Crime Rate

Load Data

```
rm(list = ls())

library(tidyverse)
library(DAAG)
library(car)

crime_data = read.table("uscrime.txt.",
                        sep="",
                        fill=FALSE,
                        strip.white=TRUE,
                        header = TRUE)

#test data
crime_test <- data.frame(M = 14.0, So = 0,
                         Ed = 10.0, Po1 = 12.0,
                         Po2 = 15.5, LF = 0.640,
                         M.F = 94.0, Pop = 150,
                         NW = 1.1, U1 = 0.120,
                         U2 = 3.6, Wealth = 3200,
                         Ineq = 20.1, Prob = 0.04,
                         Time = 39.0)
```

Loading in the three packages that will be used throughout the problem. Next is just setting the working directory and reading in the crime data that was give to us. The crime test data is the information about the city which we are trying to predict the crime rate for. Once we build the model we are trying to predict the crime rate given those values about the city and then see how well our model does.

Data Exploration

```
head(crime_data)
```

```
##      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
```

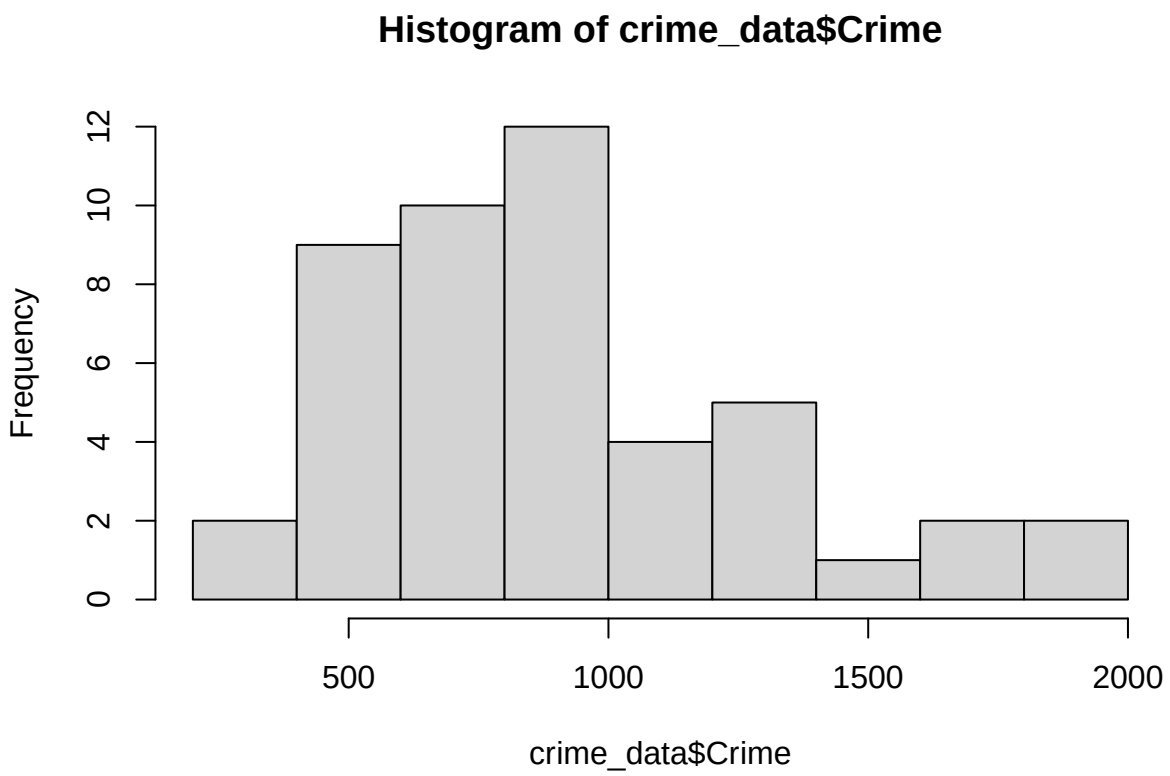
```
summary(crime_data)
```

```
##        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
```

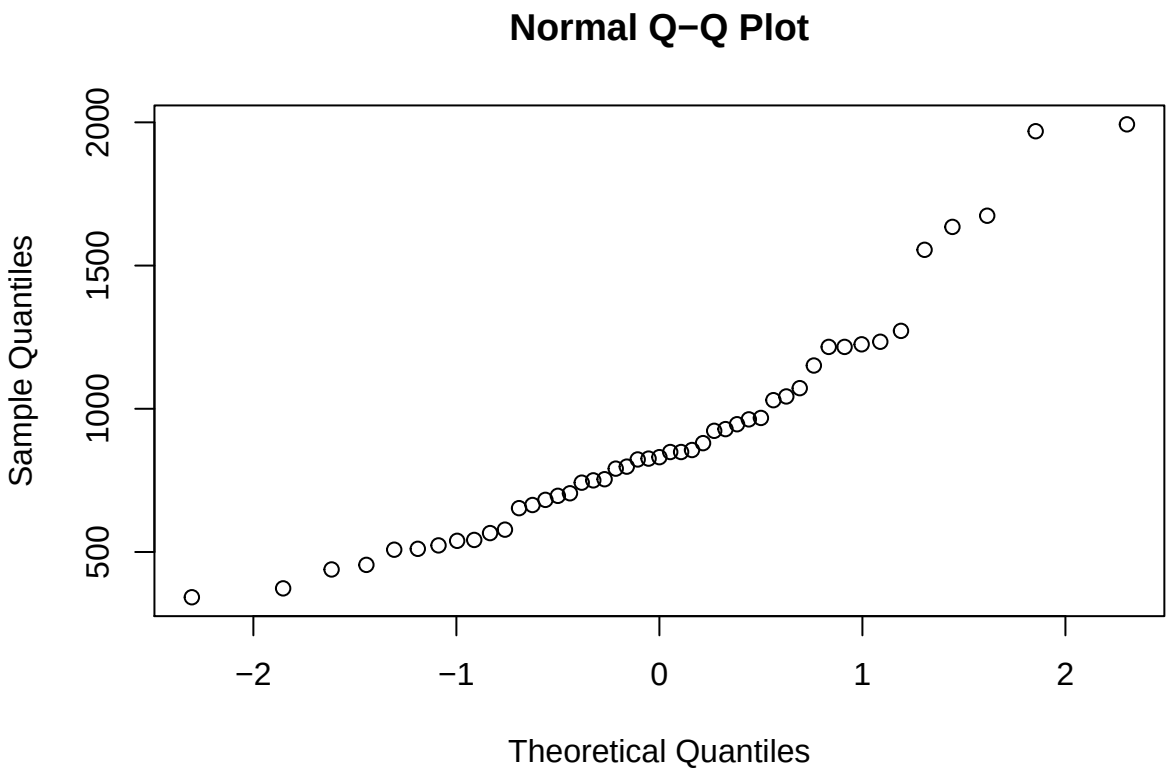
```
summary(crime_data[,12:16])
```

```
##      Wealth          Ineq            Prob              Time
##  Min.   :2880   Min.   :12.60   Min.   :0.00690   Min.   :12.20
##  1st Qu.:4595   1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60
##  Median :5370   Median :17.60   Median :0.04210   Median :25.80
##  Mean   :5254   Mean   :19.40   Mean   :0.04709   Mean   :26.60
##  3rd Qu.:5915   3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45
##  Max.   :6890   Max.   :27.60   Max.   :0.11980   Max.   :44.00
##      Crime
##  Min.   : 342.0
##  1st Qu.: 658.5
##  Median : 831.0
##  Mean   : 905.1
##  3rd Qu.:1057.5
##  Max.   :1993.0
```

