ISYE 6501 HW5

Ryan Cherry

2024-02-08

Question 8.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

I am currently trying to reach a certain weight that is more appropriate for my height. A linear regression model would be useful for determining what it will take to achieve that weight. Predictors could include type of food consumed, caloric intake, sleep in hours, amount of physical activity per week, unhealthy beverages consumed per week, and type of exercise per day, blood sugar level, and many others.

This way, I could reach the intended weight faster than the rate I am at currently.

Question 8.2

Using crime data from http://www.statsci.org/data/general/uscrime.txt (file uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with the following data: 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 Show your model (factors used and their coefficients), the software output, and the quality of fit.

Several different linear regression models with different numbers of predictors were tested to answer this question.

```
#read in us crime data set
crime <- read.table("C:/Users/ryanc/Downloads/uscrime.txt", header = TRUE)
head(crime)
```

```
##      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
```

I obtained the mean, median, standard deviation, minimum, maximum, range, skew, and standard error for each variable in the model in order to get a feel for the data. Based on the output, the variables each seem

to be on different scales, as evidenced by M, M.F. and Pop having significantly larger values for median, mean, and standard error than other predictors. Scaling the data goes beyond the scope of this analysis, but should be done if the results were to be presented to an employer.

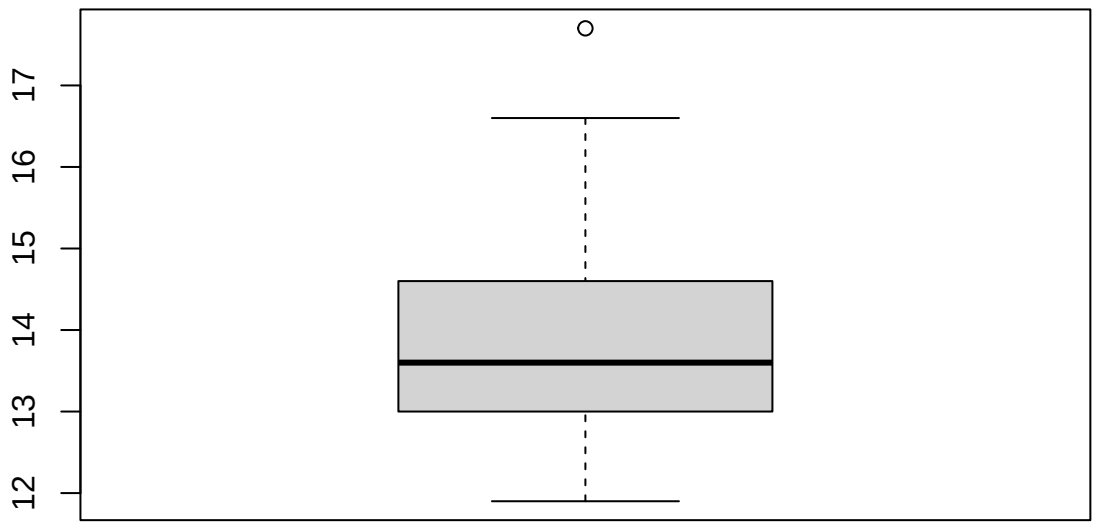
```
#obtain descriptive statistics including mean, median, standard deviation, min, max, range, skew, and s
Descriptive_stats <- describe(crime)
Descriptive_stats <- select(Descriptive_stats, n, mean, sd, median, min, max, range, skew, se)

Descriptive_stats
```

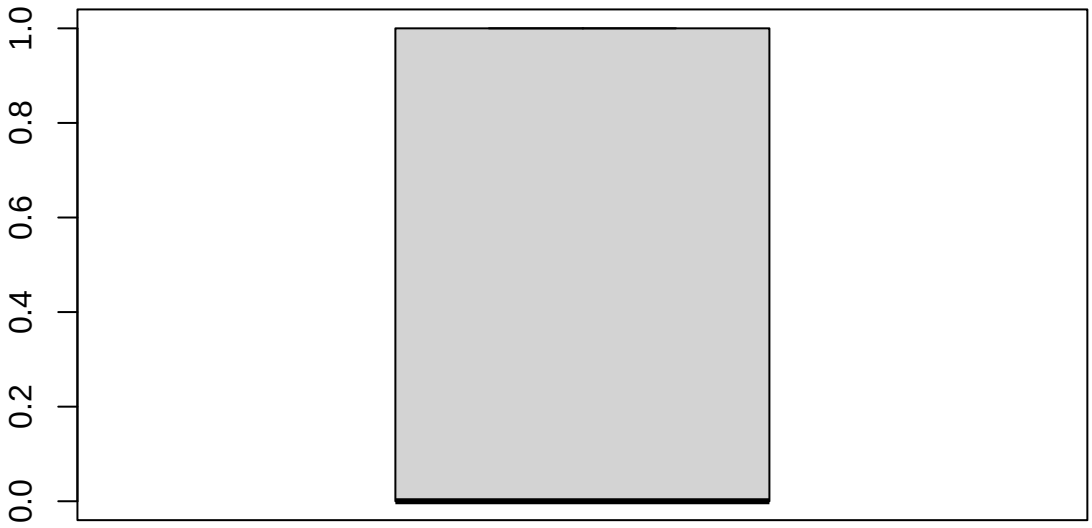
```
##         n    mean     sd  median     min     max   range  skew     se
## M      47   13.86   1.26   13.60   11.90   17.70    5.80  0.82   0.18
## So     47    0.34   0.48    0.00    0.00    1.00    1.00  0.65   0.07
## Ed     47   10.56   1.12   10.80    8.70   12.20    3.50 -0.32   0.16
## Po1    47    8.50   2.97    7.80    4.50   16.60   12.10  0.89   0.43
## Po2    47    8.02   2.80    7.30    4.10   15.70   11.60  0.84   0.41
## LF     47    0.56   0.04    0.56    0.48    0.64    0.16  0.27   0.01
## M.F    47   98.30   2.95   97.70   93.40  107.10   13.70  0.99   0.43
## Pop    47   36.62  38.07   25.00    3.00  168.00  165.00  1.85   5.55
## NW     47   10.11  10.28    7.60    0.20   42.30   42.10  1.38   1.50
## U1     47    0.10   0.02    0.09    0.07    0.14    0.07  0.77   0.00
## U2     47    3.40   0.84    3.40    2.00    5.80    3.80  0.54   0.12
## Wealth 47 5253.83 964.91 5370.00 2880.00 6890.00 4010.00 -0.38 140.75
## Ineq   47   19.40   3.99   17.60   12.60   27.60   15.00  0.37   0.58
## Prob   47    0.05   0.02    0.04    0.01    0.12    0.11  0.88   0.00
## Time   47   26.60   7.09   25.80   12.20   44.00   31.80  0.37   1.03
## Crime  47  905.09 386.76  831.00  342.00 1993.00 1651.00  1.05  56.42
```

