Final Project Report

Introduction

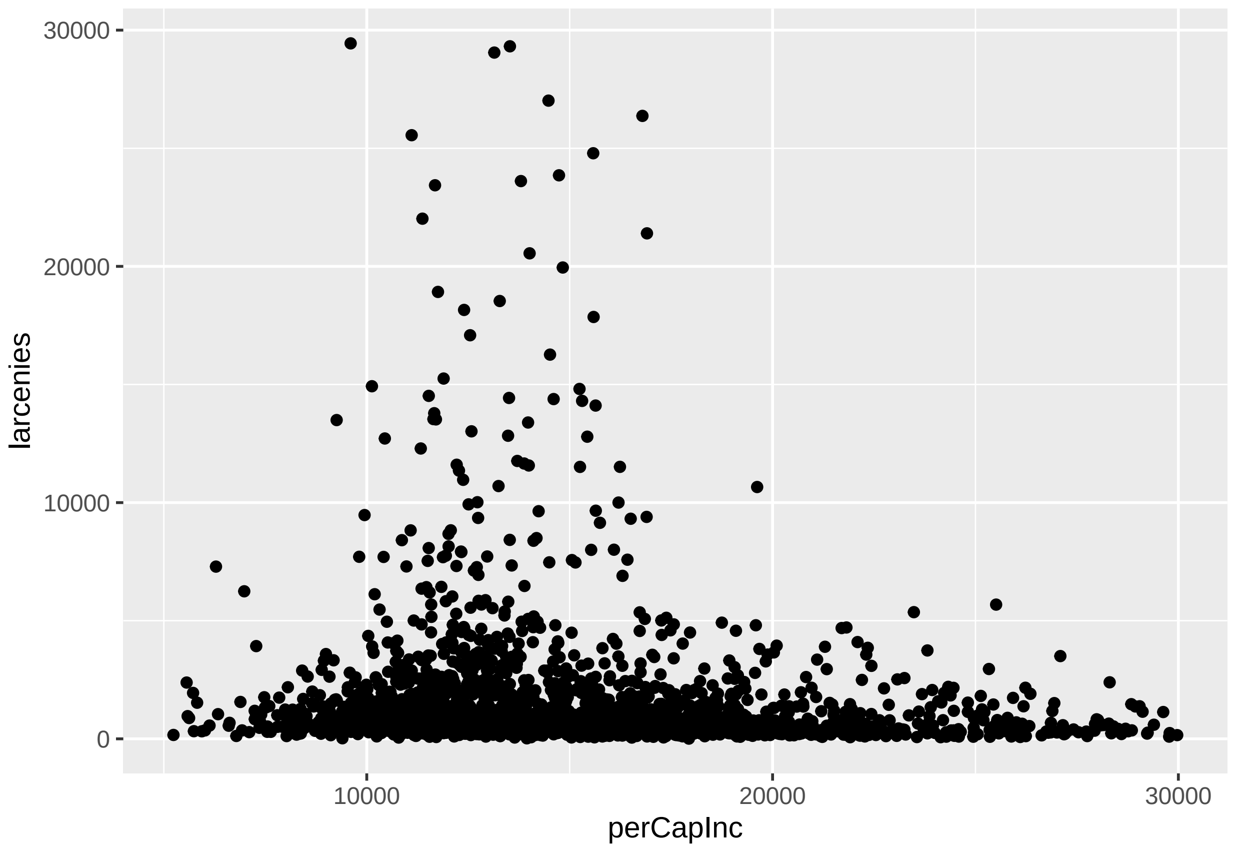
The purpose of this assignment is to figure out if we can predict the number of larcenies with various independent variables. The data set consists of information from the 1990 census, law enforcement data, and the 1995 Uniform Crime Reports FBI team. There ar 125 independent variables that are used to determine the 18 dependent variables in the data set. This data was concatenated from all those 3 sources specifically to perform linear regression. One piece of information that I found interesting was that this data was collected from police stations with at least 100 officers and a few random smaller police departments sprinkled in. That could cause some sort of bias because larger departments are usually in relatively major cities, which takes away smaller towns that are usually safer and could lower the overall crime numbers.