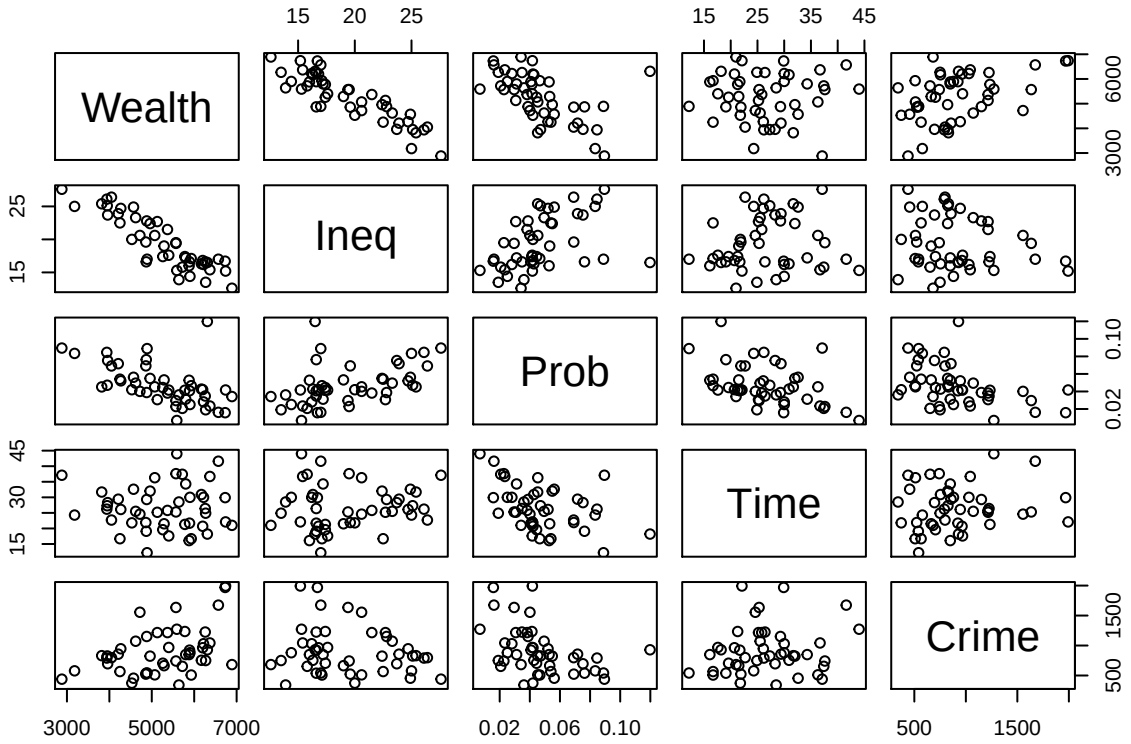
```
hist(crime_data$Crime)
```




```
qqnorm(crime_data$Crime)
```




```
pairs(crime_data[,12:16])
```




```
cor(crime_data[,12:16])
```

```
##               Wealth       Ineq       Prob          Time      Crime
## Wealth  1.0000000000 -0.8839973 -0.5553347  0.0006485587  0.4413199
## Ineq   -0.8839972758  1.0000000  0.4653219  0.1018228182 -0.1790237
## Prob   -0.5553347075  0.4653219  1.0000000 -0.4362462614 -0.4274222
## Time    0.0006485587  0.1018228 -0.4362463  1.0000000000  0.1498661
## Crime   0.4413199490 -0.1790237 -0.4274222  0.1498660617  1.0000000
```

```
#wealth and ineq are correlated
cor(crime_data)
```

```
##                  M          So          Ed         Po1         Po2         LF
## M        1.00000000  0.58435534 -0.53023964 -0.50573690 -0.51317336 -0.1609488
## So       0.58435534  1.00000000 -0.70274132 -0.37263633 -0.37616753 -0.5054695
## Ed      -0.53023964 -0.70274132  1.00000000  0.48295213  0.49940958  0.5611780
## Po1     -0.50573690 -0.37263633  0.48295213  1.00000000  0.99358648  0.1214932
## Po2     -0.51317336 -0.37616753  0.49940958  0.99358648  1.00000000  0.1063496
## LF      -0.16094882 -0.50546948  0.56117795  0.12149320  0.10634960  1.0000000
## M.F     -0.02867993 -0.31473291  0.43691492  0.03376027  0.02284250  0.5135588
## Pop     -0.28063762 -0.04991832 -0.01722740  0.52628358  0.51378940 -0.1236722
## NW       0.59319826  0.76710262 -0.66488190 -0.21370878 -0.21876821 -0.3412144
## U1      -0.22438060 -0.17241931  0.01810345 -0.04369761 -0.05171199 -0.2293997
## U2      -0.24484339  0.07169289 -0.21568155  0.18509304  0.16922422 -0.4207625
## Wealth -0.67005506 -0.63694543  0.73599704  0.78722528  0.79426205  0.2946323
```

```
## Ineq    0.63921138  0.73718106 -0.76865789 -0.63050025 -0.64815183 -0.2698865
## Prob    0.36111641  0.53086199 -0.38992286 -0.47324704 -0.47302729 -0.2500861
## Time    0.11451072  0.06681283 -0.25397355  0.10335774  0.07562665 -0.1236404
## Crime  -0.08947240 -0.09063696  0.32283487  0.68760446  0.66671414  0.1888663
##                M.F         Pop          NW           U1          U2
## M      -0.02867993 -0.28063762  0.59319826 -0.224380599 -0.24484339
## So     -0.31473291 -0.04991832  0.76710262 -0.172419305  0.07169289
## Ed      0.43691492 -0.01722740 -0.66488190  0.018103454 -0.21568155
## Po1     0.03376027  0.52628358 -0.21370878 -0.043697608  0.18509304
## Po2     0.02284250  0.51378940 -0.21876821 -0.051711989  0.16922422
## LF      0.51355879 -0.12367222 -0.34121444 -0.229399684 -0.42076249
## M.F     1.00000000 -0.41062750 -0.32730454  0.351891900 -0.01869169
## Pop    -0.41062750  1.00000000  0.09515301 -0.038119948  0.27042159
## NW     -0.32730454  0.09515301  1.00000000 -0.156450020  0.08090829
## U1      0.35189190 -0.03811995 -0.15645002  1.000000000  0.74592482
## U2     -0.01869169  0.27042159  0.08090829  0.745924815  1.00000000
## Wealth  0.17960864  0.30826271 -0.59010707  0.044857202  0.09207166
## Ineq   -0.16708869 -0.12629357  0.67731286 -0.063832178  0.01567818
## Prob   -0.05085826 -0.34728906  0.42805915 -0.007469032 -0.06159247
## Time   -0.42769738  0.46421046  0.23039841 -0.169852838  0.10135833
## Crime   0.21391426  0.33747406  0.03259884 -0.050477918  0.17732065
##               Wealth        Ineq         Prob          Time       Crime
## M      -0.6700550558  0.63921138  0.361116408  0.1145107190 -0.08947240
## So     -0.6369454328  0.73718106  0.530861993  0.0668128312 -0.09063696
## Ed      0.7359970363 -0.76865789 -0.389922862 -0.2539735471  0.32283487
## Po1     0.7872252807 -0.63050025 -0.473247036  0.1033577449  0.68760446
## Po2     0.7942620503 -0.64815183 -0.473027293  0.0756266536  0.66671414
## LF      0.2946323090 -0.26988646 -0.250086098 -0.1236404364  0.18886635
## M.F     0.1796086363 -0.16708869 -0.050858258 -0.4276973791  0.21391426
## Pop     0.3082627091 -0.12629357 -0.347289063  0.4642104596  0.33747406
## NW     -0.5901070652  0.67731286  0.428059153  0.2303984071  0.03259884
## U1      0.0448572017 -0.06383218 -0.007469032 -0.1698528383 -0.05047792
## U2      0.0920716601  0.01567818 -0.061592474  0.1013583270  0.17732065
## Wealth  1.0000000000 -0.88399728 -0.555334708  0.0006485587  0.44131995
## Ineq   -0.8839972758  1.00000000  0.465321920  0.1018228182 -0.17902373
## Prob   -0.5553347075  0.46532192  1.000000000 -0.4362462614 -0.42742219
## Time    0.0006485587  0.10182282 -0.436246261  1.0000000000  0.14986606
## Crime   0.4413199490 -0.17902373 -0.427422188  0.1498660617  1.00000000
```

```
#Po1 and Po2 highly correlated
#wealth seems to be correlated to most predictors
```