In order to get a visual of each predictor in the crimes data set, box-plots were generated. From the results, several predictors such as M, Ed, Po1, Po2, u2, wealth, and inequality have a skewed distribution. In addition, outliers seem to be present in the upper tale of the data for M.f., population, nw, and prob predictors. All the other predictors seem to be normally distributed. One more thing of note is tat one of the variables, So, is categorical.

```
#create box-plots to view the distribution of each predictor
boxplot(crime$M, xlab = 'M')
```

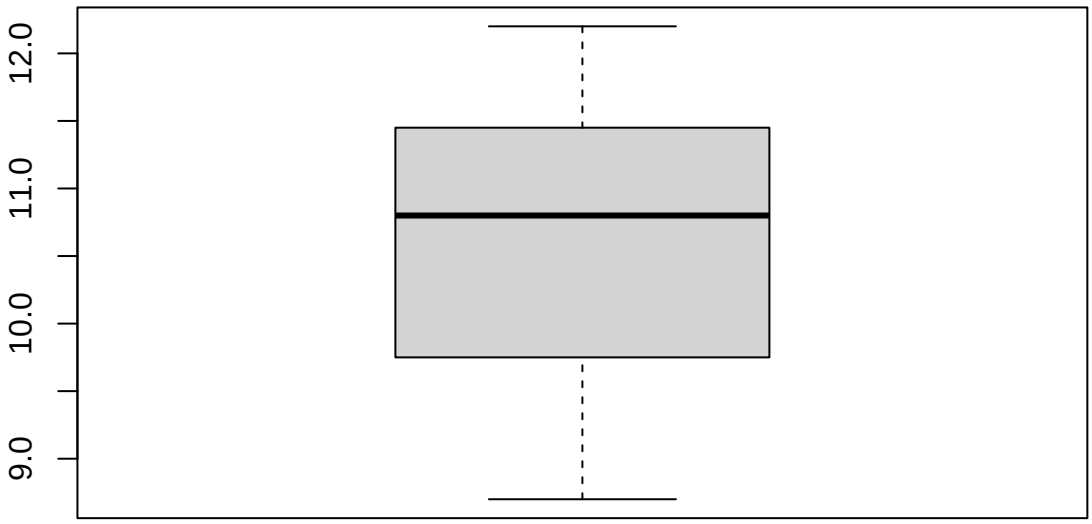



```
boxplot(crime$So, xlab = 'So')
```

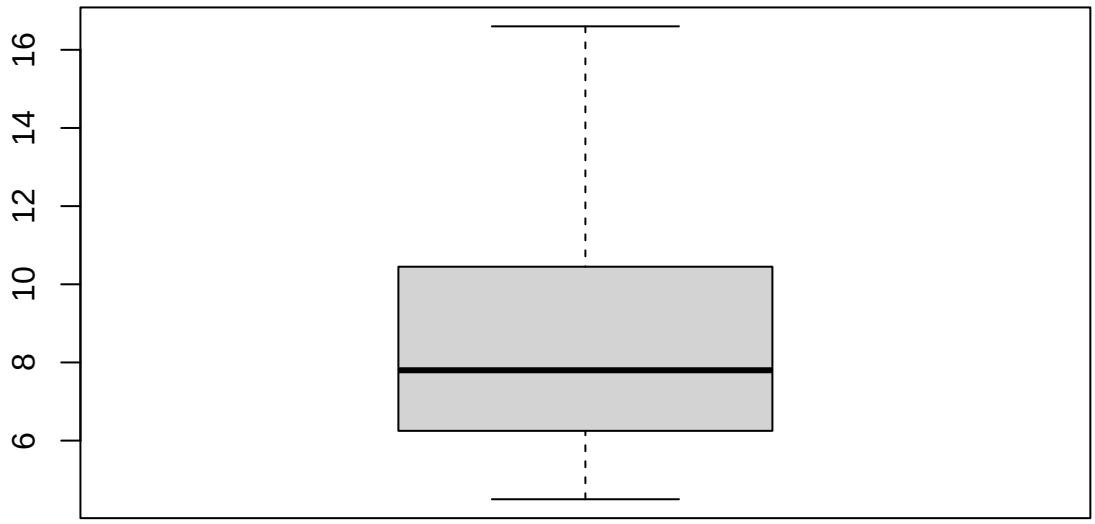


So

```
boxplot(crime$Ed, xlab = 'Ed')
```

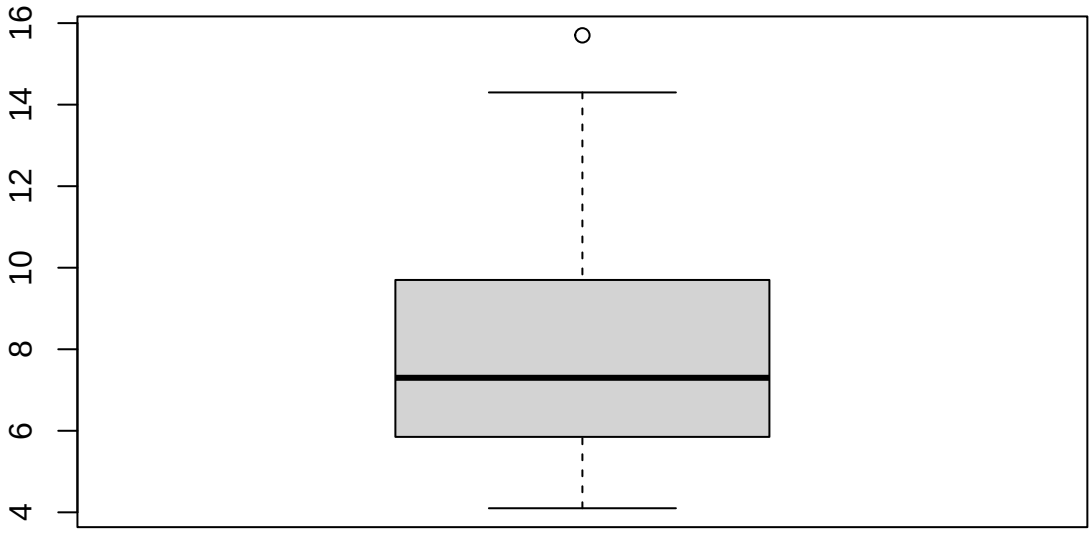



```
boxplot(crime$Po1, xlab = 'Po1')
```



