Tree-Based Regression Models

This situation required regression models be fit on the *uscrime* data set to model a response in crime per capita against a number of predictor variables. Prior to fitting any models, however, the nature of the data needed to be examined.

Data Pre-Processing

Normality

First, the distribution of each variable needed to be examined. For linear regression, each variable following a normal distribution is an important assumption for fitting a model. To examine the normality of each variable, the Shapiro-Wilk Test for normality was used.

```
norm.df = data.frame()
for(i in 1:16){
  norm.df[i,'col'] = colnames(crime[i])
  norm.df[i,'normality p-value'] = round(shapiro.test(crime[[i]])$p.value,4)
}
norm.df
```

```
##       col normality p-value
## 1       M            0.0361
## 2      So            0.0000
## 3      Ed            0.0032
## 4     Po1            0.0043
## 5     Po2            0.0066
## 6      LF            0.1720
## 7     M.F            0.0038
## 8     Pop            0.0000
## 9      NW            0.0000
## 10     U1            0.0086
## 11     U2            0.2133
## 12 Wealth            0.3375
## 13   Ineq            0.0132
## 14   Prob            0.0179
## 15   Time            0.6132
## 16  Crime            0.0019
```

From this, it can be seen that the variables LF, Time, Wealth, and U2 all have p-values below the threshold of 0.05, showing that they are not normally distributed. When examining the residual diagnostics of the linear regression model fit, if the residuals are not normally distributed then any of these variables used will be standardized to a normal distribution to help remedy this problem.

Outliers

Next, the data set needed to be examined for outliers. From the results below it can be seen that there are no collective outliers present, another important assumption for fitting a linear regression model.

```
sum(is.na(crime))
```

```
## [1] 0
```

Next, each variable was examined for point outliers. Significant outliers can worsen the fit of a linear regression model, and therefore any predictor variables with point outliers would be examined when fitting a model. To identify possible point outliers, the Grubbs Test was used to find both high and low outliers for each variable in the *uscrime* data set.

```
outlier.df = data.frame()
for(i in 1:16){
  outlier.df[i,'col'] = colnames(crime[i])
  outlier.df[i,'high outlier p-value'] = round(grubbs.test(x = crime[,i],
                                                           type = 10,
                                                           opposite = F,
                                                           two.sided = F)$p.value,4)
  outlier.df[i,'low outlier p-value'] = round(grubbs.test(x = crime[,i],
                                                          type = 10,
                                                          opposite = T,
                                                          two.sided = F)$p.value,4)
}
outlier.df
```

```
##       col high outlier p-value low outlier p-value
## 1       M               0.0303              1.0000
## 2      So               1.0000              1.0000
## 3      Ed               1.0000              1.0000
## 4     Po1               0.1083              1.0000
## 5     Po2               0.1009              1.0000
## 6      LF               0.9608              1.0000
## 7     M.F               0.0405              1.0000
## 8     Pop               0.0051              1.0000
## 9      NW               0.0223              1.0000
## 10     U1               0.1784              1.0000
## 11     U2               0.0702              1.0000
## 12 Wealth               0.2643              1.0000
## 13   Ineq               0.8519              1.0000
## 14   Prob               0.0166              1.0000
## 15   Time               0.2682              0.9063
## 16  Crime               0.0789              1.0000
```

From this, it can be seen that the predictor variables M, M.F, Pop, NW, Prob all contained possible high point outliers, as their p-values for high outliers were below the threshold of

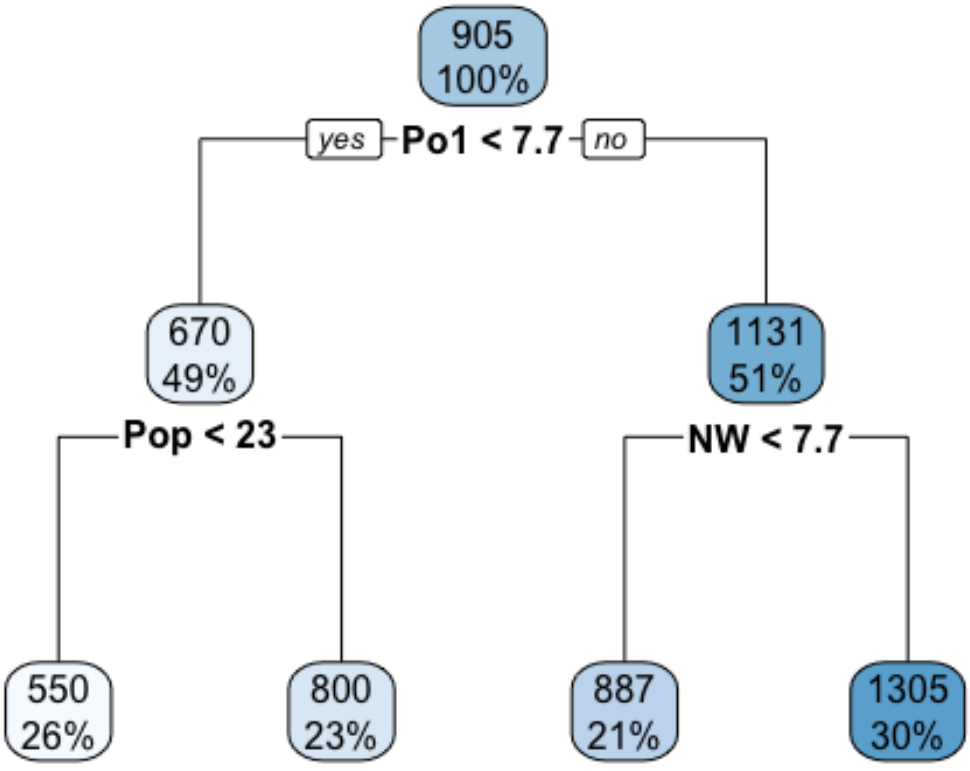
0.05. If any of these variables are selected for the final set of predictors, models with and without the outliers will be fit to examine the effect these point outliers may have.