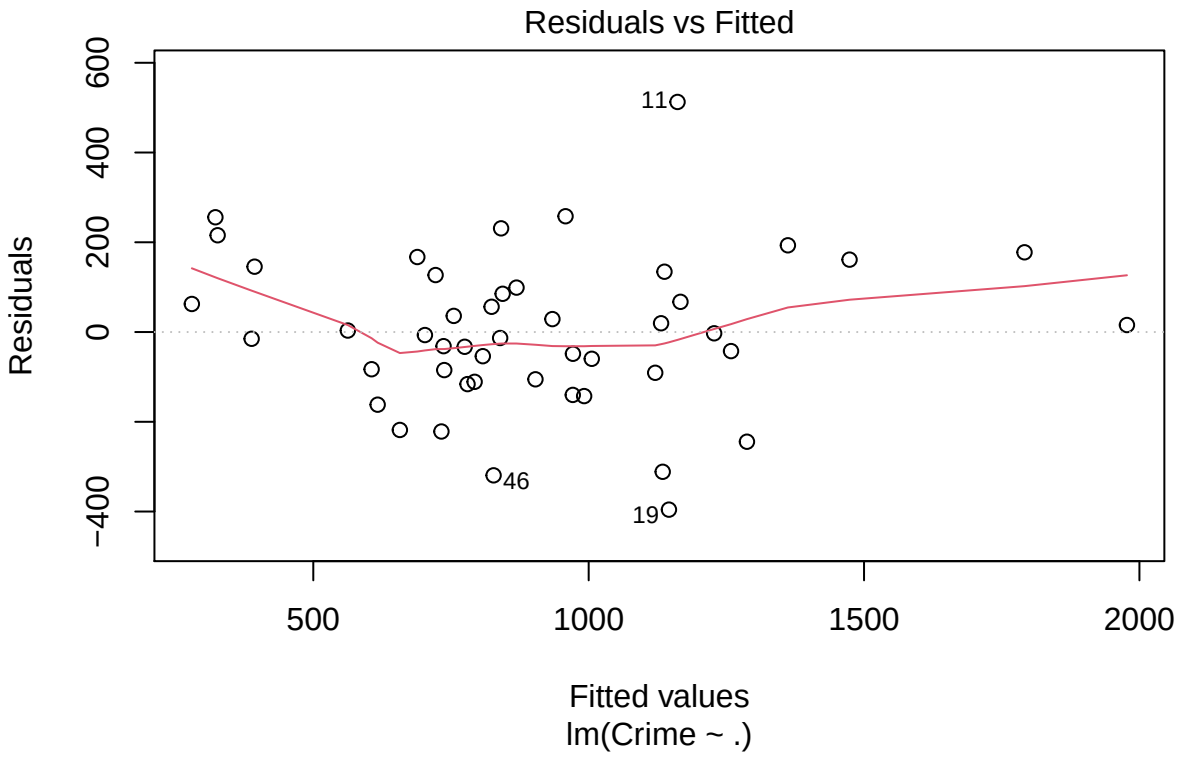
Next, lets look at the data and see what we are working with. From the head tab there seems to be 16 variables in the data set with only 47 observations. Already we know our model wont be the best because the amount of observations is really low and a lot of predictors in the set. The histogram gives us an idea of the data which should that a lot of the crime rates is beloew 1500 and even a lot of the frequency is below 1,000. The QQ plot shows that the distrubtion is fairly normal but at the end there seems to be some fluctuation. The correlation plot and chart shows how closely correlated the predicator are to each other. Wealth and Ineq seem to be highly correlated and well as Po1 and Po2. Predictors that are highly correlated could lead to false over stating the importance of a predictor.