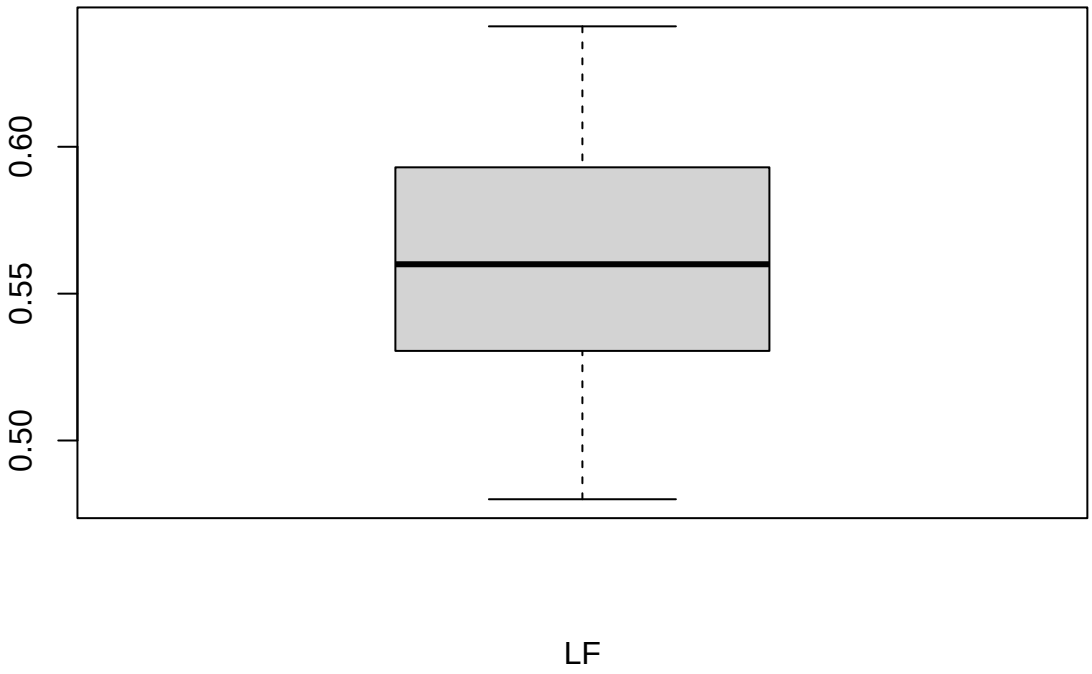
Po1

```
boxplot(crime$Po2, xlab = 'Po2')
```

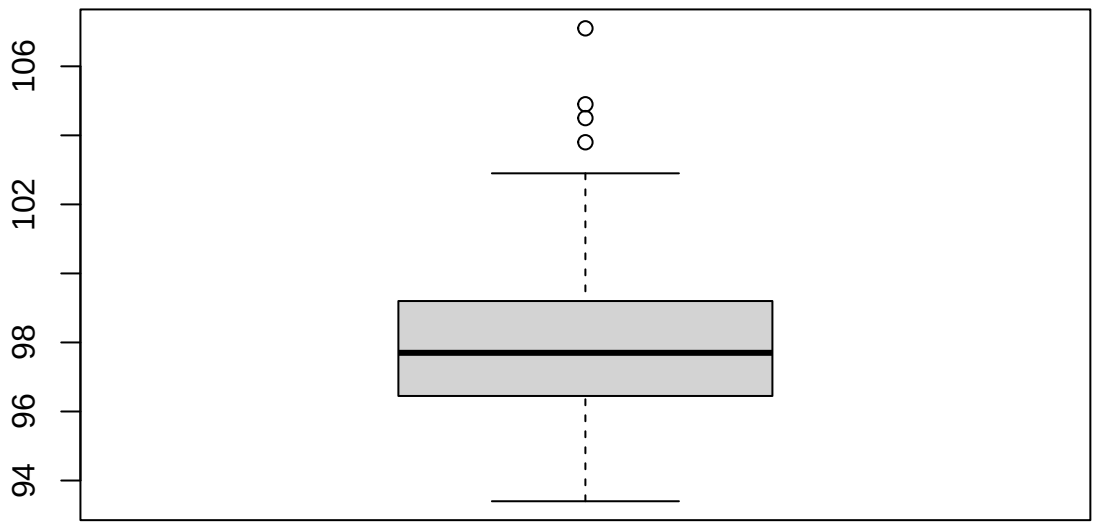


Po2

```
boxplot(crime$LF, xlab = 'LF')
```

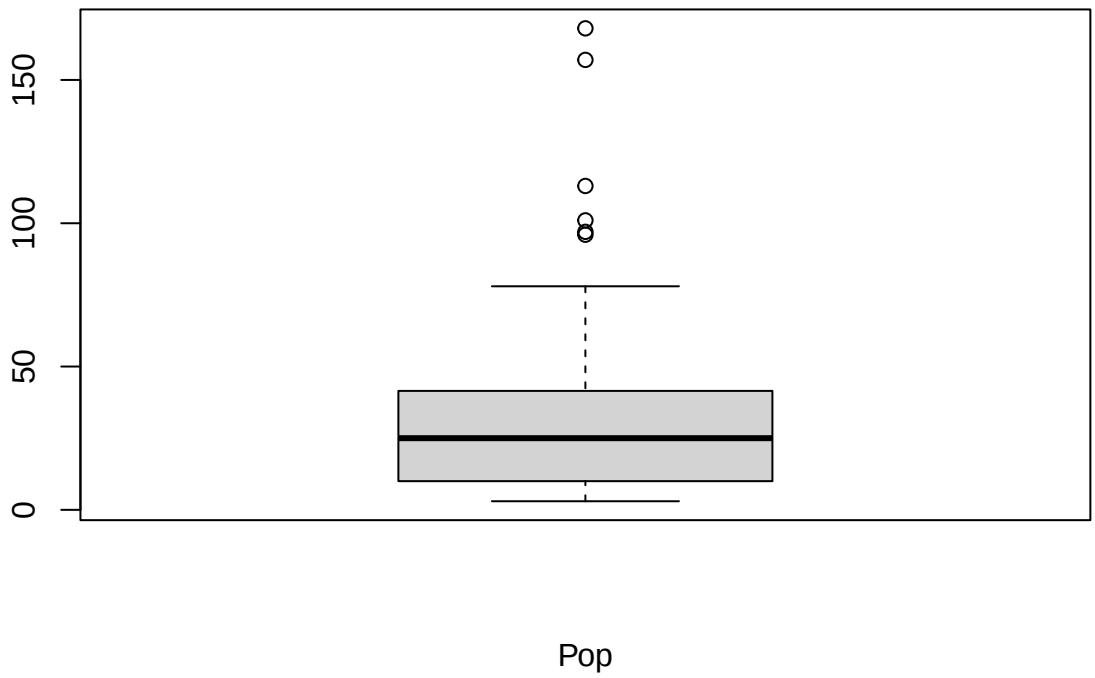



```
boxplot(crime$M.F, xlab = 'M.F.')
```