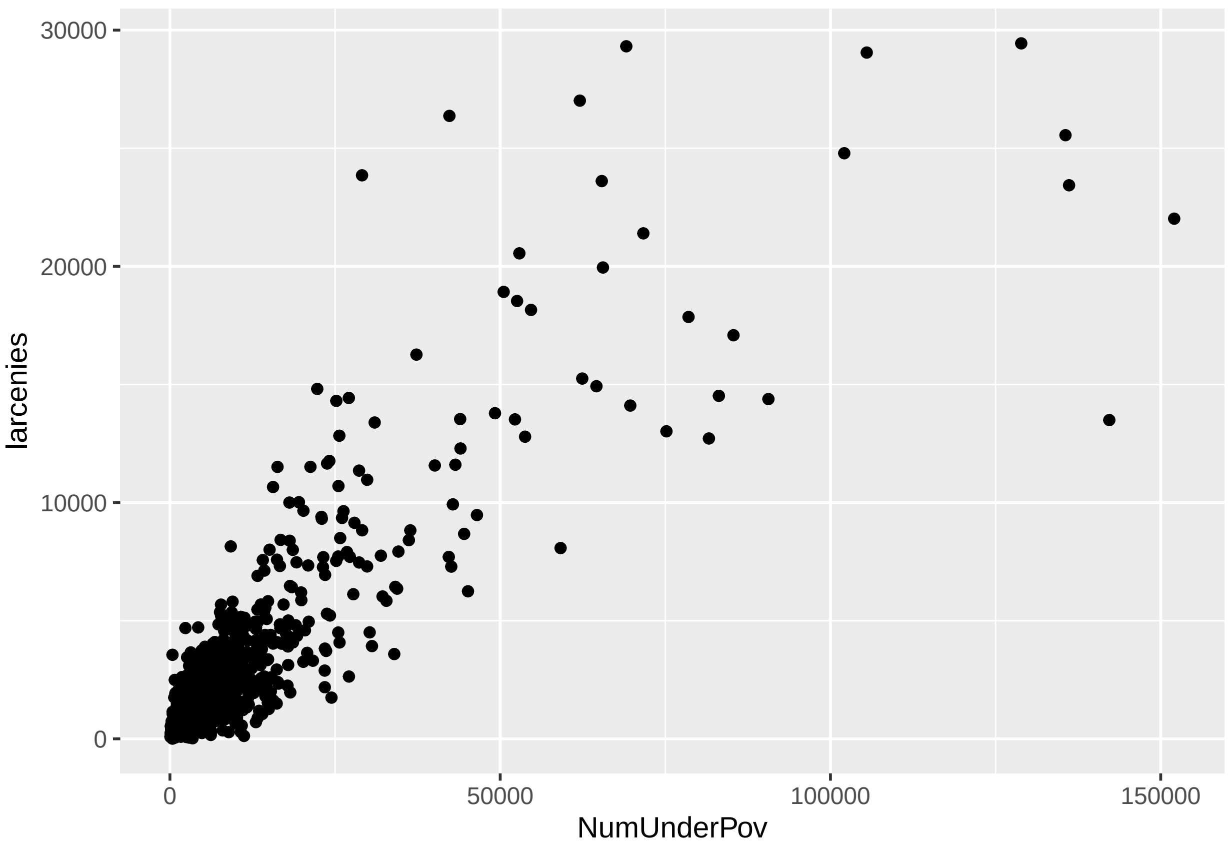
Step 1

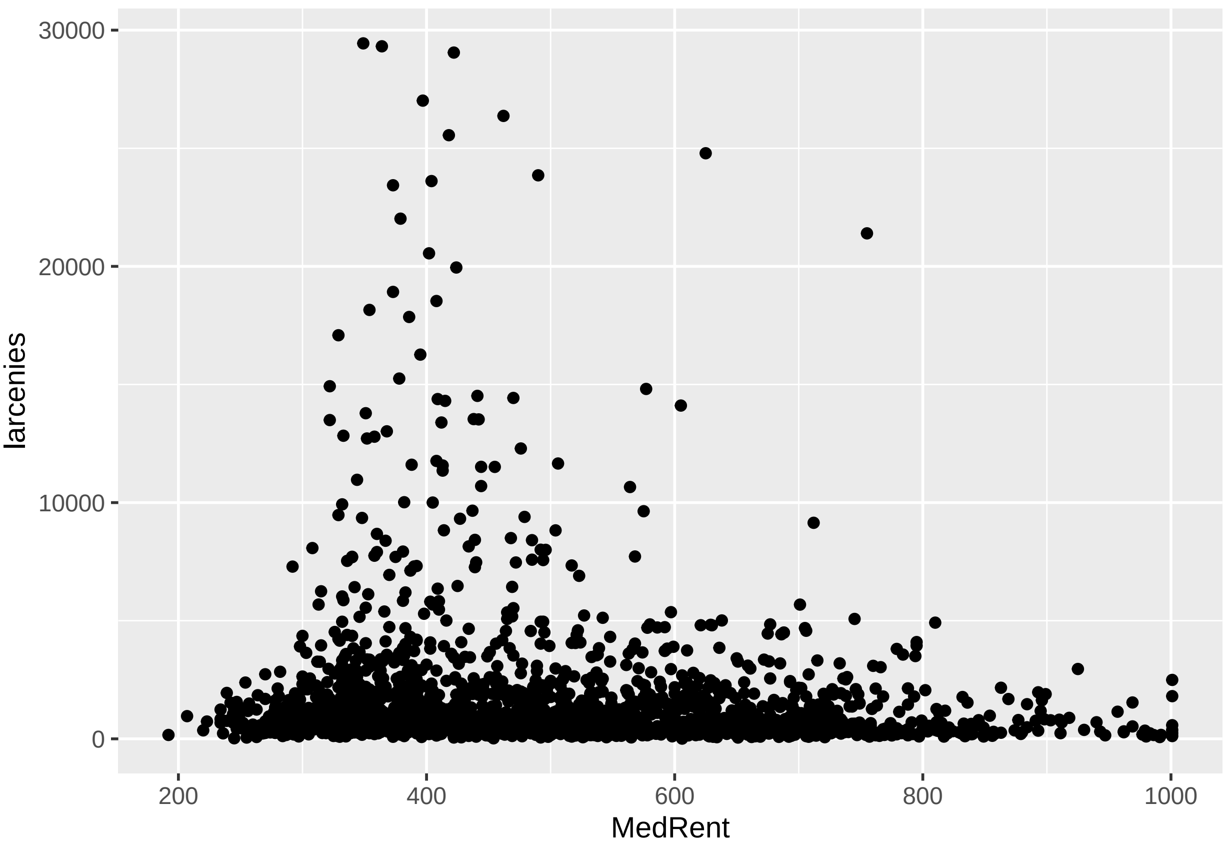
In the first step, before I chose the variables, I got rid of the data points that had missing values and I filtered the data to take out some of the extreme values. This made the data patterns more visible when graphed, especially to identify heteroscedasticity and correlation. The way I decided to filter the data was just trying to look at where the majority of data was, which was a thick cluster of points on the left side of the graph. The variables that I thought would we good predictors based on my intuition were per capita income (perCapInc), the number of homeless people counted in the street (NumStreet), the median gross rent (MedRent), and the number of people under the poverty level.