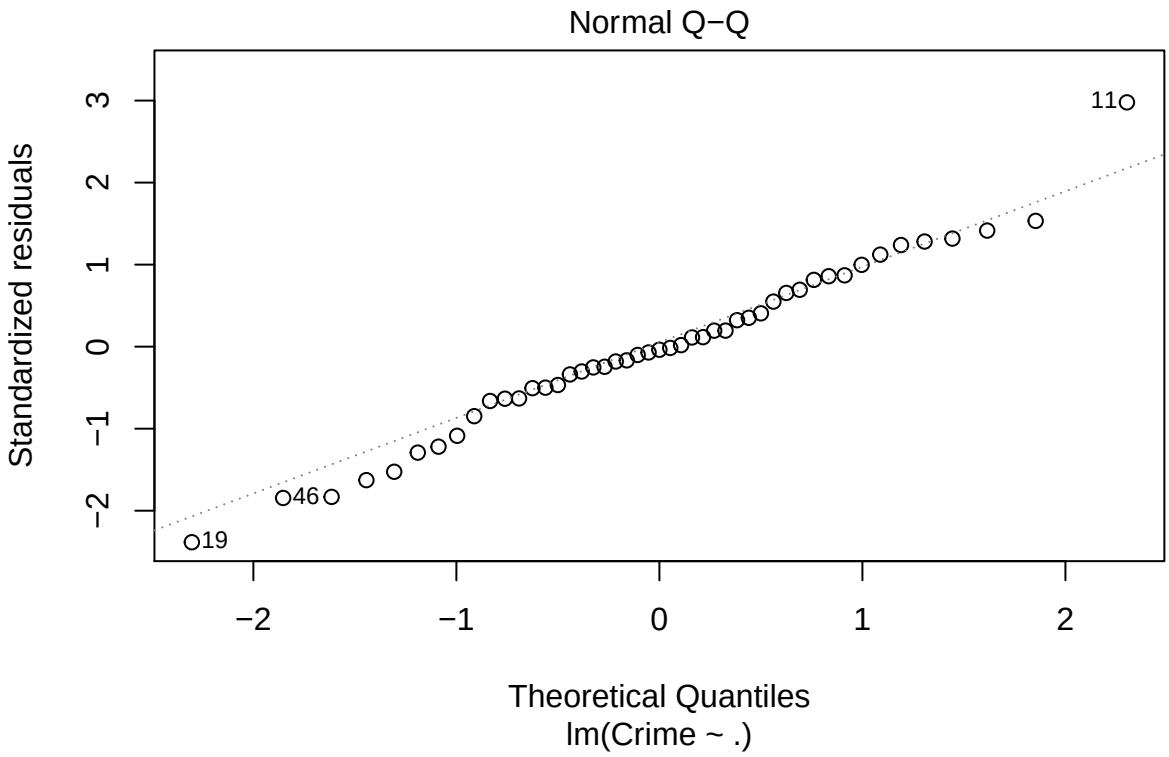
LM Models

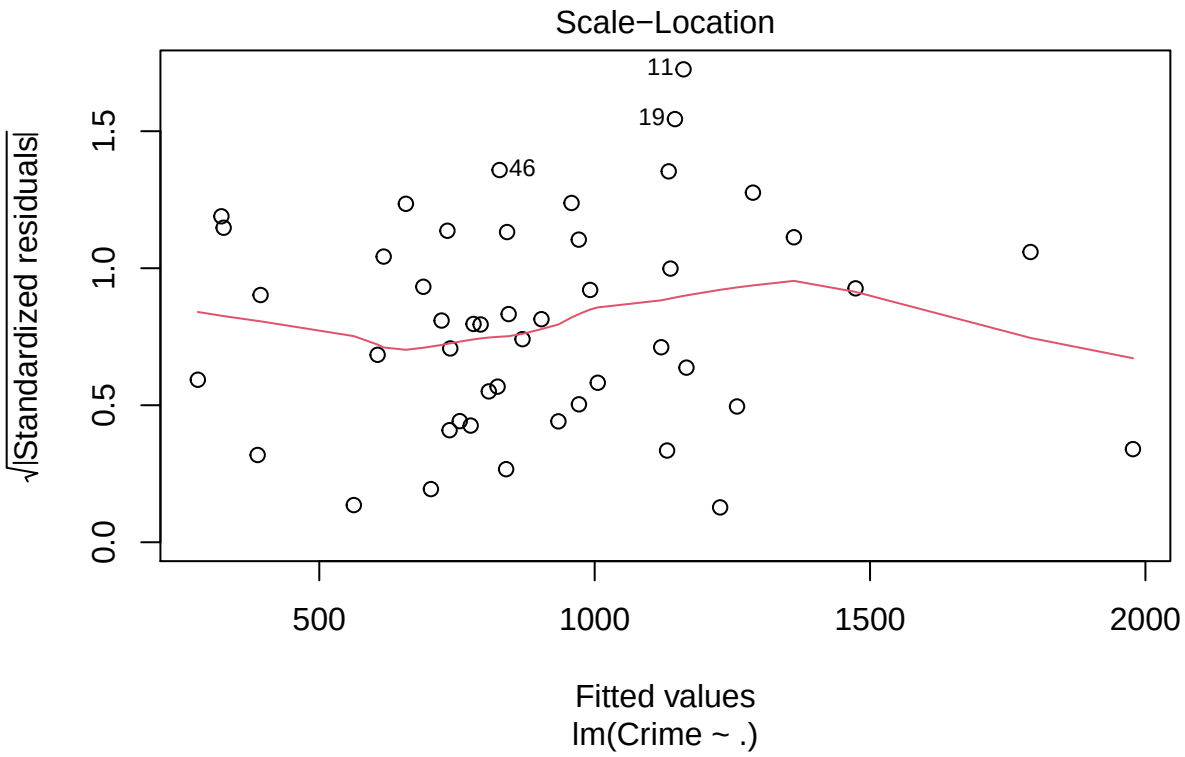
```
#run some models
lm_model <- lm(Crime ~. , data = crime_data)
summary(lm_model)
```