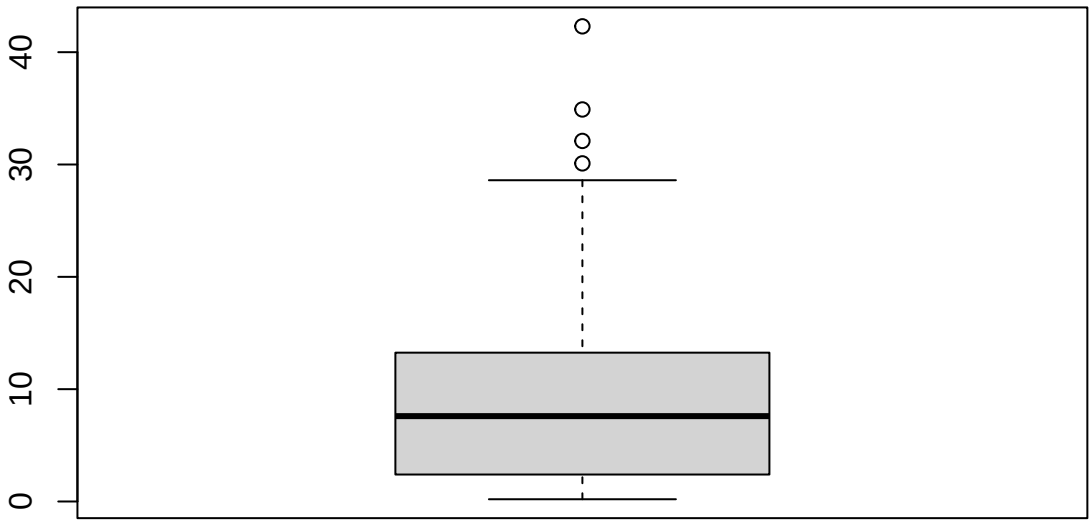


M.F.

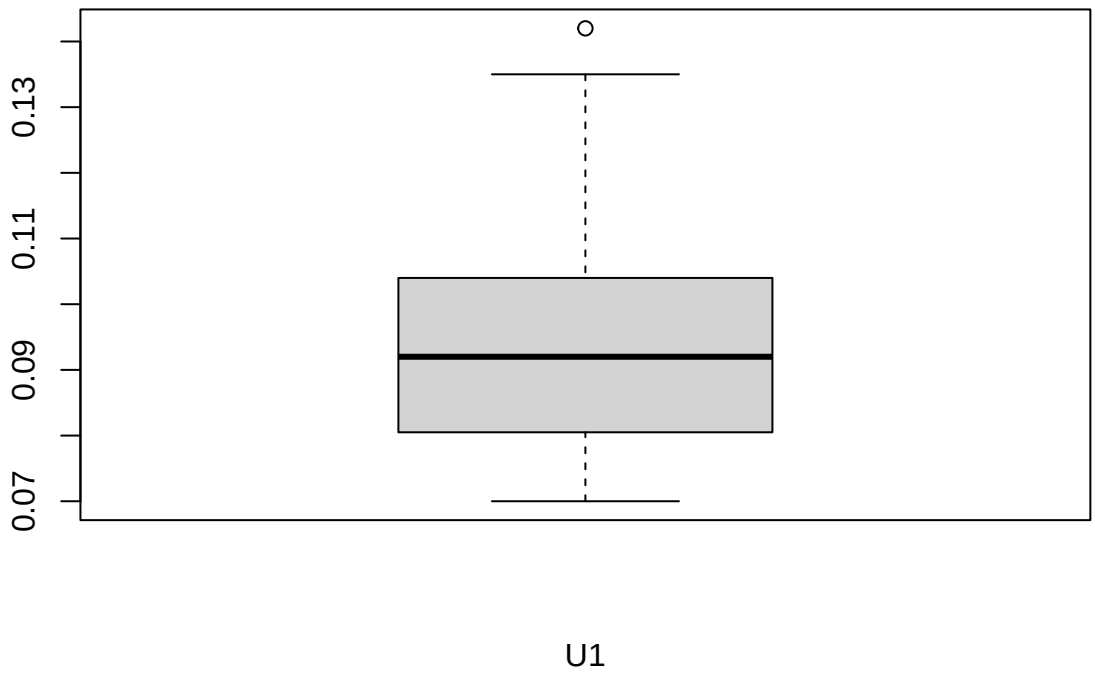
```
boxplot(crime$Pop, xlab = 'Pop')
```