Regression Tree Model

The first model we will be fitting on the *uscrime* data set is a regression tree model. While previous regression models have been fit on the *uscrime* data set, fitting a regression tree model could have the added benefit of providing more specific regression equations to better explain the nature of the data.

Using the *rpart* function, a regression tree model was fit. This function automates the process splitting the data set, choosing and splitting on factors on half the data, and calculating the estimate error for each leaf on the other half of the data. Initially, no threshold was provide to fit overfitting on the data, but when examining the model created if there are a small number of data points in any leaf (<5%), this will be addressed.

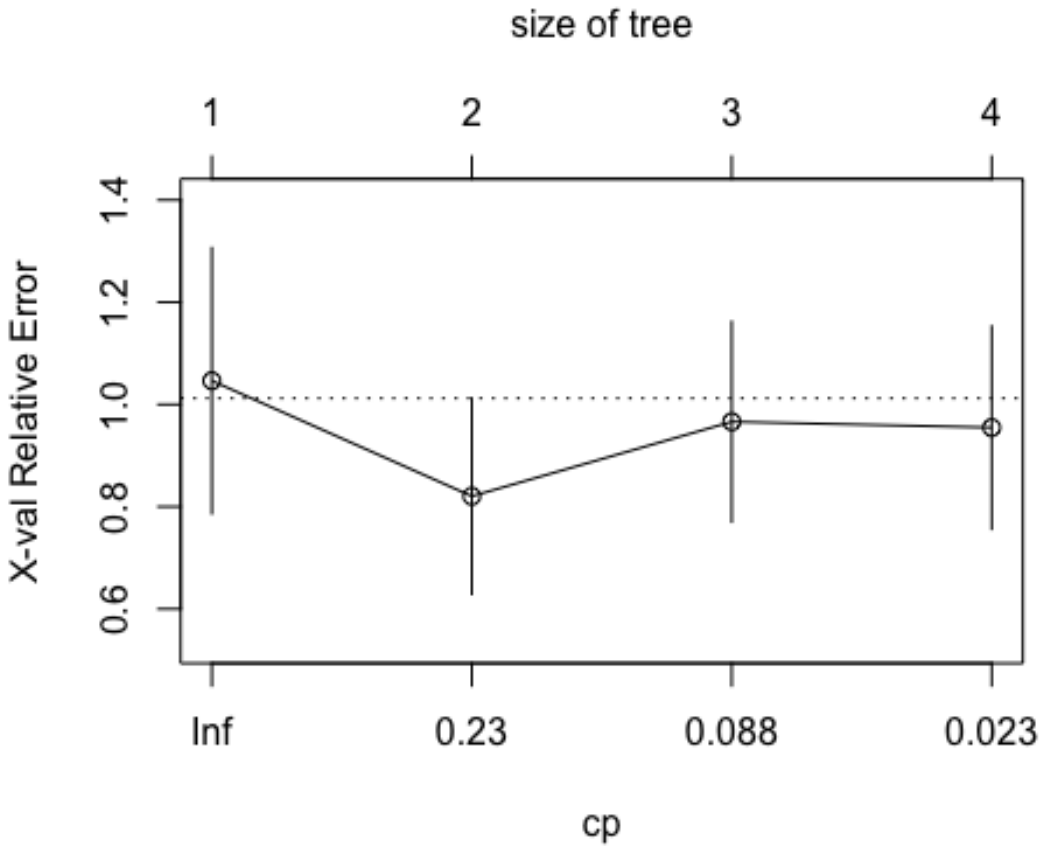
```
rt.1 = rpart(Crime~.,
             data = crime,
             method='anova')
```



From this tree diagram, it can be seen that first the factor *Po1* was split at 7.7, and there was an additional split at *Pop* = 23 for *Po1* < 7.7, and a split at *NW* = 7.7 for *Po1* > 7.7. None

of the final four leafs contain less than 5% of the total data, so overfitting was not an issue with this model.