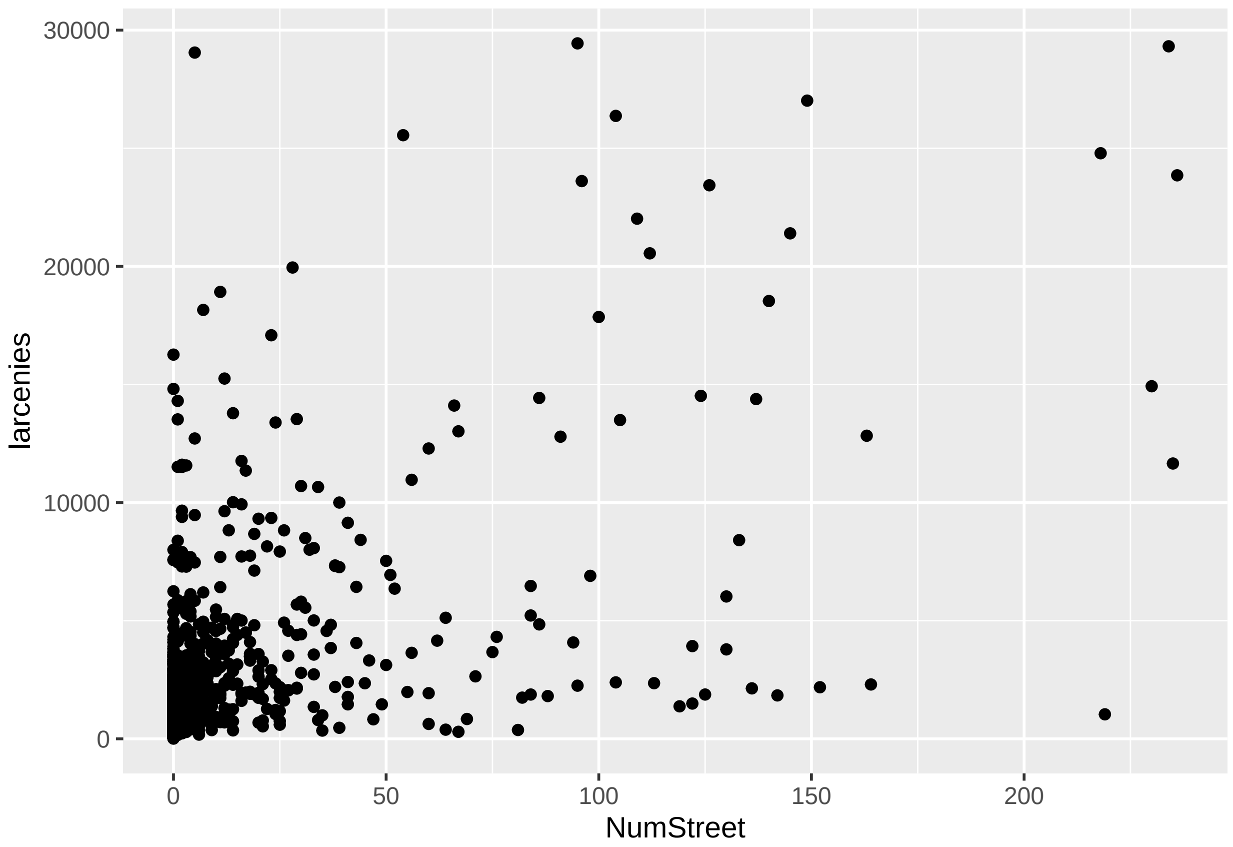
```r
data <- read.delim('CommViolPredUnnormalizedData.txt', header = TRUE, sep = ',')
names <- read.csv('FormattedColumnNames.csv', header = FALSE)
colnames(data) <- names$V1
select_data <- dplyr::select(data, larcenies, perCapInc, NumStreet, MedRent, NumUnderPov)
select_data <- select_data[!select_data$larcenies == '?',]
select_data <- select_data[!select_data$perCapInc == '?',]
select_data <- select_data[!select_data$NumStreet == '?',]
select_data <- select_data[!select_data$MedRent == '?',]
select_data <- select_data[!select_data$NumUnderPov == '?',]
select_data <- na.omit(select_data)
select_data$larcenies <- as.numeric(select_data$larcenies)
select_data <- select_data %>% filter(select_data$larcenies <= 30000)
select_data <- select_data %>% filter(select_data$NumStreet <= 250)
select_data <- select_data %>% filter(select_data$perCapInc <= 30000)
```

```r
ggplot(select_data, aes(y=larcenies, x= perCapInc), fig(7,5)) + geom_point()
ggplot(select_data, aes(y=larcenies, x= NumUnderPov), fig(7,5)) + geom_point()
ggplot(select_data, aes(y=larcenies, x= MedRent), fig(7,5)) + geom_point()
ggplot(select_data, aes(y=larcenies, x= NumStreet), fig(7,5)) + geom_point()
```