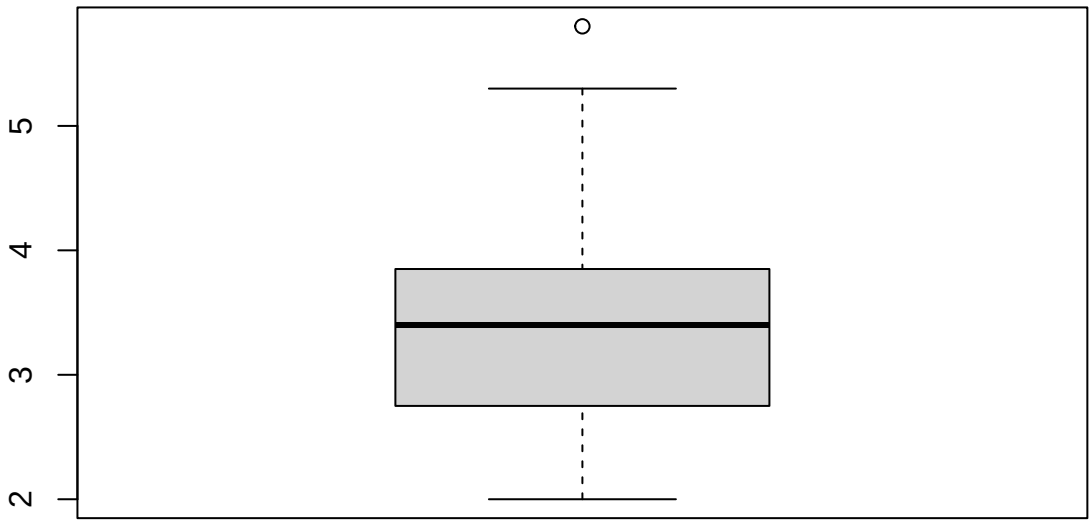

```
boxplot(crime$NW, xlab = 'NW')
```



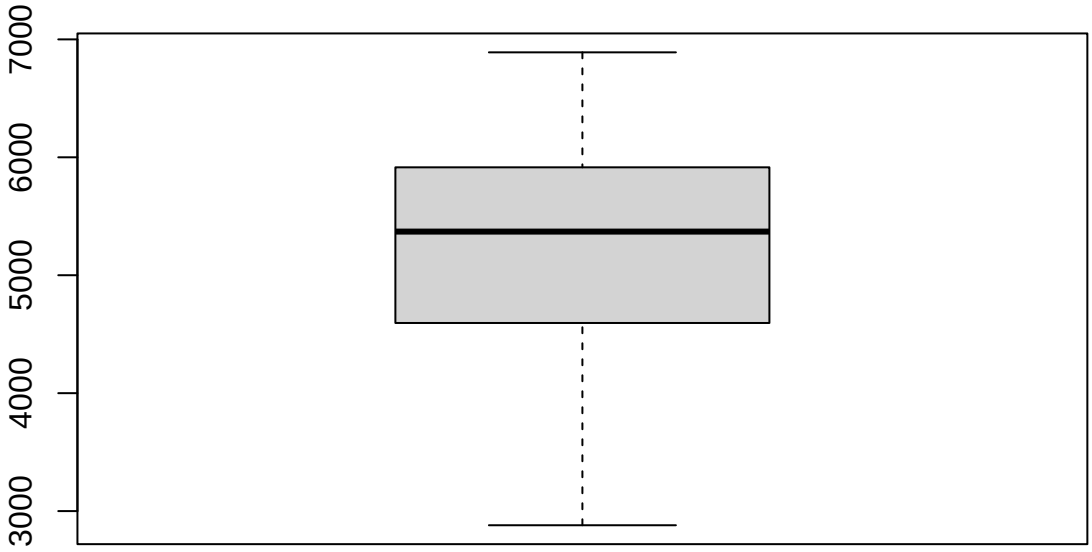

```
boxplot(crime$U1, xlab = 'U1')
```



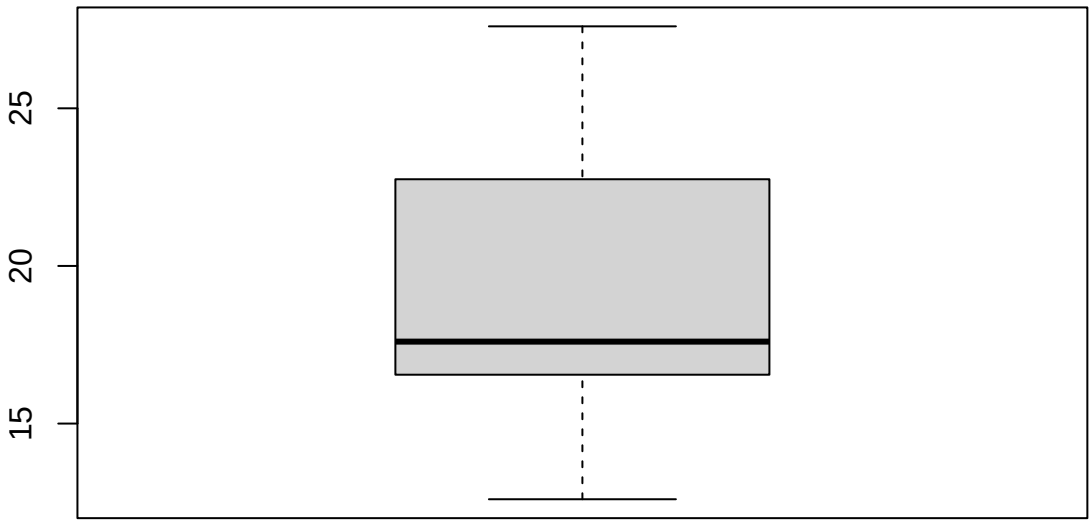

```
boxplot(crime$U2, xlab = 'U2')
```



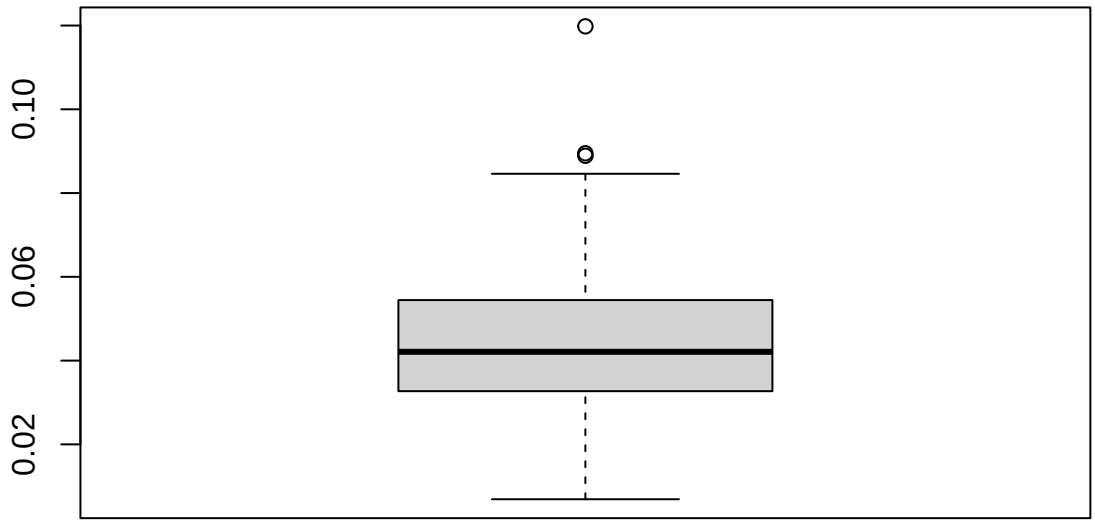

```
boxplot(crime$Wealth, xlab = 'Wealth')
```



