University of Waterloo
Faculty of Engineering
Department of Management Sciences

Status Update on Exploring and Predicting Violent Crime in Chicago

University of Waterloo
200 University Ave W, Waterloo, ON N2L 3G1
Waterloo, Ontario, Canada

Prepared by

Yingzi Zhang
20515934
4A Mechatronics Engineering

And

Xiang Li
20574900
4A Mechatronics Engineering

13 November 2018

1 Summary

The R programming language is used due to one of the authors' familiarity with the language. The scope of the project is also slightly reduced and recommendations from the project proposal are applied. Most notably, geographic data collected are all aggregated based on the standardized community areas in Chicago. In addition, the dimension of time is not considered. In total, 12 explanatory variables are collected, each having a unit of 'per community area': average school rating, average SSL rating, total park area, number of hospitals, teenage mother birth rate, infant mortality rate, and proportion of Hispanics, blacks, whites, Asians, and other races, and percent of children in poverty.

Preliminary exploratory data analysis shows that many of these explanatory variables seem to be correlated with the class variable, the percentage of violent crime per community area. Teenage mother birth rate, infant mortality rate, percentage of black people, and percentage of children in poverty had a positive correlation with the class variable. In contrast, percentage of Hispanics, whites, and Asians had a negative correlation.

So far, regression and clustering analysis has been done. With both ordinary least squares and elastic nets linear regression, an R^2 of 0.82 was found, which seems to be reasonably accurate. Using the k-means algorithms and looking at the elbow curve, it seems clear that the data should be grouped into 3 clusters, and that these clusters generally seem to make sense: for example, low teen birth rate is clustered with low infant mortality rate; the vice versa is also clustered together.

2 Data Collection Progress

Table 1 displays all of the final chosen datasets, which increased from the original project proposal due to the inclusion of additional predictors such as public health data, and because some predictors required additional datasets to be able to transform them into usable data for prediction and clustering analysis.

Table 1. Information on Final Chosen Datasets.

Dataset	Number of Rows	What Does a Row Represent?
Crimes from 2001 [1]	6,706,459	Reported Crime
Strategic Subject List [2]	398,684	Person Likely to be Involved in a Shooting
Chicago Public Schools - School Profile Information SY1718 [3]	661	School
Population and Poverty Data by Chicago Community Area [4]	77	Community Area
Parks - Chicago Park District Park Boundaries (current) [5]	N/A (Shapely File of 597 Parks)	N/A (Shapely File of 597 Parks)
Boundaries - Community Areas (current) [6]	N/A (Shapely File of 77 Community Areas)	N/A (Shapely File of 77 Community Areas)
Hospitals – Chicago [7]	N/A (Shapely File of 42 Hospitals)	N/A (Shapely File of 42 Hospitals)
Public Health Statistics - Births to mothers aged 15-19 years old in Chicago, by year, 1999-2009 [8]	77	Community Area
Public Health Statistics- Infant mortality in Chicago, 2005– 2009 [9]	77	Community Area

The Appendix will describe how all the predictors and the class variable are obtained from the above datasets. Note that the code written in R to conduct these data transformations are also available in the Appendix section.

3 Exploratory Data Analysis Progress

Scatter plots, histograms, and box plots are used to analyse each variable. It is worth noting that some variables are not normally distributed, but rather exponentially distributed, including the class variable: the violent crime rate. Explanatory variables that have non-normal distributions include: total park area, number of hospitals, percentage of Hispanics, percentage of whites, and percentage of Asians. Interestingly, the distribution of the percentage of black people has an upside-down bell curve shape, which implies that community areas generally either have few to no black people, or have a large percentage of black people. Very few seem to have a moderate percentage of black people.

Finally, out of the 12 explanatory variables, 7 of them visually have a clear correlation with the class variable, as seen from scatter plots. The rest do not seem to have as strong of a correlation. Specifically, teenage mother birth rate, infant mortality rate, percentage of black people, and percentage of children in poverty had a strong positive correlation with the class variable. In contrast, percentage of Hispanics, whites, and Asians had a strong negative correlation. Some of these correlations also seemed to be non-linear. These include: total park area, teenage mother birth rate, infant mortality rate, percentage of Hispanics, percentage of whites, percentage of Asians, and percentage of children in poverty.

Note that all scatter plots, histograms, and box plots are included in the Appendix, and short comments on each are included.

4 Supervised and Unsupervised Learning Progress

Numeric regression and clustering analysis were successfully conducted as of current. However, association rule mining is not completed yet.