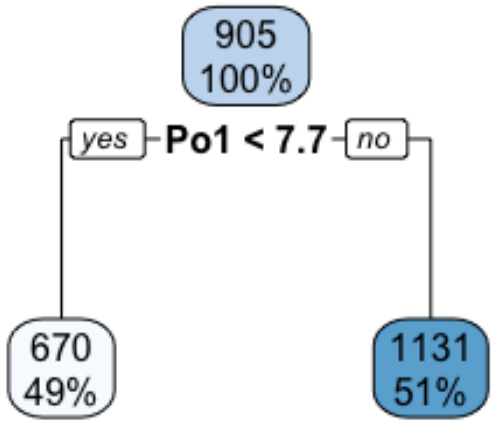
When viewing the cross-validated error summary (seen below), it can be seen that when the size of the tree = 2 (one split, two leaves), the relative error is the lowest.



Because of this, the regression tree model would be pruned to only include the factor split at *Po1* = 7.7.

```
split1.cp = rt.1$cptable[which.min(rt.1$cptable[,"xerror"]),"CP"]
rt.1.prune = prune(rt.1,
                   cp = split1.cp)
```

The final pruned regression tree model can be seen below:



Leaf Regression Models

After determining a good regression tree model for the *uscrime* data set, linear rergession models needed to be fit for each leaf.

To begin, the data was separated at *Po1* = 7.7

```
crime.lesser = crime[crime$Po1 < 7.7,]
crime.greater = crime[crime$Po1 > 7.7,]
```

Then, regression models were fit for both subsets of the data using cross-validation to examine their accuracy

```
train.control = trainControl(method = "cv", number = 10)
# fit initial models
fit.lesser.1 = train(Crime~.,
                     data = crime.lesser,
                     trControl = train.control,
                     method = "lm")
fit.greater.1 = train(Crime~.,
                      data = crime.greater,
                      trControl = train.control,
```

```
                      method = "lm")
# check summaries
summary(fit.lesser.1)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -109.147  -52.803   -6.495   53.784  127.196
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -48.5477  2044.9766  -0.024   0.9817
## M              45.8622    58.6256   0.782   0.4597
## So            380.4815   223.1072   1.705   0.1319
## Ed            187.9074    89.5799   2.098   0.0741 .
## Po1            -3.5138   157.7513  -0.022   0.9829
## Po2            44.6382   148.5528   0.300   0.7725
## LF           1059.3652  1187.9722   0.892   0.4021
## M.F           -22.5521    21.4677  -1.051   0.3284
## Pop            10.6413     5.0929   2.089   0.0750 .
## NW              0.1010     7.9019   0.013   0.9902
## U1           4878.2802  4874.8165   1.001   0.3503
## U2             -5.5126   133.5094  -0.041   0.9682
## Wealth         -0.1022     0.1752  -0.583   0.5779
## Ineq            4.7779    35.5290   0.134   0.8968
## Prob        -7317.4407  3280.7511  -2.230   0.0609 .
## Time          -20.0603     7.7287  -2.596   0.0357 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115.9 on 7 degrees of freedom
## Multiple R-squared:  0.8794, Adjusted R-squared:  0.6209
## F-statistic: 3.403 on 15 and 7 DF,  p-value: 0.0541

summary(fit.greater.1)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -206.805 -120.407   -9.489  103.985  248.226
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8634.1701  2366.4043  -3.649  0.00651 **
```

```
## M              5.6032    96.1623   0.058  0.95496
## So           179.6267   409.5210   0.439  0.67254
## Ed           263.0845   146.4229   1.797  0.11010
## Po1          235.2349   166.1289   1.416  0.19452
## Po2         -140.7023   193.8759  -0.726  0.48869
## LF          1442.4214  4832.4463   0.298  0.77294
## M.F           -1.2379    54.8160  -0.023  0.98254
## Pop           -3.7686     2.8833  -1.307  0.22751
## NW            -0.5396    24.5039  -0.022  0.98297
## U1         -3779.9843 10923.3434  -0.346  0.73823
## U2           163.7106   150.5361   1.088  0.30848
## Wealth         0.3017     0.2051   1.471  0.17946
## Ineq         155.3754    65.5077   2.372  0.04511 *
## Prob       -3624.0711  4381.4724  -0.827  0.43214
## Time          21.9335    14.6754   1.495  0.17338
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 229.9 on 8 degrees of freedom
## Multiple R-squared:  0.8827, Adjusted R-squared:  0.6626
## F-statistic: 4.012 on 15 and 8 DF,  p-value: 0.02669
```

From the summaries above, it can be seen that there are a number of insignificant predictors in both regression models. Next, these predictors were removed and models were refit.

```
fit.lesser.2 = train(Crime ~ Ed + Pop + Prob + Time,
      data = crime.lesser,
      trControl = train.control,
      method = "lm")
fit.greater.2 = train(Crime ~ Ineq,
                      data = crime.greater,
                      trControl = train.control,
                      method = "lm")
```

The accuracy of the full and reduced models for each subset of the data were examined next. From this, it can be seen that the cross-validated RMSE of the reduced model for both regression models was lower. Because of this, the reduced models would be retained for further analysis.