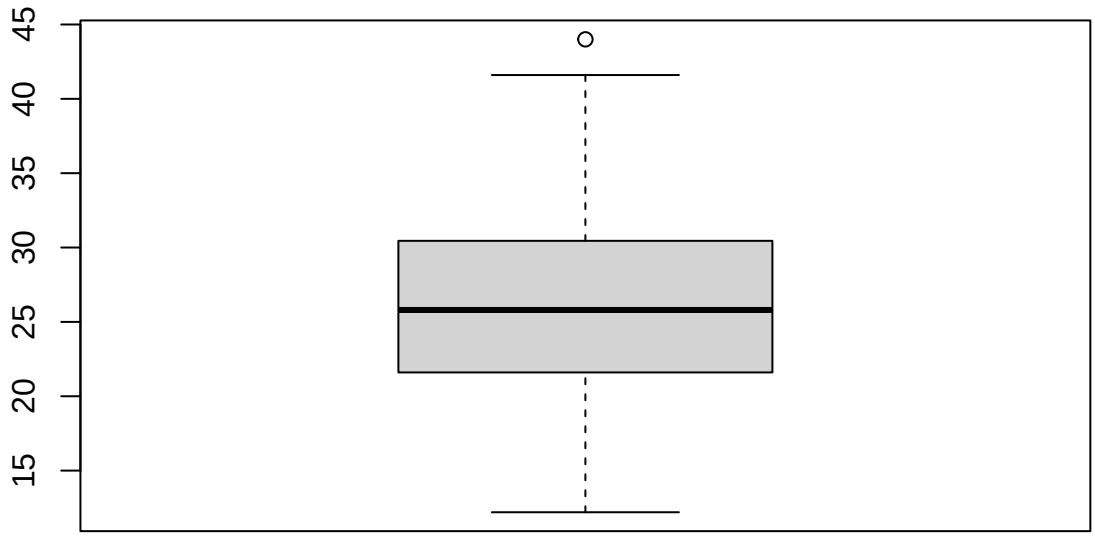

```
boxplot(crime$Ineq, xlab = 'Ineq')
```



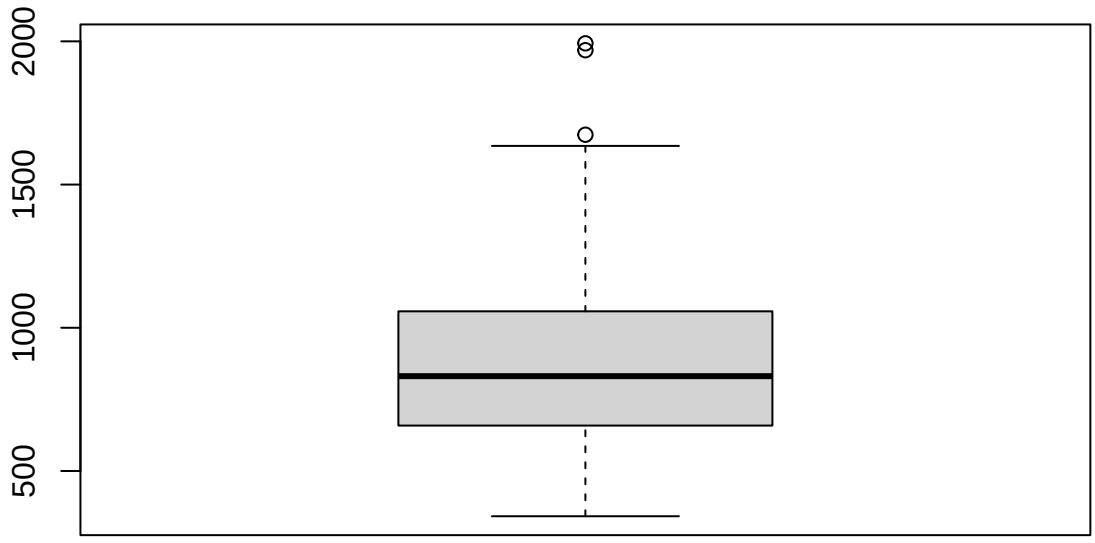

```
boxplot(crime$Prob, xlab = 'Prob')
```




```
boxplot(crime$Time, xlab = 'Time')
```




```
boxplot(crime$Crime, xlab = 'Crime')
```



Crime

To create a linear regression model to predict crime based on a certain series of predictors, the data was split 70-30 into training and test sets. A linear regression model would initially be fitted using all of the data in the data set. After performing hypothesis testing, the best model would then be evaluated on the test data, and if necessary, the model would be further modified in order to improve its performance on real-world data. then, once the model was refined, its performance would be evaluated on the training set, consisting of data the model has not seen before.

```
#split data into training and test using a 70-30 split
random_split <- sample(1:nrow(crime), as.integer(0.7 * nrow(crime)))
crime_train <- crime[random_split, ]

crime_test <- crime[-random_split, ]
```

The first linear regression model to predict crime was created with all of the variables from the original data set. The model had an r-squared value of 0.8032, but a significantly lower adjusted r-squared value of 0.7078, which indicates that many of the predictors so nor seem relevant for predicting crime. The F test for overall regression resulted in a large F-statistic and resulting p-value near zero, indicating that the overall model is significant for predicting crime. However, many of the individual predictors themselves are not significant and the standard error is also large. these signs indicate that this model may be over-fitting the data and that there may be a problem with multicollinearity.

Also, only 6 predictors (M, Ed, Po1, U2, Ineq, Prob) out of the 16 including in the model are statistically significant at a level of 0.1. This indicates that a different model should be run with only those predictors included to determine if performance improves.

```r
#create a linear regression model using training data with all of the predictors
model1 <- lm(Crime ~ . , data = crime)
summary(model1)
```

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

Even though the model above was over-fit, the crime rate was predicted for a city with 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, and Time = 39.0. The predicted crime rate was 155, which seems low given the values of each predictor. this provides further evidence that a new model should be tested with only those predictors significant at a level of 0.05.

```r
#make prediction with coefficient values included in the assignment on the original model
prediction_point <-data.frame(M = 14.0,So = 0, Ed = 10.0, Po1 = 12.0, Po2 = 15.5,LF = 0.640, M.F = 94.0

predictions1 <- predict(model1, prediction_point)
predictions1
```

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

After seeing how over-fit the original model was, I tried a model with only predictors significant at a level of 0.05, but the resulted r-squared, adjusted r-squared, and residual standard error got much worse. For the sake of space, the code and output for that model are not included here.

As a result, I changed the model around to include variables that were significant at the level of 0.1. The resulting r-squared and adjusted r-squared values were slightly lower than those from the full model (0.7659 and 0.7307 vs. 0.8031 and 0.7078). However, there was not as large of a difference between r-squared and adjusted r-squared. Also, the residual standard error decreased slightly and each predictor was significant at a level of 0.05.

As a result, this will be the model that will be used to check assumptions and test on new data.