4.1 Numeric Regression Analysis

Three types of regressions were used: ordinary least squares (OLS) linear regression, elastic net regularized regression, and generalized additive linear model. Elastic nets is a mixture of lasso and ridge regression, both of which penalize the coefficients in regular OLS regression in order to prevent overfitting. Ridge regression is where the square of each coefficient for each explanatory variable is penalized, and lasso regression is where the absolute value of all the coefficients for each explanatory variable is penalized. Lasso regression can force coefficients of uncorrelated features to be zero (which can be seen as feature selection), though this may lead to information loss. Thus, elastic nets are seen as the “best of both worlds”. However, as seen from the previous section of this report, many of the relationships between the explanatory variables and the class variable seem to be nonlinear in nature. Thus, the authors believed GAM might be give fruitful results, since the GAM algorithm attempts to fit polynomial splines on the explanatory variables and conduct an additive linear regression on these transformed predictors.

10-fold cross validation was used to obtain the alpha and lambda parameters for the elastic nets regression. Note that alpha is mixing parameter with a range from 0 to 1. A value of 0 represents 100% ridge regression, and a value of 1 represents

100% lasso regression. Also note that lambda is the amount of penalization with increasing coefficients for the explanatory variables in the final model.

Leave-one-out cross validation was used to obtain three performance metrics for each regression type, which include: root mean squared error, (RMSE), coefficient of determination (R^2), and mean absolute error (MAE).

4.1.1 OLS Linear Regression

Figure 1 shows the predicted class variables versus the actual class variables for OLS linear regression. Ideally, the points should be uniformly distributed around the line $y = x$. As shown, the predictions have this trend, which means the predictions seem to be roughly accurate. However, the general trend has somewhat of a small nonlinear concave curve.

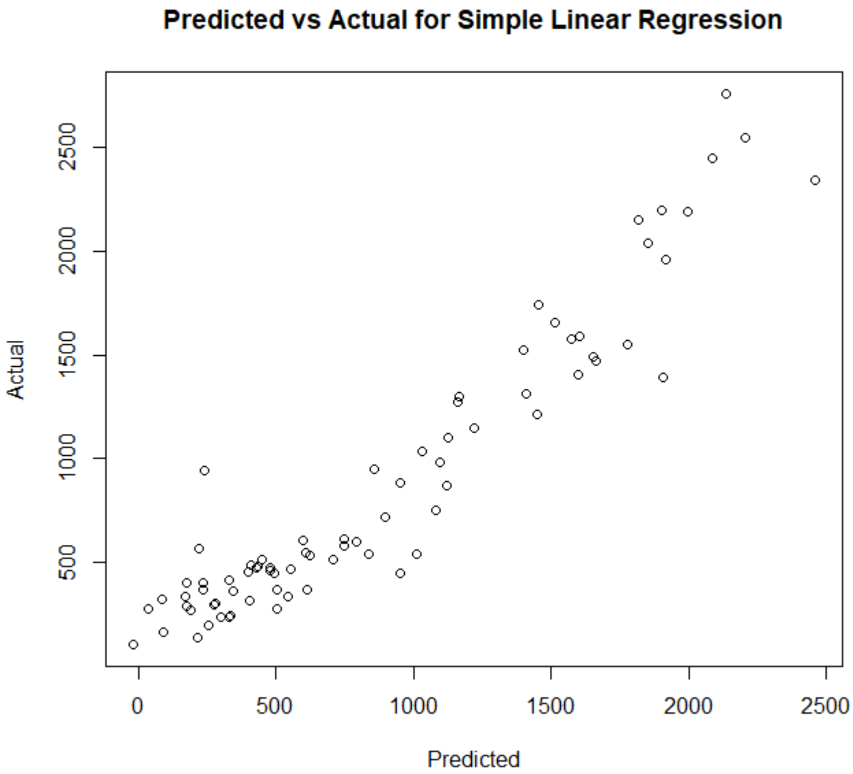

Figure 1. Predicted Class Values vs Actual Values for OLS Linear Regression

This concavity is more pronounced in the residual plot shown in *Figure 2*. Evidently, the scatter plots show that the data's uniformity around the line $y = 0$ is not independent of the value of x (the predicted values), which suggests linear regression may not be the best algorithm.