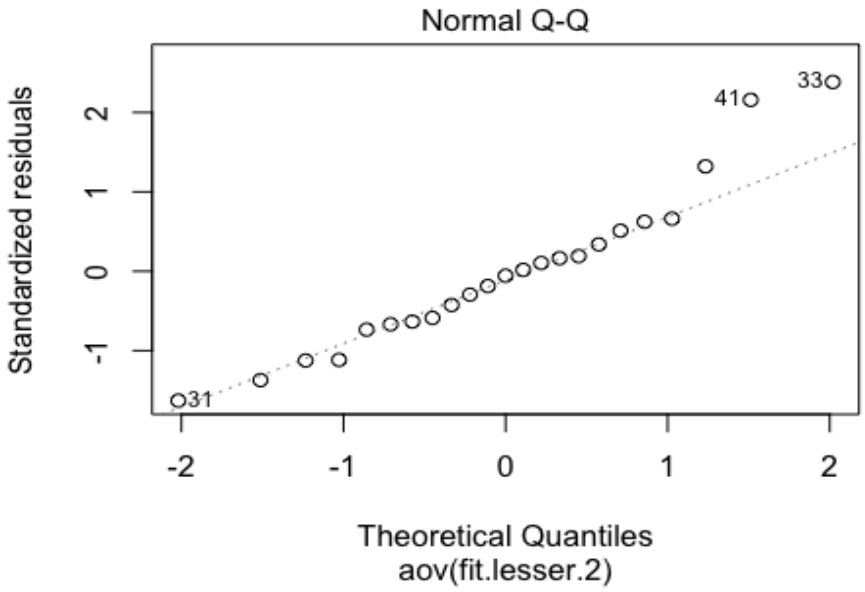
```
fit.lesser.1$results$RMSE

## [1] 290.2356

fit.lesser.2$results$RMSE

## [1] 158.4142

fit.lesser.1$results$Rsquared

## [1] 0.8863891
```

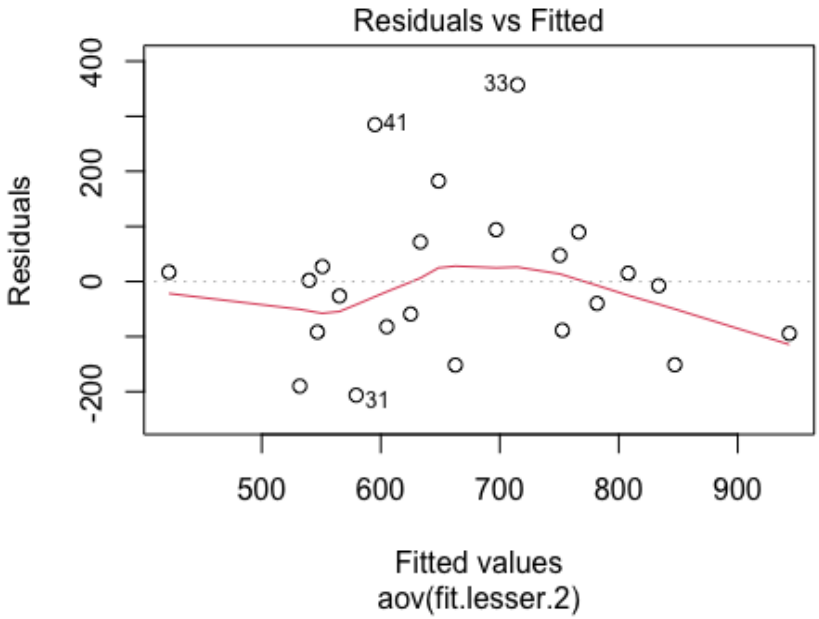
```
fit.lesser.2$results$Rsquared

## [1] 0.7117982

fit.greater.1$results$RMSE

## [1] 452.463

fit.greater.2$results$RMSE

## [1] 352.8472

fit.greater.1$results$Rsquared

## [1] 0.8441368

fit.greater.2$results$Rsquared

## [1] 0.8747934
```