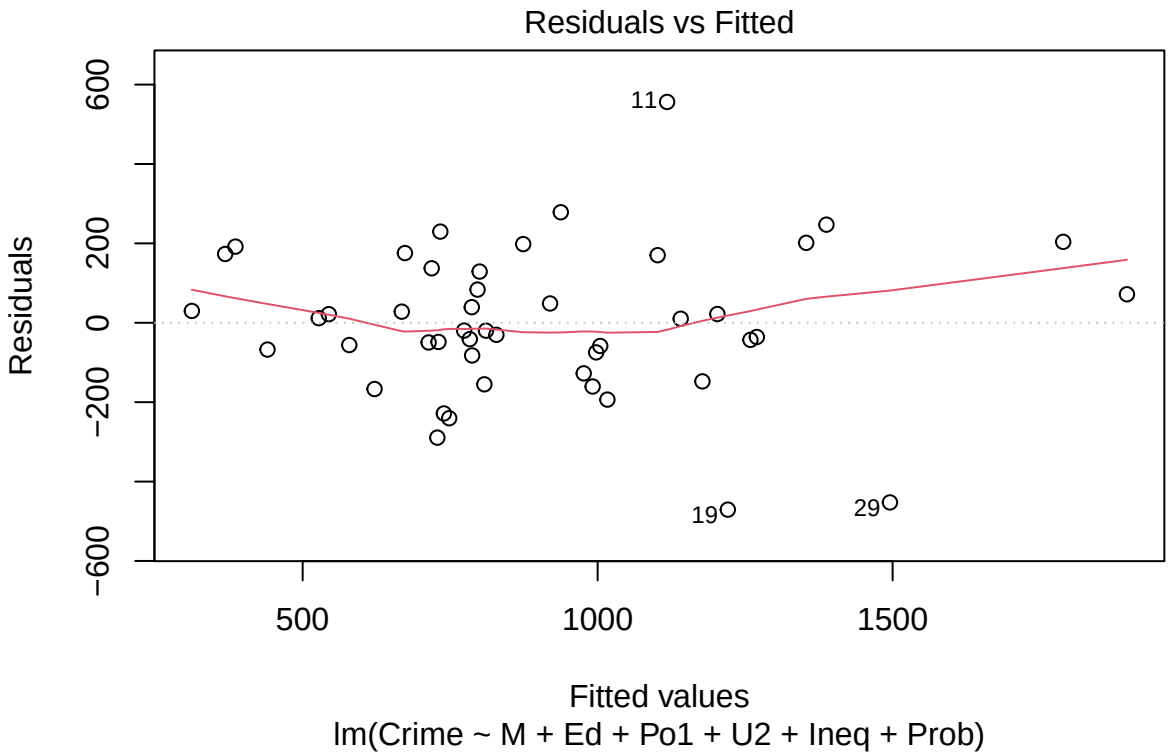
```
#create a new model with only those predictors that were significant at the 0.1 level
model2 <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime)
summary(model2)
```

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

Before running this model on test data, the four assumptions for regression of zero mean, uncorrelated errors, constant variance, and normality needed to be checked.

The residuals vs fitted values plot was obtained in order to check whether the assumptions of equal variance, uncorrelated errors, and zero mean were met. According to the plot, the residuals mostly seem to be scattered along the zero line, with no evidence of a funnel-like shape. The mean of the residuals also seems to mostly hover around zero, with the exception of the upper tail which seems to only be influenced by a couple of data points. The residuals are not grouped in clusters either. As a result, the assumptions of uncorrelated errors, zero mean, and constant variance are met.

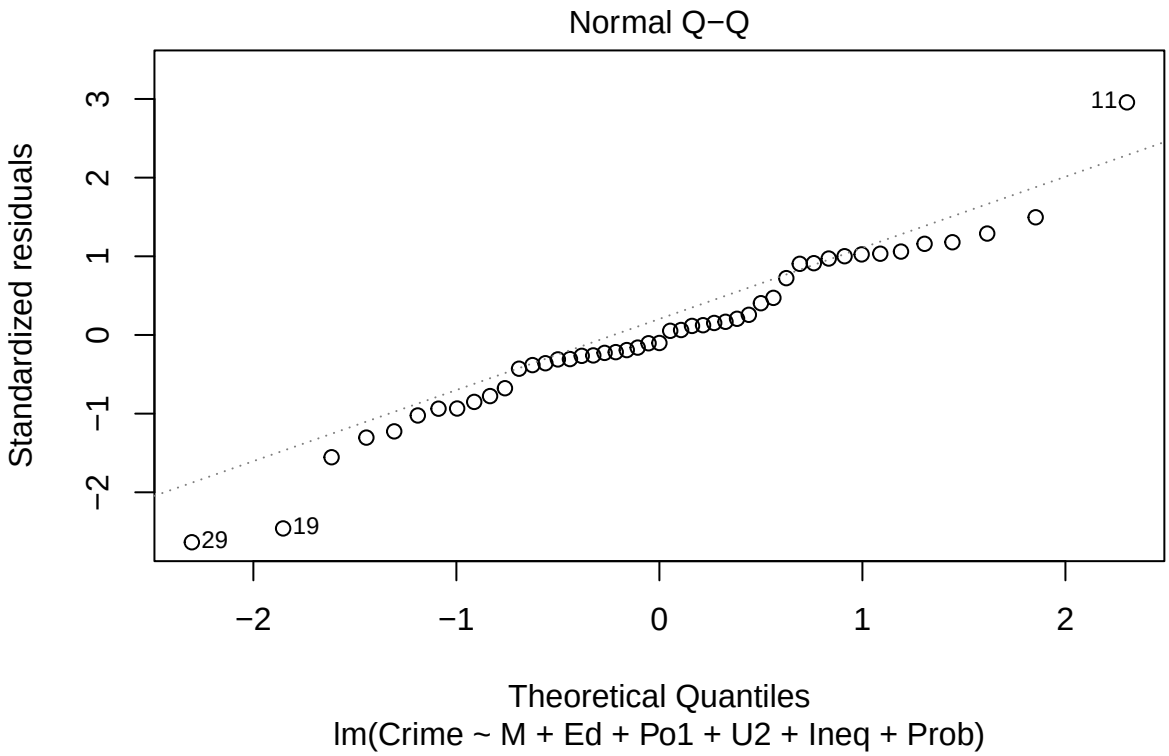
```
#Check linear regression assumptions of zero mean, uncorrelated errors, normality, and equal variance
plot(model2, 1)
```



The qq-plot was created in order to check whether the assumption of normality was met. Here, we want the residuals to follow the straight line without deviating too far off at either tail. From this plot, the residuals mostly follow the straight line with the exception of observations 11, 19, and 29 which are potential outliers or influential points. Since the residuals mostly followed the straight line, the assumption of normality is met as well.