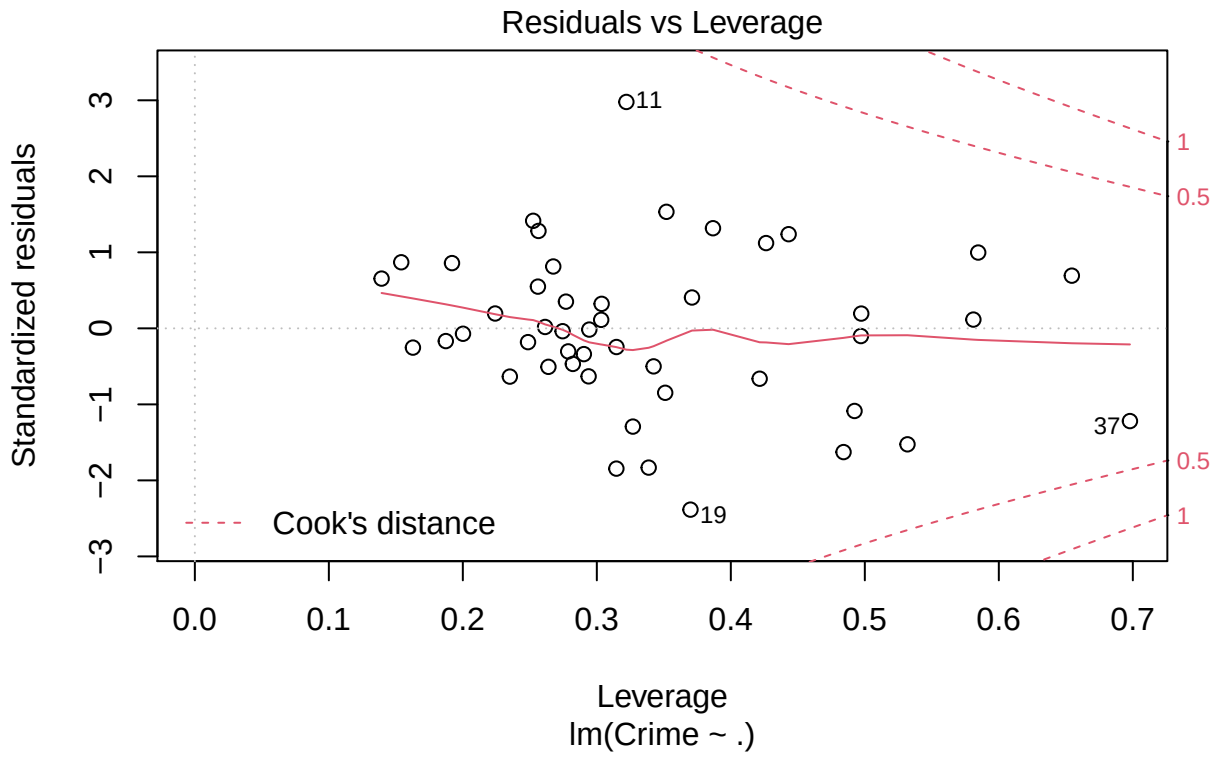
```
##
## Call:
## lm(formula = Crime ~ ., data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            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
```

```
plot(lm_model)
```


Residuals vs Fitted
Residuals
600
400
200
0
−400
11
46
19
500
1000
1500
2000
Fitted values
lm(Crime ~ .)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Crime ~ .)
11
46
19
−2
−1
0
1
2
3


Scale−Location
11
19
46
√|Standardized residuals|
0.0
0.5
1.0
1.5
500
1000
1500
2000
Fitted values
lm(Crime ~ .)




```
vif(lm_model)
```

```
##          M         So         Ed        Po1        Po2         LF        M.F
##   2.892448   5.342783   5.077447 104.658667 113.559262   3.712690   3.785934
##        Pop         NW         U1         U2     Wealth       Ineq       Prob
##   2.536708   4.674088   6.063931   5.088880  10.530375   8.644528   2.809459
##       Time
##   2.713785
```

```
#values greater than 10 are problematic and we have three of them
# Po1, Po2, and wealth
#data seems to be funky
dwt(lm_model)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.1303644      1.723274   0.332
##  Alternative hypothesis: rho != 0
```

```
# want to gets values close to 2 which we got 1.72
```

The first model is ran with all 15 predictors to just give us a baseline. We can see from the summary output that only 6 predictors have a significant relationship, so we are going to focus on those predictors. There are other ways of determining variable importance but for this homework we will use the previously stated method. Plotting the model gives us four different plots: first the residuals are within reason, 2nd is the

qq norm which fairly normal distrubuted, 3rd the scale is never greater than 1, and finally the residuals are never greater thean the min or max. Vif and Dwt are from the car package which also tests normality. Vif values that are too large can create problems which three are and dwt you want a value lower than 2 which we achieved.

```
#cut off 0.05
lm_model2 <- lm(Crime ~ M + Ed + Ineq + Prob,
                data = crime_data)
summary(lm_model2)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Ineq + Prob, data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -532.97 -254.03  -55.72  137.80  960.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1339.35    1247.01  -1.074  0.28893
## M              35.97      53.39   0.674  0.50417
## Ed            148.61      71.92   2.066  0.04499 *
## Ineq           26.87      22.77   1.180  0.24458
## Prob        -7331.92    2560.27  -2.864  0.00651 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 347.5 on 42 degrees of freedom
## Multiple R-squared:  0.2629, Adjusted R-squared:  0.1927
## F-statistic: 3.745 on 4 and 42 DF,  p-value: 0.01077
```

```
#cut off 0.1
lm_model3 <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob,
                data = crime_data)
summary(lm_model3)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_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
```

I ran multiple models and decided that these two are good represenations. The first model looks at the p values less than 0.05 from the summary of the first model. The first model did output 6 varaibles but two of these did not make the .05 threshold. From the output we can see that the R-squared and adjusted R both went down signifcantly. Those R values aren't the best comparison but does show that are model is less accruate. However, the first model could be overfitting since we have so many varaibles and not a lot of data points. The third model uses a higher cut off value of 0.1 which includes the predictors Po1 and U2. The R values for the third model are a lot higher but let's see a better comparison of fit.

```
predict1 <- predict(lm_model, crime_test)
predict1
```

```
##        1
## 155.4349
```

```
range(crime_data$Crime)
```

```
## [1]  342 1993
```

```
predict2 <- predict(lm_model2, crime_test)
predict2
```

```
##        1
## 897.2307
```

```
predict3 <- predict(lm_model3, crime_test)
predict3
```

```
##        1
## 1304.245
```