30000
20000
10000
0
larcenies
10000
20000
30000
perCapInc


30000
20000
10000
0
larcenies
0
50000
100000
150000
NumUnderPov


30000
20000
10000
0
larcenies
200
400
600
800
1000
MedRent



The first scatterplot shows that there is little to no correlation between larcenies and Per Capita income. The second scatterplot shows that there is little to no correlation between larcenies and the number of homeless people in the street. The third scatterplot shows that there is little to no correlation between larcenies and the median gross rent.' The last scatterplot shows that relationship between larcenies and the number of people under the poverty level is positive and has a strong correlation.

Step 2

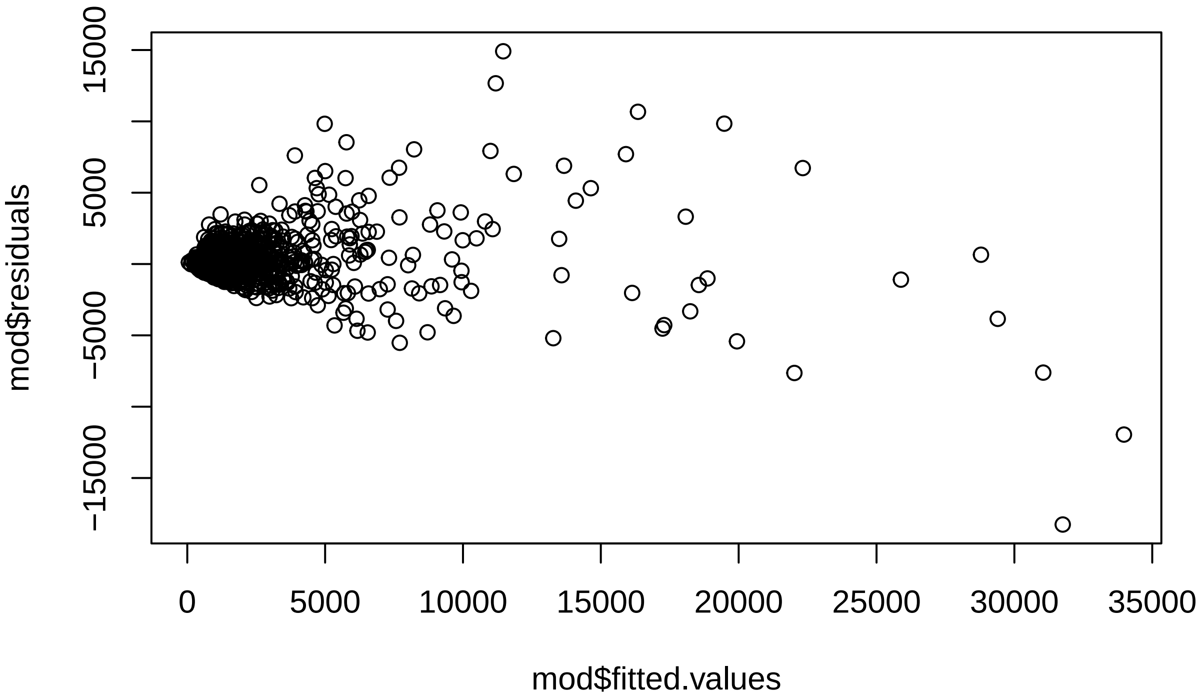
After the data set was finalized, I fit a model with those four variables predicting the number of larcenies and looked at the summary to see how well the variables I selected did to predict.

```
mod <- lm(larcenies~perCapInc+NumStreet+MedRent+NumUnderPov, data = select_data)
summary(mod)
```

```
##
## Call:
## lm(formula = larcenies ~ perCapInc + NumStreet + MedRent + NumUnderPov,
##     data = select_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18260.5   -422.9   -165.7    198.1  14910.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.127e+02  1.073e+02  -1.981 0.047683 *
## perCapInc    7.630e-02  9.973e-03   7.651 3.02e-14 ***
## NumStreet    2.012e+01  1.625e+00  12.386  < 2e-16 ***
## MedRent     -1.021e+00  2.846e-01  -3.588 0.000341 ***
## NumUnderPov  2.073e-01  3.108e-03  66.695  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1319 on 2104 degrees of freedom
## Multiple R-squared:  0.7927, Adjusted R-squared:  0.7923
## F-statistic:  2012 on 4 and 2104 DF,  p-value: < 2.2e-16
```

- The statistically significant predictors are perCapInc, NumStreet, NumUnderPov, and MedRent.
- The R-squared value is 0.7927, which means that these 4 variables predict about 80% of the data.
- I would assume since the all variables had financial implications that the financial status and the crime rate of an area are correlated.

```
plot(mod$fitted.values, mod$residuals)
```



I plotted the diagnostic plot to test for heteroscedasticity, also known as non-constant variance, and I got a cone shaped graph. That told me that I have to adjust my model in some way to make the model more reliable.