Residual Diagnostics

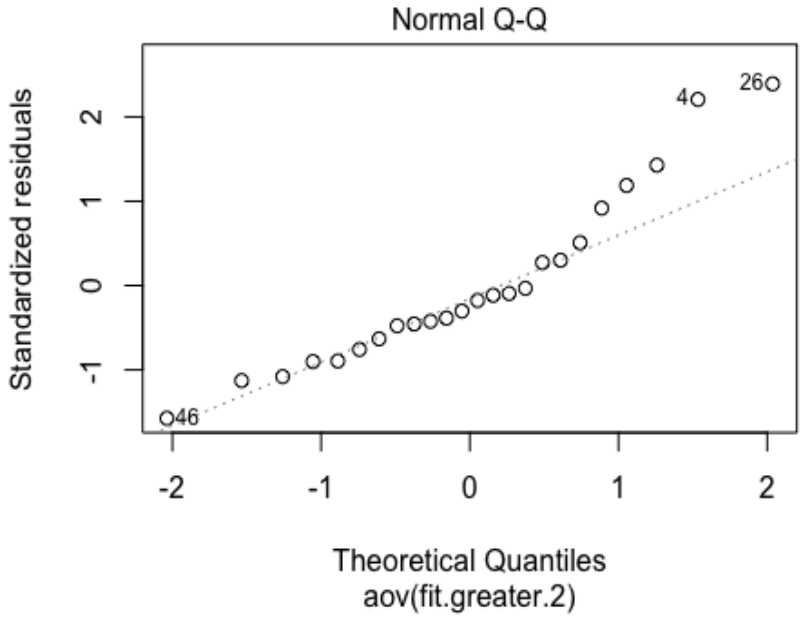
After determining a good regression equation for both subsets of the data, the residuals associated with each model needed to be examined to ensure regression assumptions were met.

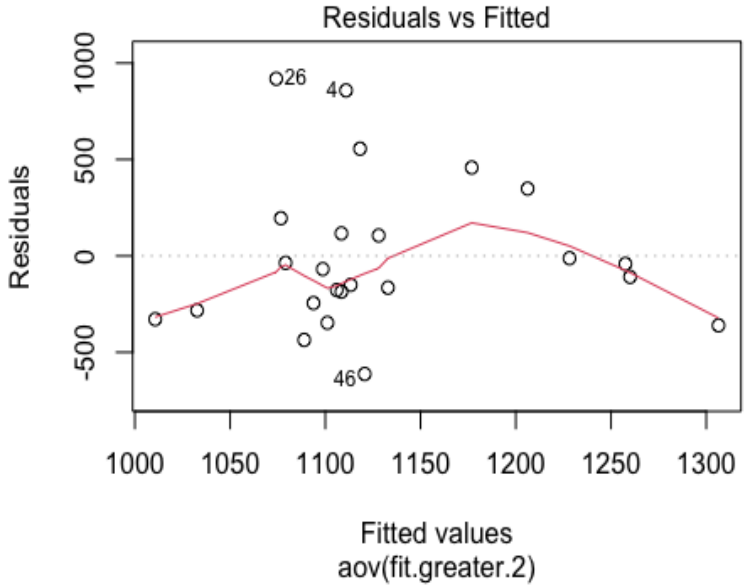
From the plots below, it can be seen that the residuals associated with the model when *Po1* < 7.7 were relatively normally distributed and with normal noise, albeit with a few outliers.





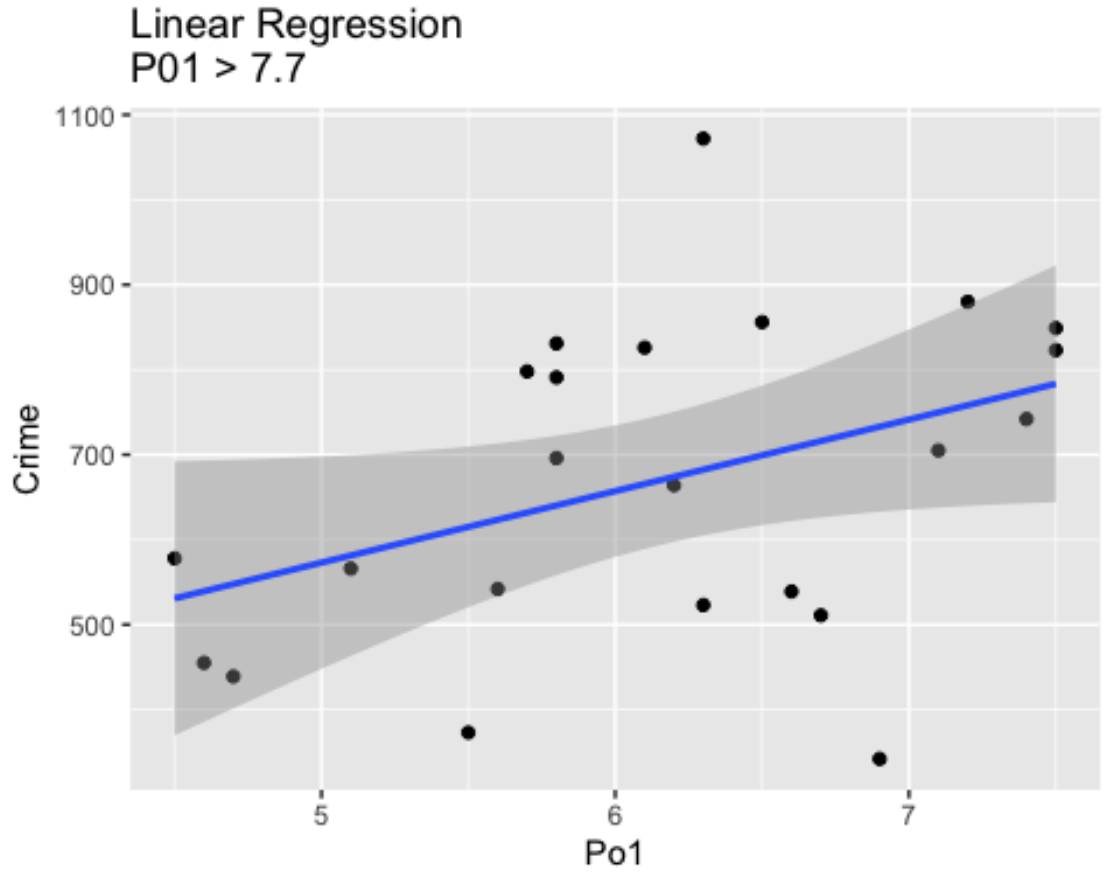
Similar conclusions can be drawn about the model when *Po1* > 7.7. Showing that the assumptions were met for both of these models.