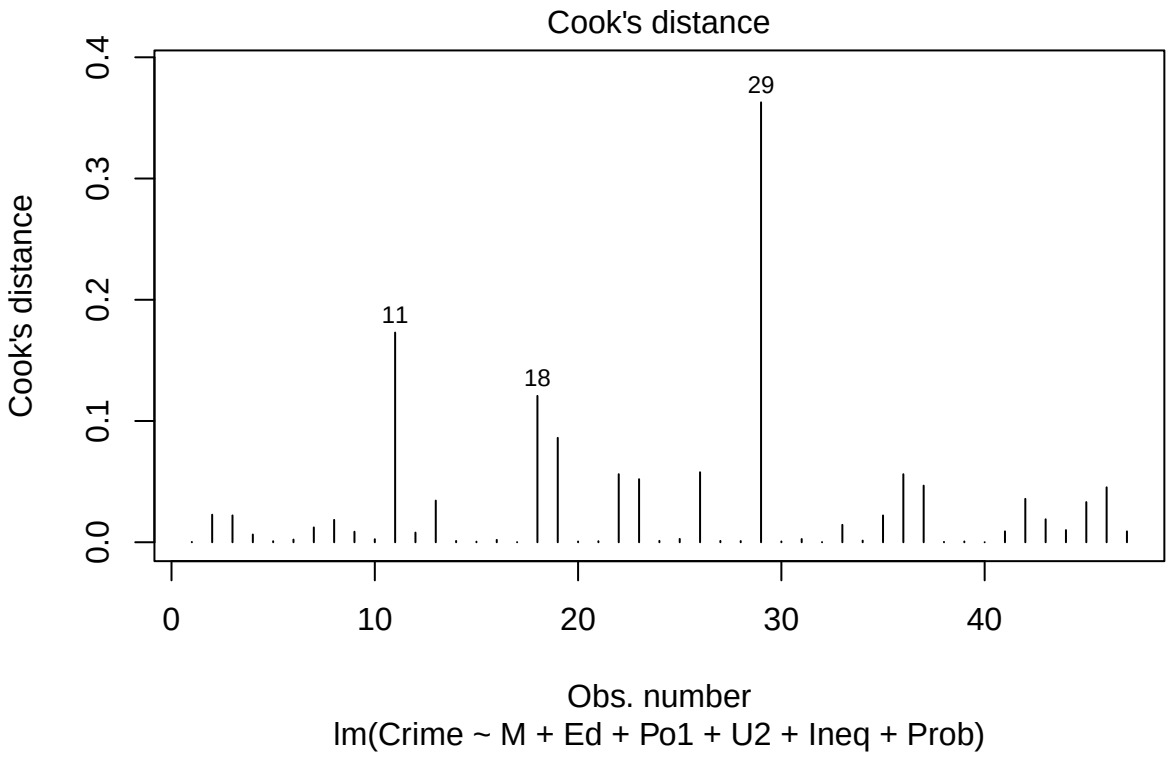
Since each of the four assumptions were met, this model will be further refined on the training data.

```
#obtain qq plot to check the assumption of normality
plot(model2, 2)
```



A plot showing the Cook's distance, which is a measure of how much the model changes when a certain observation is removed from the data set, was obtained in order to discern for influential data points. Based on the plot, there are two observations, 11 and 29, which are flagged fo having much larger Cook's distance than the other observations. If I were going to present this model to an employer, these two observations would be investigated further and the model would be run with each outlier removed in order to compare model performance. For purposes of this assignment, though, I kept the outliers in the model and ran the model on training data below.

```
plot(model2, 4)
```



The model with M, Ed, Po1, U2, Ineq, and Prob as predictors was evaluated on training data in order to see if any more adjustments needed to be made. Based on the output below, each of the model predictors was significant at a level 0.1, an improvement from the full model. The overall regression is still statistically significant, based on the results of the F-test for overall regression. The r-squared value of 0.7393 and adjusted r-squared of 0.6768 were lower than when the model was tested on the complete data set, but not enough to be a major concern. This is a strong sign that the model is not over-fitting the data as much, which is much better for testing this model on real-world data.

This model performance was encouraging, so it seemed ready to evaluate its performance on the test data, data the refined model has not seen before.

```
#use the chosen model on training data
model_train <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_train)
summary(model_train)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -490.23  -95.56    8.54   98.10  558.31
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5118.14    1233.03  -4.151 0.000336 ***
## M             104.83      48.32   2.170 0.039750 *
```

```
## Ed            211.56      62.05   3.410 0.002214 **
## Po1           113.33      17.37   6.524 7.82e-07 ***
## U2             81.98      62.79   1.306 0.203605
## Ineq           65.86      18.78   3.507 0.001733 **
## Prob        -4032.60    1884.38  -2.140 0.042299 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 219.8 on 25 degrees of freedom
## Multiple R-squared:  0.7792, Adjusted R-squared:  0.7262
## F-statistic: 14.71 on 6 and 25 DF,  p-value: 3.906e-07
```

Below, the model with M, Ed, Po1, U2, Ineq, and Prob as predictors was evaluated on the test data. However, the problems that were evident in the full model showed up again in the test data. The overall regression was significant, but only 2 of the 6 predictors were statistically significant at a level of 0.1. The standard error increased to 244, much higher than the error on the full model. The difference between r-squared and adjusted r-squared increased (0.8352 and 0.7116) indicate the model is over-fitting the data and has potential problem with multicollinearity. As a result of this performance, this model does not seem to generalize well to real-world data, and I would recommend going back to the drawing board and either gather additional data, use a different modeling approach, or discard the model altogether.

```
#use the chosen model on test data
model_test <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_test)
summary(model_test)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -221.91  -80.46  -12.90  107.20  254.08
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4219.73    2001.28  -2.109   0.0680 .
## M             103.61      54.67   1.895   0.0947 .
## Ed            136.27      91.50   1.489   0.1747
## Po1            88.61      42.86   2.067   0.0726 .
## U2            147.48      66.30   2.225   0.0568 .
## Ineq           67.26      26.66   2.523   0.0356 *
## Prob        -6051.00    4533.25  -1.335   0.2187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 193.1 on 8 degrees of freedom
## Multiple R-squared:  0.7629, Adjusted R-squared:  0.5851
## F-statistic: 4.291 on 6 and 8 DF,  p-value: 0.03119
```

Even though the model performance did not meet expectations, the observed crime rate was predicted for a city with 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, and Time = 39.0. The model predicted a crime rate of 1330, which seems to make more sense based on the coefficient values, but since

the model over-fit the data and showed some signs of multicollinearity, this result is likely not very accurate at all.

```
#obtain the final prediction using the model evaluated on the test data
final_prediction <- predict(model_test, prediction_point)
final_prediction
```

```
##        1
## 1297.538
```