Step 3

In order to try and remedy the model I performed two separate kinds of model selection processes to help me determine which variables. These models test various versions of the model and provide the variables that should be kept.

fastbw() method

```
mod2 <- ols(mod, data = select_data)
fastbw(mod2, rule = 'p', sls = 0.05)
```

```
##
## No Factors Deleted
##
## Factors in Final Model
##
## [1] perCapInc   NumStreet   MedRent     NumUnderPov
```

stepAIC() method

```
stepAIC(mod)
```

```
## Start:  AIC=30310.95
## larcenies ~ perCapInc + NumStreet + MedRent + NumUnderPov
##
##               Df  Sum of Sq        RSS   AIC
## <none>                       3.6625e+09 30311
## - MedRent      1   22407388 3.6849e+09 30322
## - perCapInc    1  101888516 3.7644e+09 30367
## - NumStreet    1  267033838 3.9296e+09 30457
## - NumUnderPov  1 7743266434 1.1406e+10 32705
```

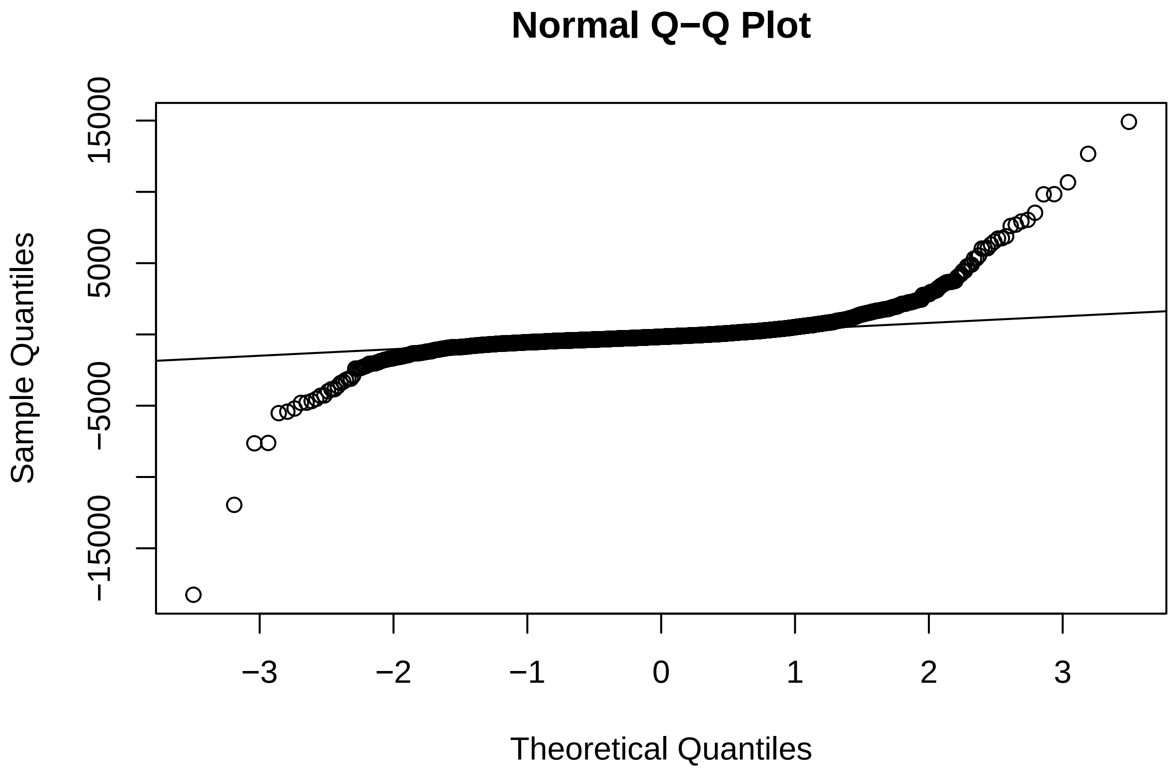
```
##
## Call:
## lm(formula = larcenies ~ perCapInc + NumStreet + MedRent + NumUnderPov,
##     data = select_data)
##
## Coefficients:
## (Intercept)    perCapInc    NumStreet      MedRent  NumUnderPov
##   -212.6883       0.0763      20.1218      -1.0211       0.2073
```

After both model selection techniques, the results stayed true to the output of the original model summary. However, I still have the issue where the model has inconsistent variance, so something else needs to change.