Normal Q-Q
Standardized residuals
Theoretical Quantiles
aov(fit.greater.2)
26
4
46


Residuals vs Fitted
Residuals
Fitted values
aov(fit.greater.2)
26
4
46

Interpretation

The fit and summary for the model when *Po1* < 7.7 can be seen below:

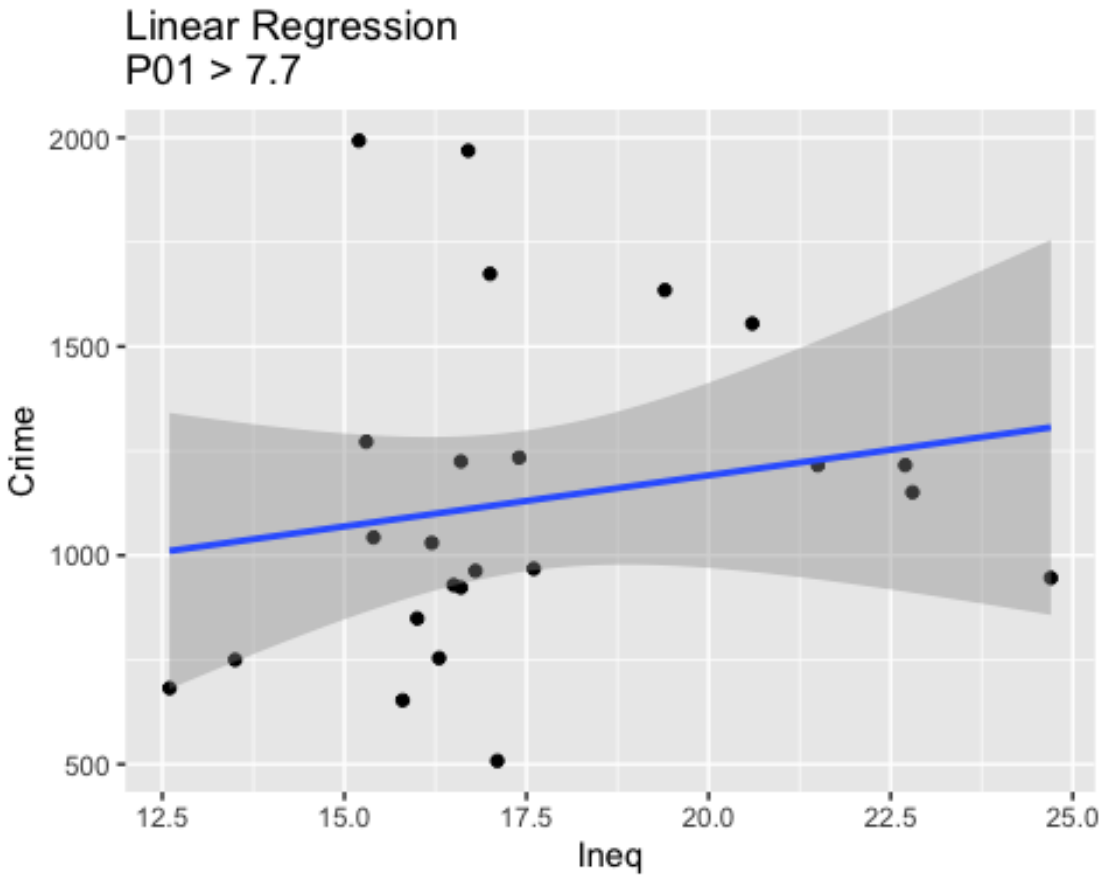



```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -206.35  -90.22   -7.59   59.64  357.11
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   819.960    515.112   1.592   0.1288
## Ed              9.499     34.869   0.272   0.7884
## Pop            11.395      3.229   3.529   0.0024 **
## Prob        -3164.075   2095.755  -1.510   0.1485
## Time          -12.130      6.830  -1.776   0.0927 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 154.5 on 18 degrees of freedom
```

```
## Multiple R-squared:  0.4485, Adjusted R-squared:  0.3259
## F-statistic: 3.659 on 4 and 18 DF,  p-value: 0.02379
```