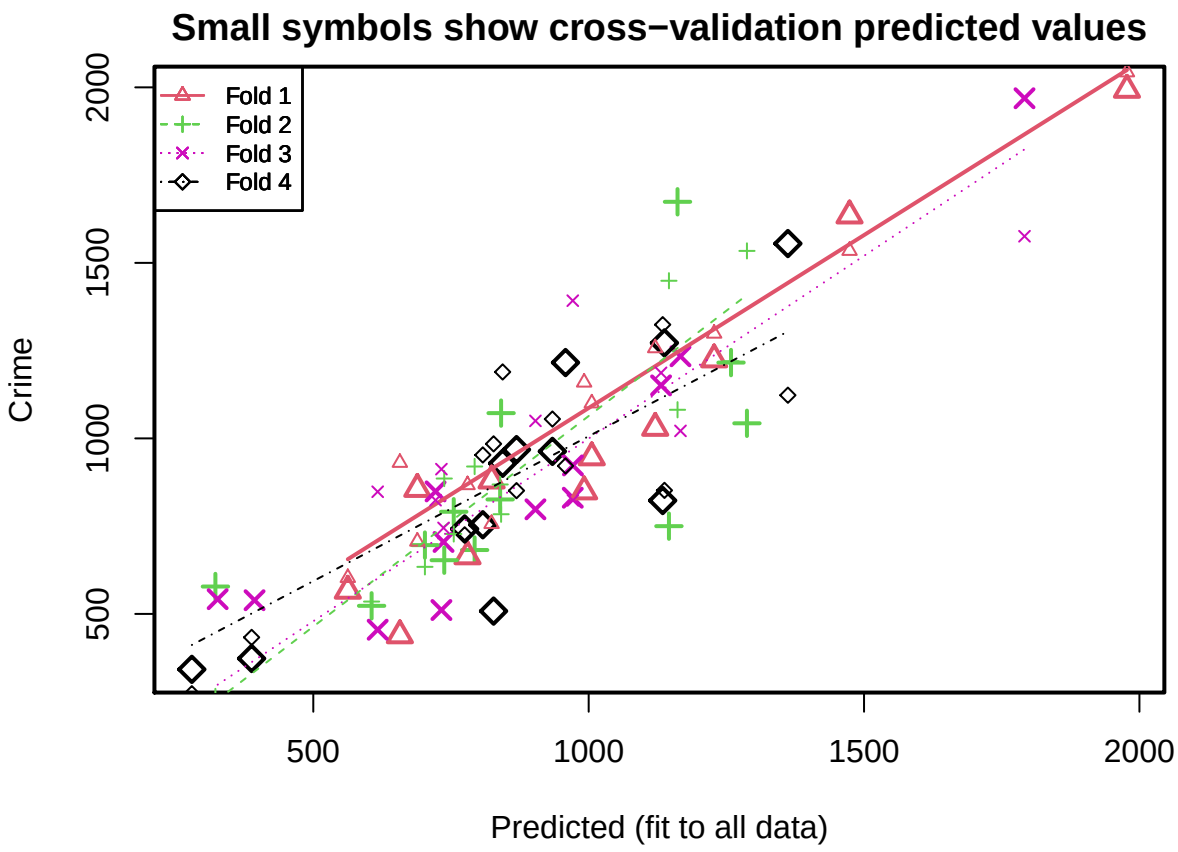
Now that we have the three models lets predict what crime rate value each of them will produce. The first model predicts a crime rate of 155 which if you remeber from the data exploration section seems a little low. Let's look at the range for crime rate values which we see the lowest is 342. That minmum number is over double the rate that we predicted from the first model. The other two models predict a crime rate value within the current range of values. That makes me think that model 2 and 3 are going to be a better prediction than the first model.

Best fit Model

```r
set.seed(4147)
#cross validate models
cv_model1 <- cv.lm(crime_data, lm_model, m=4)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## M          1   55084   55084    1.26  0.2702
## So         1   15370   15370    0.35  0.5575
## Ed         1  905668  905668   20.72 7.7e-05 ***
## Po1        1 3076033 3076033   70.38 1.8e-09 ***
## Po2        1  153024  153024    3.50  0.0708 .
## LF         1   61134   61134    1.40  0.2459
## M.F        1  111000  111000    2.54  0.1212
## Pop        1   42649   42649    0.98  0.3309
## NW         1   14197   14197    0.32  0.5728
## U1         1    7065    7065    0.16  0.6904
## U2         1  269663  269663    6.17  0.0186 *
## Wealth     1   34748   34748    0.79  0.3795
## Ineq       1  547423  547423   12.52  0.0013 **
## Prob       1  222620  222620    5.09  0.0312 *
## Time       1   10304   10304    0.24  0.6307
## Residuals 31 1354946   43708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Warning in cv.lm(crime_data, lm_model, m = 4):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 11
##                2   9   14   16     20   22     26    38  41   44   47
## Predicted   1474 689  780 1006 1227.8  657 1977.4 562.7 824 1121  992
## cvpred      1535 706  867 1100 1298.9  931 2043.3 602.8 757 1257 1159
## Crime       1635 856  664  946 1225.0  439 1993.0 566.0 880 1030  849
## CV residual  100 150 -203 -154  -73.9 -492  -50.3 -36.8 123 -227 -310
##
## Sum of squares = 512057    Mean square = 46551    n = 11
##
## fold 2
## Observations in test set: 12
##                 1   3    6   11   19    25      28   29  30   33   35    39
## Predicted   755.0 322  793 1161 1146 605.9 1258.48 1287 703  841  738 839.3
## cvpred      727.7 265  920 1082 1449 535.1 1219.78 1534 634  784  886 868.7
## Crime       791.0 578  682 1674  750 523.0 1216.00 1043 696 1072  653 826.0
## CV residual  63.3 313 -238  592 -699 -12.1   -3.78 -491  62  288 -233 -42.7
##
## Sum of squares = 1382466    Mean square = 115205    n = 12
##
## fold 3
## Observations in test set: 12
##                4    5    10  12   13   15  17    34   37     40    42   45
## Predicted   1791 1167 736.5 722  733  903 393 971.5  971 1131.5 326.3  617
## cvpred      1576 1021 745.1 824  912 1050 103 823.4 1392 1186.8 -85.5  848
```

```
## Crime       1969 1234 705.0 849  511  798 539 923.0  831 1151.0 542.0  455
## CV residual  393  213 -40.1  25 -401 -252 436  99.6 -561  -35.8 627.5 -393
##
## Sum of squares = 1491541    Mean square = 124295    n = 12
##
## fold 4
## Observations in test set: 12
##                  7    8   18    21   23  24    27    31   32   36   43   46
## Predicted    934.2 1362  844 774.9  958 869 279.5 388.0  808 1138 1134  827
## cvpred      1055.1 1123 1189 725.3  922 851 272.7 433.1  953  852 1324  984
## Crime        963.0 1555  929 742.0 1216 968 342.0 373.0  754 1272  823  508
## CV residual  -92.1  432 -260  16.7  294 117  69.3 -60.1 -199  420 -501 -476
##
## Sum of squares = 1065774    Mean square = 88814    n = 12
##
## Overall (Sum over all 12 folds)
##    ms
## 94720
```