Step 4

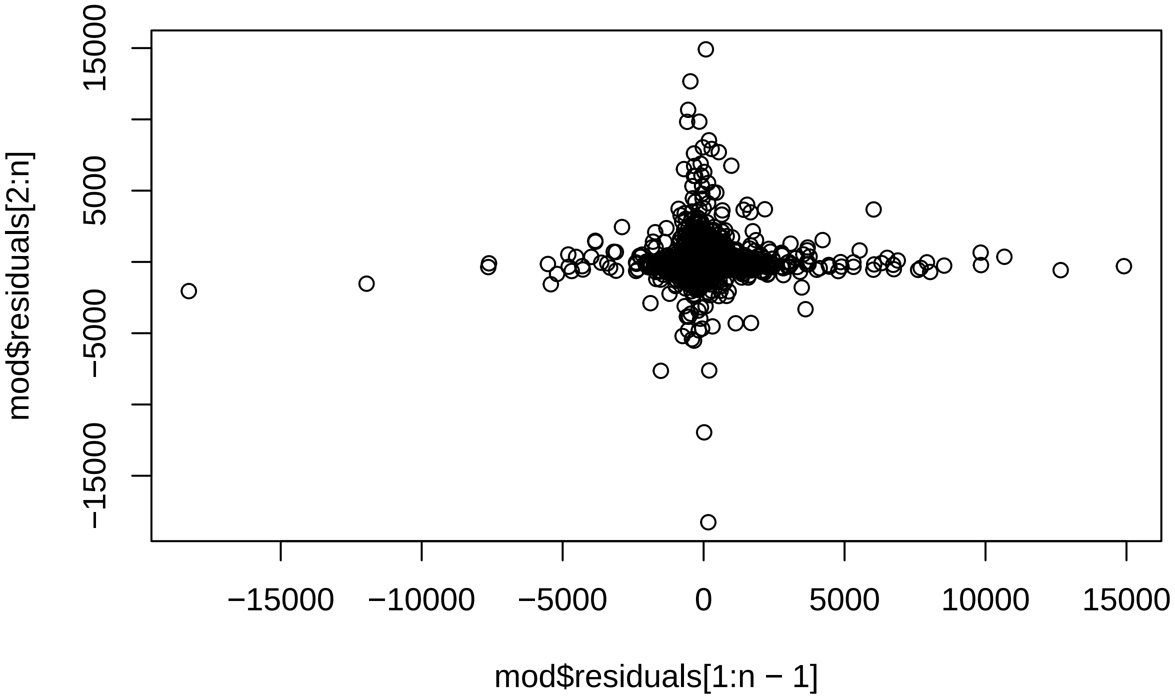
Q-Q plot

```
qqnorm(mod$residuals)
qqline(mod$residuals)
```



Lagged Residuals plot

```
n <- dim(select_data)[1]
plot(mod$residuals[1:n-1],mod$residuals[2:n])
```



To confirm the change from the previous step, I made a couple more graphs, a Q-Q plot and a lagged residual plot. The Q-Q plot, which graphs two different distributions, would produce a straight line if the two distributions are similar. As you can see, the Q-Q plot strays enough from the center line to warrant looking further into the model. The lagged residual plot, which tests for independence from point to point, showed clear randomness, resulting in a circular pattern.

Step 5

```
n <- dim(model.matrix(mod))[1]
p <- dim(model.matrix(mod))[2]
num.df <- p
den.df <- n-p
Fstat <- qf(0.5, num.df, den.df)
Fstat
```

```
## [1] 0.8705716
```

```
cooks_dist <- cooks.distance(mod)
cooks_dist <- as.data.frame(cooks_dist)
cooks_out <- cooks_dist > Fstat
cooks_out <- cooks_out[cooks_out == TRUE]
n_out <- length(cooks_out)
cooks_dist <- cooks.distance(mod)
sort(cooks_dist, decreasing = TRUE)[1:n_out]
```

```
##     1981     747     1087
## 3.270196 1.681316 1.408659
```

Since I wasn't sure if the model needed change, I went back to the data to see if there any more data points that are outliers that I might not have seen before. Using Cook's distance and standardized residuals, I was able to identify a few points that stood out. Cook's distance is a statistic used to measure the influence that an observation can have on a linear regression model. That doesn't necessarily imply negative impact, which is why we also standardized residuals.

```
stand_res <- rstandard(mod)
cooks_out2 <- stand_res > 3
cooks_out2 <- cooks_out2[cooks_out2 == TRUE]
n_out2 <- length(cooks_out2)
sort(stand_res, decreasing = TRUE)[1:n_out2]
```

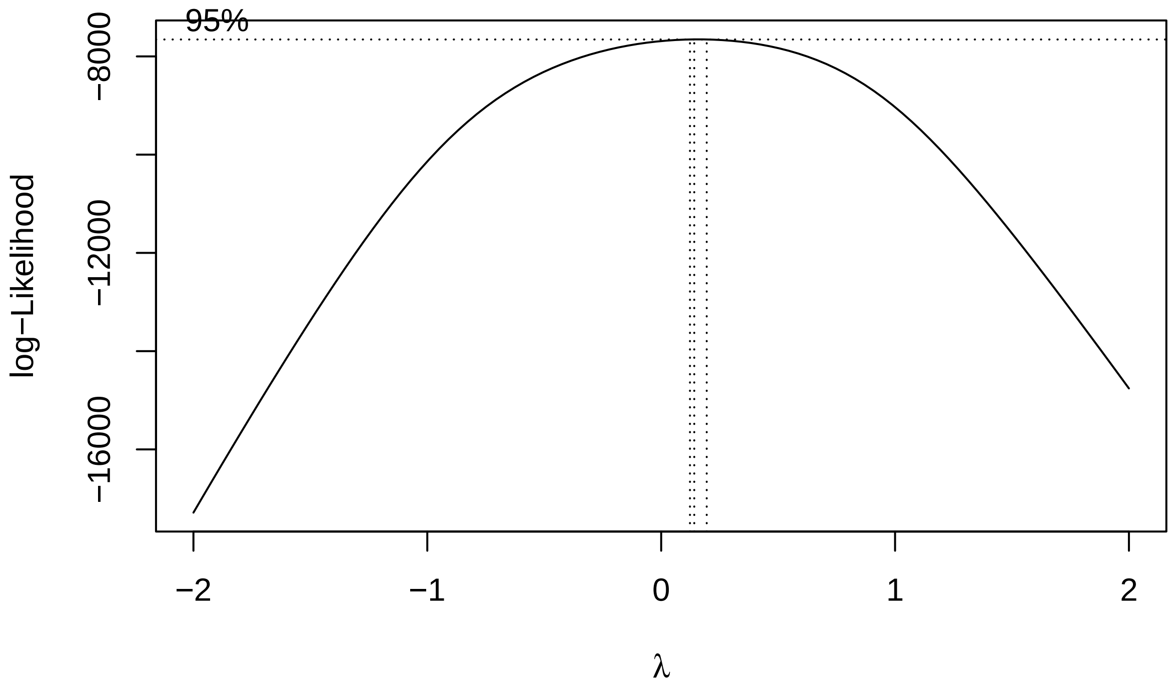
```
##       730      1087        85       196      2002       609       282      1874 
## 11.369219  9.937062  8.185270  7.676728  7.459106  6.477919  6.112707  6.036650 
##       703      1407      1797       772       933        19       554      1861 
##  5.882209  5.771225  5.257789  5.245319  5.132788  4.941926  4.816637  4.593522 
##       947       903       960      1342      1292      1963      1851      1426 
##  4.576330  4.568510  4.196229  4.063424  4.033488  3.712099  3.686140  3.625254 
##       160       108      1318       264       686 
##  3.399382  3.386548  3.200856  3.124787  3.045373
```

