Prob
##  + 0 * Time
```