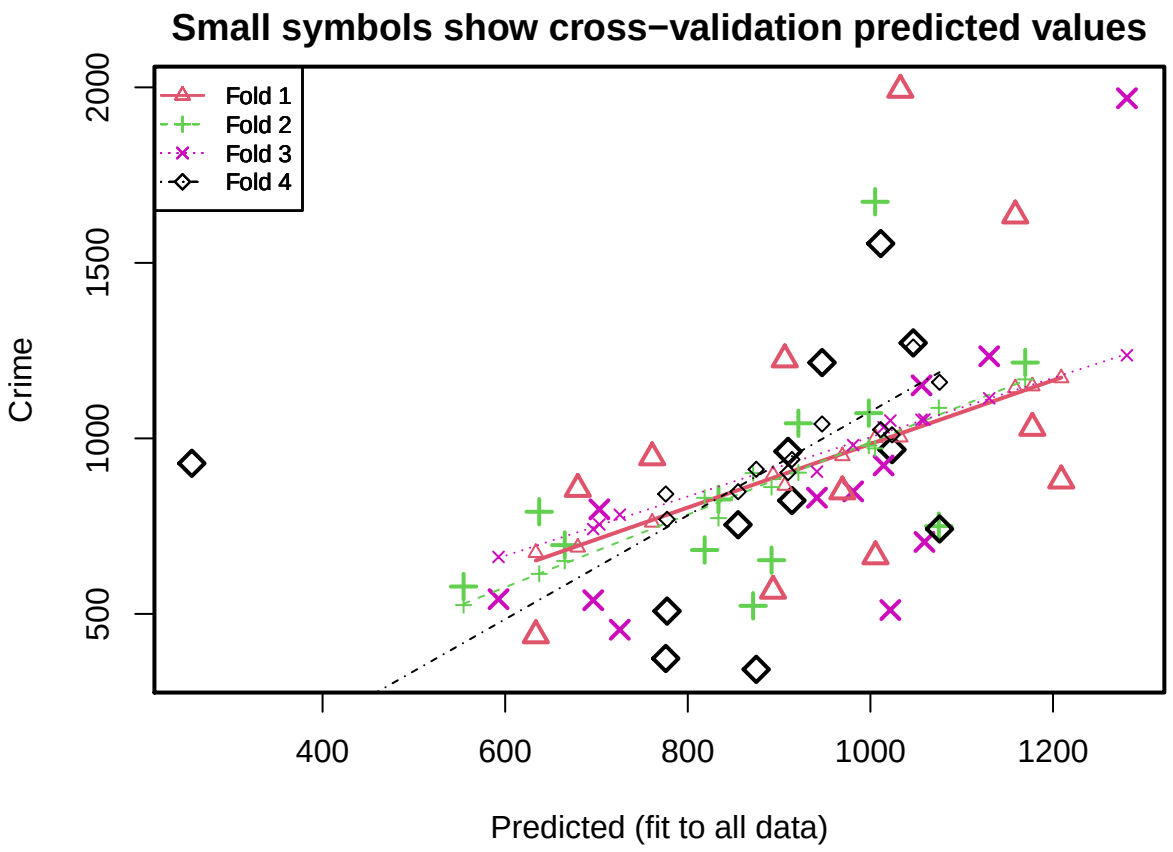
```
#calculate the root squared error
sse <- 94720 *nrow(crime_data)
sst <- sum((crime_data$Crime - mean(crime_data$Crime))^2)
rsq <- 1 - sse / sst
rsq
```

```
## [1] 0.353
```

```
cv_model2 <- cv.lm(crime_data, lm_model2, m=4)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value Pr(>F)
## M          1   55084   55084    0.46 0.5031
## Ed         1  725967  725967    6.01 0.0185 *
## Ineq       1   37674   37674    0.31 0.5794
## Prob       1  990334  990334    8.20 0.0065 **
## Residuals 42 5071868  120759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Warning in cv.lm(crime_data, lm_model2, m = 4):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 11
##                2   9   14  16   20   22   26   38   41   44   47
## Predicted   1159 680 1006 761  906  634 1033  893 1209 1177  969
## cvpred      1143 689 1002 760  866  674 1003  896 1172 1149  950
## Crime       1635 856  664 946 1225  439 1993  566  880 1030  849
## CV residual  492 167 -338 186  359 -235  990 -330 -292 -119 -101
##
## Sum of squares = 1800673    Mean square = 163698    n = 11
##
## fold 2
## Observations in test set: 12
##                1     3    6   11   19   25     28   29  30     33   35     39
## Predicted   637 554.4  818 1005 1075  872 1169.5  921 665  998.2  892 833.6
## cvpred      614 525.2  831  972 1087  901 1168.1  902 651  979.2  861 772.8
## Crime       791 578.0  682 1674  750  523 1216.0 1043 696 1072.0  653 826.0
## CV residual 177  52.8 -149  702 -337 -378   47.9  141  45   92.8 -208  53.2
##
## Sum of squares = 885058    Mean square = 73755    n = 12
##
## fold 3
## Observations in test set: 12
##                4    5   10   12   13    15   17   34    37     40   42   45
## Predicted   1281 1130 1059  981 1022 703.2  696 1014 941.5 1056.0  593  726
## cvpred      1237 1114 1054  981 1051 754.7  742 1032 905.3 1053.1  662  783
```

```
## Crime       1969 1234  705  849  511 798.0  539  923 831.0 1151.0  542  455
## CV residual  732  120 -349 -132 -540  43.3 -203 -109 -74.3   97.9 -120 -328
##
## Sum of squares = 1172278    Mean square = 97690    n = 12
##
## fold 4
## Observations in test set: 12
##                 7    8    18   21   23     24   27   31  32     36   43   46
## Predicted   909.8 1011 256.8 1076  947 1023.7  875  776 855 1046.9  914  777
## cvpred      902.6 1025 -56.1 1160 1041 1010.2  912  842 848 1260.2  940  769
## Crime       963.0 1555 929.0  742 1216  968.0  342  373 754 1272.0  823  508
## CV residual  60.4  530 985.1 -418  175  -42.2 -570 -469 -94   11.8 -117 -261
##
## Sum of squares = 2097384    Mean square = 174782    n = 12
##
## Overall (Sum over all 12 folds)
##     ms
## 126710
```