Notably, when the per capita expenditure on police protection in 1960 was less than 7.7, when the state population increased by 11.395 (in hundred thousands) or the months served by offenders in state prisons before their release decreased by 12.130, the number of offenses in 1960 (per 100,000 people) increased by 1. These were the two most significant predictors of crime rate when *Po1* was less than 7.7, and this model accounted for 71.18% of the variance in the data, as seen by the cross-validated regression summary previously.

The fit and summary for the model when *Po1* > 7.7 can be seen below:




```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -612.67 -254.25  -88.83  136.25  918.77
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   702.70     495.52   1.418    0.170
## Ineq           24.44      27.91   0.876    0.391
##
## Residual standard error: 397.9 on 22 degrees of freedom
## Multiple R-squared:  0.03368,    Adjusted R-squared:  -0.01024
## F-statistic: 0.7668 on 1 and 22 DF,  p-value: 0.3907
```

Notably, when the per capita expenditure on police protection in 1960 was greater than 7.7, when the percentage of families earning below half the median income increase by 24.44%, the number of offenses in 1960 (per 100,000 people) increased by 1 as well. This was the most significant predictor of crime rate when *Po1* was greater than 7.7, and this model accounted for 87.48% of the variance in the data, as seen by the cross-validated regression summary previously.

Random Forest Model

The next model that was fit on the *uscrime* data set was a random forest model. In comparison to the regression tree model, the random forest approach has the benefit of possibly providing better estimates by averaging multiple trees and also avoiding overfitting, albeit at the cost of less interpretative results.

Model Fitting

The random forest model fit can be seen below:

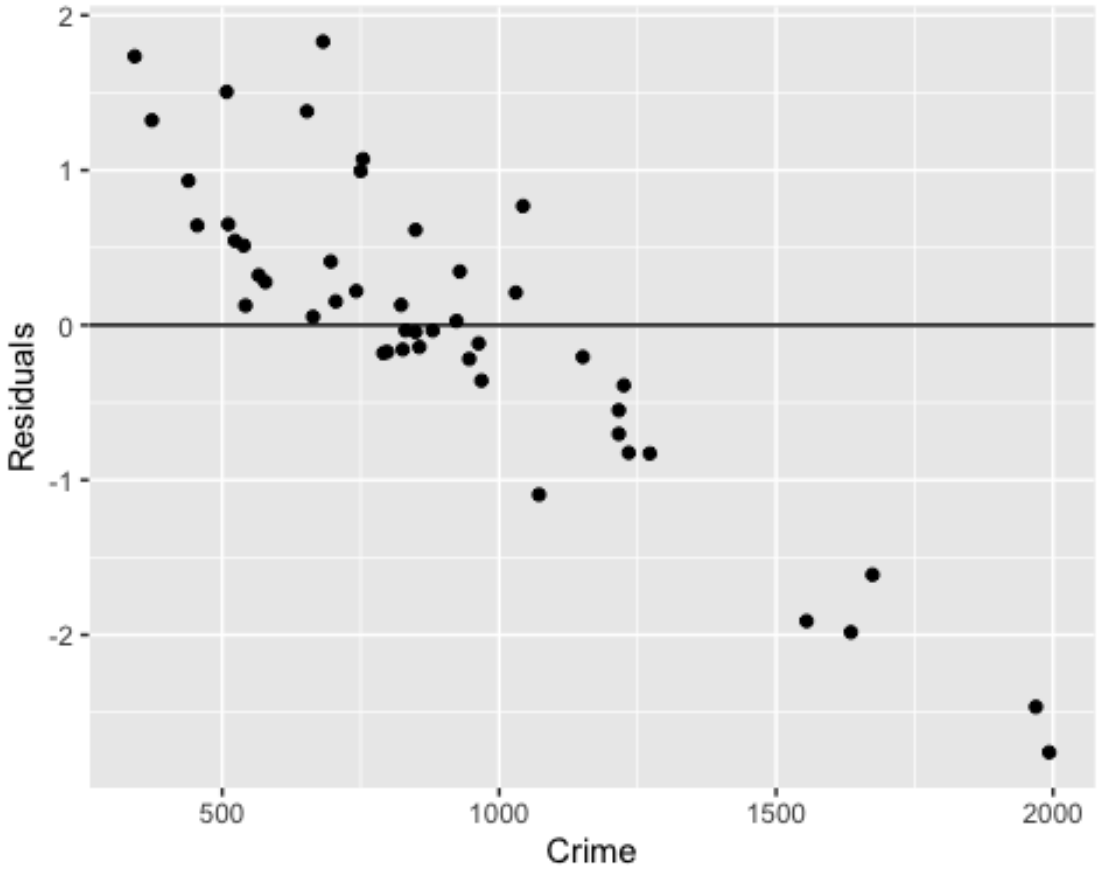
```
rf.1 = train(Crime~.,
             data=crime,
             trControl = train.control,
             method = "rf")
rf.1
```

```
## Random Forest
##
## 47 samples
## 15 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 43, 42, 43, 42, 42, 44, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##    2    279.0416  0.7399678  215.6868
##    8    286.8866  0.6663538  215.0176
##   15    301.4528  0.6375353  224.5733
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 2.
```