Standardized residuals are quantified in standard deviation units, which is why the threshold for a large standardized residual is above 3. In a normally distributed data set, 100% of the data is generally within 3 units of standard deviation, so anything above that is considered outside of the realm of normality. In this data set, I identified 27 data points that were over 3 standard deviation units away from the center of the data set.

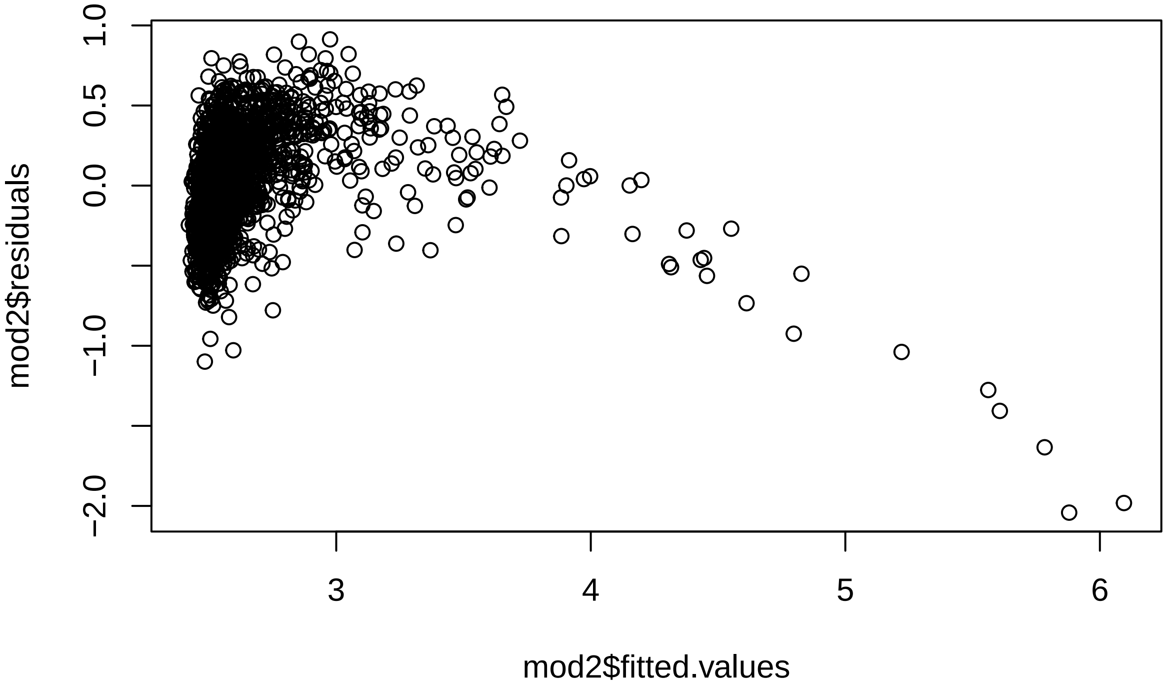
Step 6

My next step is to perform another form of transformation, this time on the response variable as opposed to the predictor variables. It is called a Box-Cox transformation and the goal is to turn potential non-normal dependent variables into a normal shape. Using the boxcox() function in R, I am able to estimate the transformation parameter and visualize the 95% confidence interval of that parameter. After performing a transformation on the response variable, I plotted the diagnostic plot again to see if there is still heteroscedasticity. The cone shape seems to have faded and there is a somewhat random clump in the beginning and a downward trend follows.

```
data_mod <- boxcox(mod,plotit = T)
```