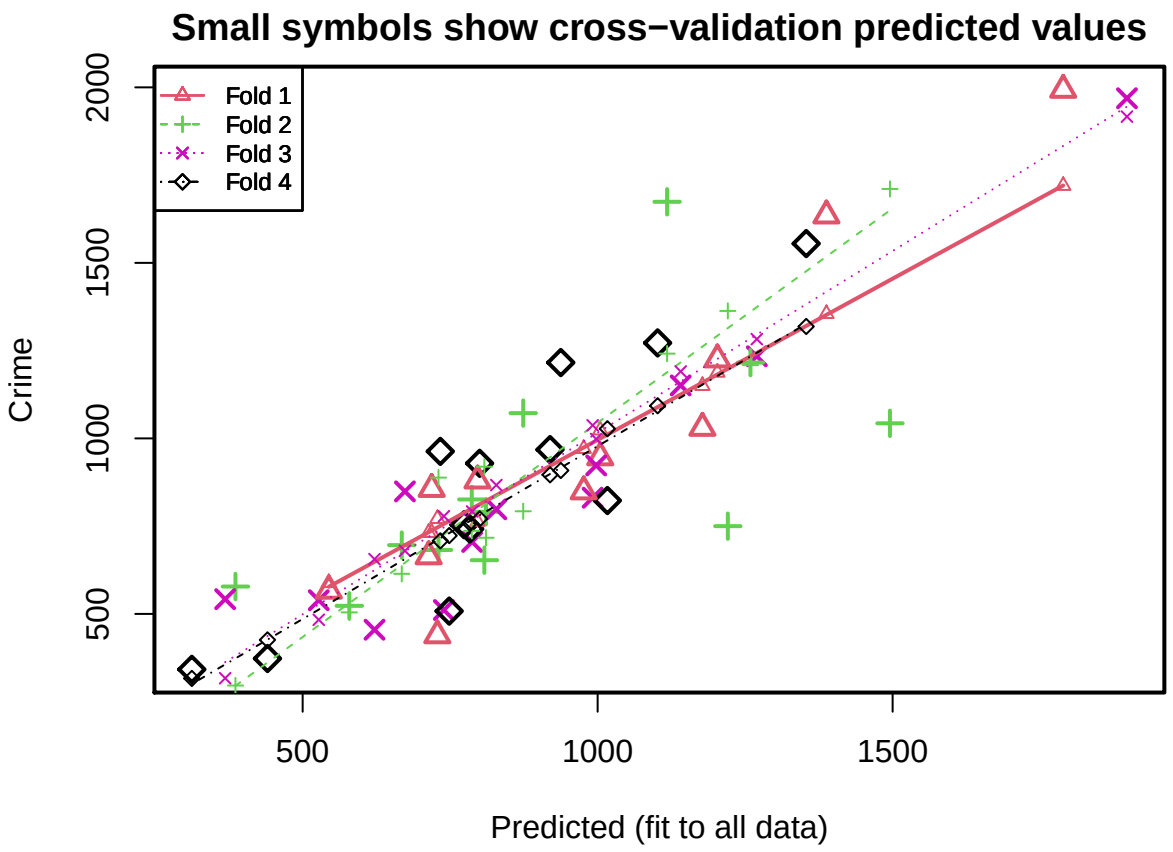
```
sse2 <- 126710 * nrow(crime_data)
sst2 <- sum((crime_data$Crime - mean(crime_data$Crime))^2)
rsq2 <- 1 - sse2 / sst2
rsq2
```

```
## [1] 0.135
```

```
cv_model3 <- cv.lm(crime_data, lm_model3, m=4)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## M          1   55084   55084    1.37 0.24914
## Ed         1  725967  725967   18.02 0.00013 ***
## Po1        1 3173852 3173852   78.80 5.3e-11 ***
## U2         1  217386  217386    5.40 0.02534 *
## Ineq       1  848273  848273   21.06 4.3e-05 ***
## Prob       1  249308  249308    6.19 0.01711 *
## Residuals 40 1611057   40276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Warning in cv.lm(crime_data, lm_model3, m = 4):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 11
##                2   9    14     16     20   22   26    38  41   44   47
## Predicted   1388 719 713.6 1004.4 1203.0  728 1789 544.4 796 1178  976
## cvpred      1355 731 731.1 1023.2 1187.6  771 1720 588.4 763 1150  970
## Crime       1635 856 664.0  946.0 1225.0  439 1993 566.0 880 1030  849
## CV residual  280 125 -67.1  -77.2   37.4 -332  273 -22.4 117 -120 -121
##
## Sum of squares = 334042    Mean square = 30367    n = 11
##
## fold 2
## Observations in test set: 12
##                 1   3    6   11   19    25     28   29    30   33   35    39
## Predicted   810.8 386  730 1118 1221 579.1 1259.0 1495 668.0  874  808 786.7
## cvpred      716.9 296  888 1241 1363 504.3 1208.7 1711 614.2  792  919 736.6
## Crime       791.0 578  682 1674  750 523.0 1216.0 1043 696.0 1072  653 826.0
## CV residual  74.1 282 -206  433 -613  18.7    7.3 -668  81.8  280 -266  89.4
##
## Sum of squares = 1300449    Mean square = 108371    n = 12
##
## fold 3
## Observations in test set: 12
##                  4      5    10  12   13  15    17    34   37     40  42   45
## Predicted   1897.2 1269.8 787.3 673  739 828 527.4 997.5  992 1140.8 369  622
## cvpred      1916.6 1282.8 791.8 680  778 867 483.3 998.2 1037 1190.7 317  656
```

```
## Crime       1969.0 1234.0 705.0 849  511 798 539.0 923.0  831 1151.0 542  455
## CV residual   52.4  -48.8 -86.8 169 -267 -69  55.7 -75.2 -206  -39.7 225 -201
##
## Sum of squares = 261503    Mean square = 21792    n = 12
##
## fold 4
## Observations in test set: 12
##               7    8  18  21   23    24    27  31  32   36   43   46
## Predicted   733 1354 800 783  938 919.4 312.2 440 774 1102 1017  748
## cvpred      708 1319 771 759  909 896.3 316.2 426 740 1093 1027  723
## Crime       963 1555 929 742 1216 968.0 342.0 373 754 1272  823  508
## CV residual 255  236 158 -17  307  71.7  25.8 -53  14  179 -204 -215
##
## Sum of squares = 369549    Mean square = 30796    n = 12
##
## Overall (Sum over all 12 folds)
##    ms
## 48203
```

```
sse3 <- 48203 * nrow(crime_data)
sst3 <- sum((crime_data$Crime - mean(crime_data$Crime))^2)
rsq3 <- 1 - sse3 / sst3
rsq3
```

```
## [1] 0.671
```

```
AIC(lm_model)
```

```
## [1] 650
```

```
AIC(lm_model2)
```

```
## [1] 690
```

```
AIC(lm_model3)
```

```
## [1] 640
```

Since we didn't split the data into a test or train set, we should cross validate the models in attempt to avoid overfitting. The R-squared for the first model is 0.353, the second model is 0.135, and the third model is 0.671. As you can see from the values, the third model is the best at prediciting crime rates. The second model had a lower value than the model with all the predictors which is somewhat suprising. We still could be overfitting with the first model though. The second test of fit is the AIC value which models with lower values are more accurate. Again, model 3 has the lowest value so I would say that it is the best of the models for predicting crime rate.