From this, it can be seen that the best model fit was when the algorithm randomly selected 2 variables to choose where to split. The cross-validated RMSE associated with this model was 279.04, and the model accounted for 74.00% of the variance in the data.

Residual Diagnostics

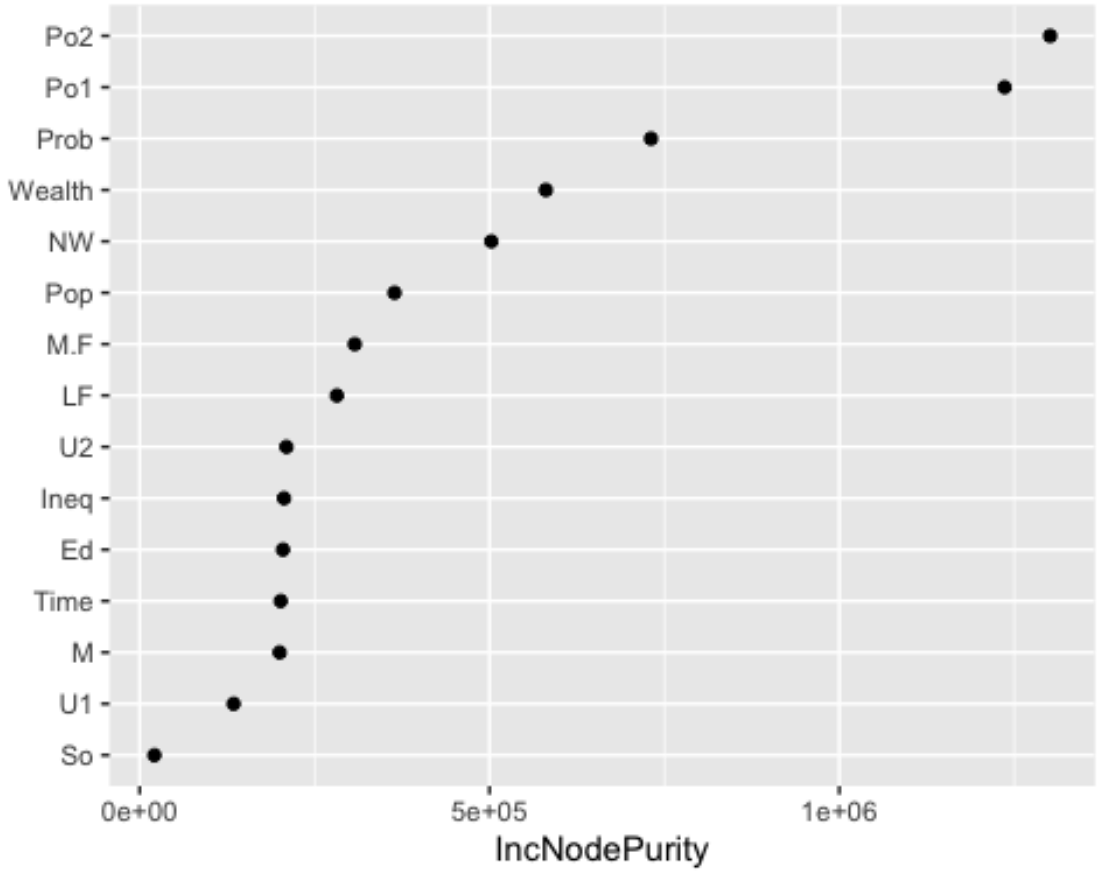
The residuals associated with the random forest model can be seen below:



Notably, there is a downward trend in the residuals showing an issue with the normality or variance in the data. This was not addressed as part of this analysis, but would need to be if this model would be of further use.

Interpretation

To interpret the results of the random forest model fit, the importance of each predictor variable can be seen below:



From this, it can be seen that *Po1* and *Po2* were the most important predictors of the response variable for this random forest model. Additionally, the model accounted for 74.00% of the variance in the data, as seen previously. When compared to the regression tree model fit on the data previously, it appears that the random forest model was less accurate than the best regression tree models as it had a higher cross-validated RMSE and lower R-squared value. This could be due to the number of leafs used in random forest model causing overfitting (the regression tree model only had one split), or it could be because more insignificant predictors were used in the random forest model. Predictors could be removed from the random forest model, but this would involve the implementation of some threshold for when to use or reject a predictor variable.