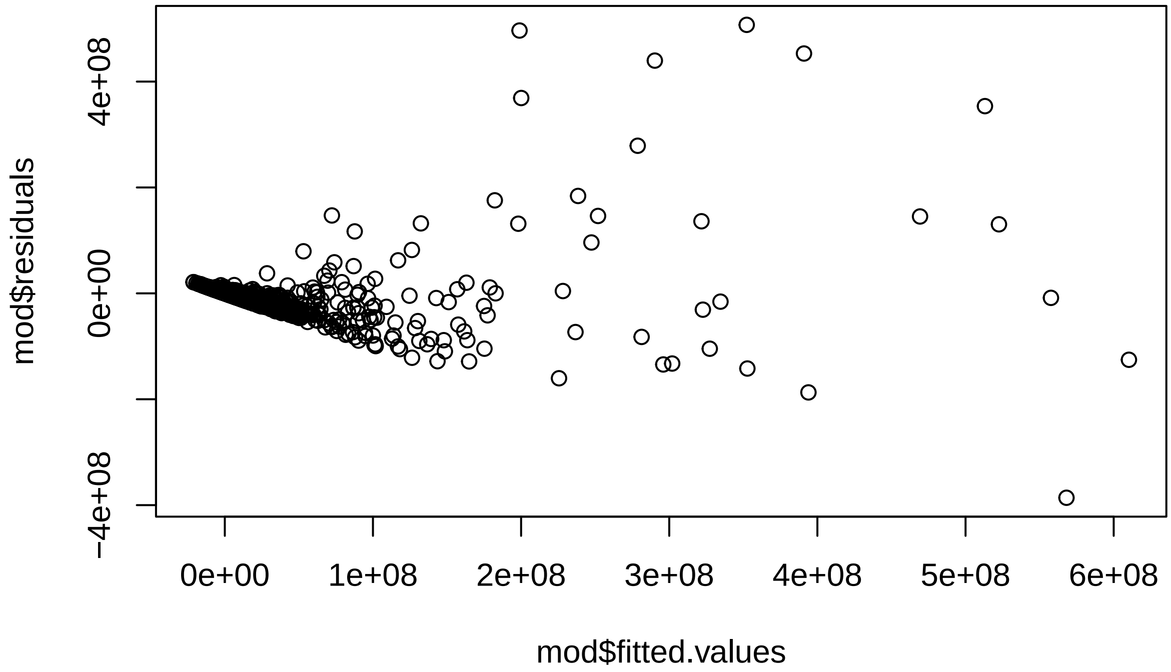
```
lambda <- data_mod$x[which.max(data_mod$y)]
mod2 <- lm((larcenies^lambda)~perCapInc+NumStreet+NumUnderPov+MedRent, data = select_data)
plot(mod2$fitted.values, mod2$residuals)
```



Step 7

Final model

```
mod <- lm((larcenies^2)~perCapInc+NumStreet+NumUnderPov+MedRent, data = select_data)
plot(mod$fitted.values, mod$residuals)
```



Table

```
p_values <- as.data.frame(summary(mod)$coefficients[,4])
p_values['predictors'] <- c('Intercept','perCapInc','NumStreet','NumUnderPov','MedRent')
par_est <- as.data.frame(summary(mod)$coefficients[,1])
par_est['predictors'] <- c('Intercept','perCapInc','NumStreet','NumUnderPov','MedRent')
table <- left_join(p_values,par_est, by = 'predictors')
table <- table[c(2,3,4,5), c(2,3,1)]
colnames(table) <- c('Predictors', 'Parameter Estimate', 'P-Values')
table
```

```
##    Predictors Parameter Estimate     P-Values
## 2   perCapInc           2048.582 6.817567e-16
## 3   NumStreet         427746.869 7.187945e-25
## 4 NumUnderPov           3812.645 0.000000e+00
## 5     MedRent         -23590.178 1.041420e-03
```

R-Squared value of the model

```
summary(mod)$r.squared
```

```
## [1] 0.6758494
```

95% C.I. of the NumUnderPov Predictor

```
fit <- lm((larcenies^2)~NumUnderPov, data = select_data)
confint(fit, 'NumUnderPov', 0.95)
```

```
##                2.5 %   97.5 %
## NumUnderPov 4005.965 4268.376
```

95% C.I. of the response variable given each predictor is equal to its mean

```
mean_capinc <- mean(select_data$perCapInc)
mean_numstreet <- mean(select_data$NumStreet)
mean_numpov <- mean(select_data$NumUnderPov)
mean_medrent <- mean(select_data$MedRent)
pred <- predict(mod, new = data.frame(perCapInc = mean_capinc, NumStreet = mean_numstreet, NumUnderPov
sqrt(pred)
```

```
##        fit      lwr      upr
## 1 3321.131 3099.632 3528.753
```

95% C.I. of the response variable given each predictor is equal to the values of the 13th observation

```
spcf_capinc <- select_data$perCapInc[13]
spcf_numstreet <- select_data$NumStreet[13]
spcf_numpov <- select_data$NumUnderPov[13]
spcf_medrent <- select_data$MedRent[13]
pred <- predict(mod, new = data.frame(perCapInc = spcf_capinc, NumStreet = spcf_numstreet, NumUnderPov
sqrt(pred)
```

```
##        fit      lwr      upr
## 1 6290.091 6075.581 6497.523
```

When I saw that the Box-Cox transformation improved the model slightly I decided to go even further and make the transformation parameter 2. As a result the diagnostic plot looked better than it did at first. Although, there are a few data points that seem to flare out, I believe the majority of data is relatively constant in the variance. I also performed a few inferences that capture the overall strength of the model.

Conclusion

Before I make my final results, I want to make a few comments on the analysis that I have done. There are many different types of further I could have done, but the two things I would have done would be to eliminate some of the outliers that I identified and to test for multicollinearity. I think I could have also used more diversity in my variables since half were dependent on financial status. Barring all that, I concluded that the 4 variables all contribute to the prediction of the number of larcenies, however, I think my strongest variable was NumUnderPov.