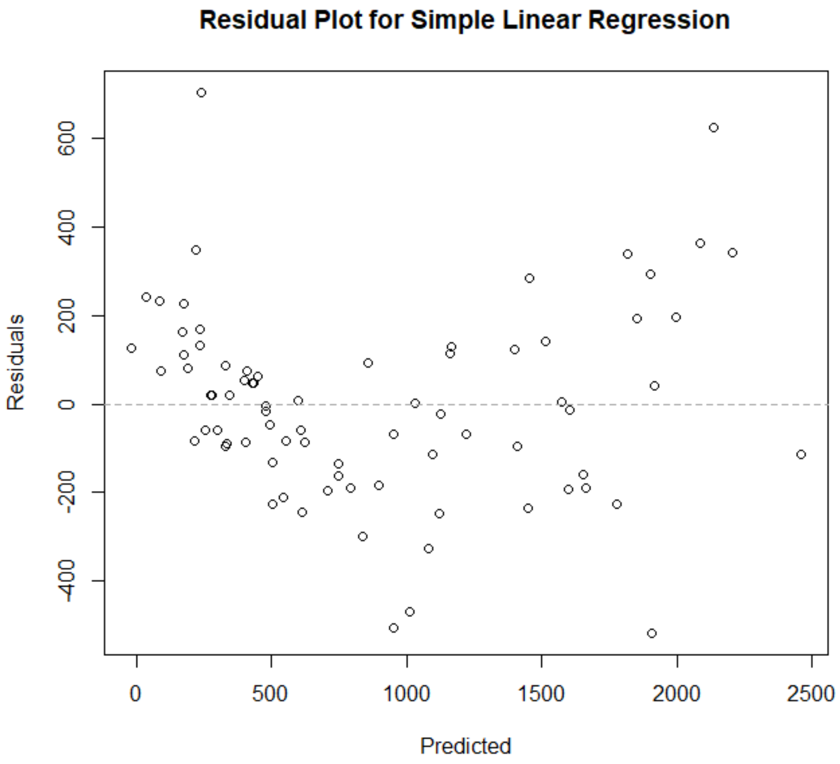

Figure 2. Residual Plot for OLS Linear Regression.

Figure 3 shows the residual histogram for OLS linear regression. While it shows a roughly uniform distribution centred around 0, it does not show the concavity pattern from the previous figure, which is one weakness of histogram plots.

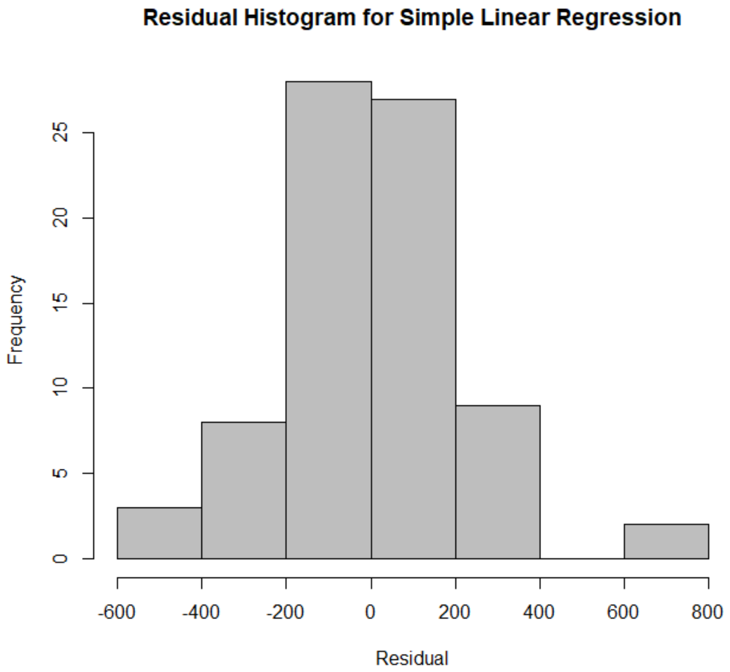

Figure 3. Residual Histogram for OLS Linear Regression.

Table 2 shows the performance metrics of this algorithm.

Table 2. Performance Metrics for OLS Linear Regression

RMSE	R^2	MAE
285.243497339416	0.822397725231063	206.537526304179

4.1.2 Elastic Net Regression

Using 10-fold cross validation, the best alpha and lambda values were determined to be 0 (100% ridge regression) and 1, respectively.

Figure 4, Figure 5, and Figure 6 show similar types of plots as ones shown for OLS regression in the last subsection. It seems that there is not a large visual difference between elastic net and OLS, though the histogram distribution slightly changed.

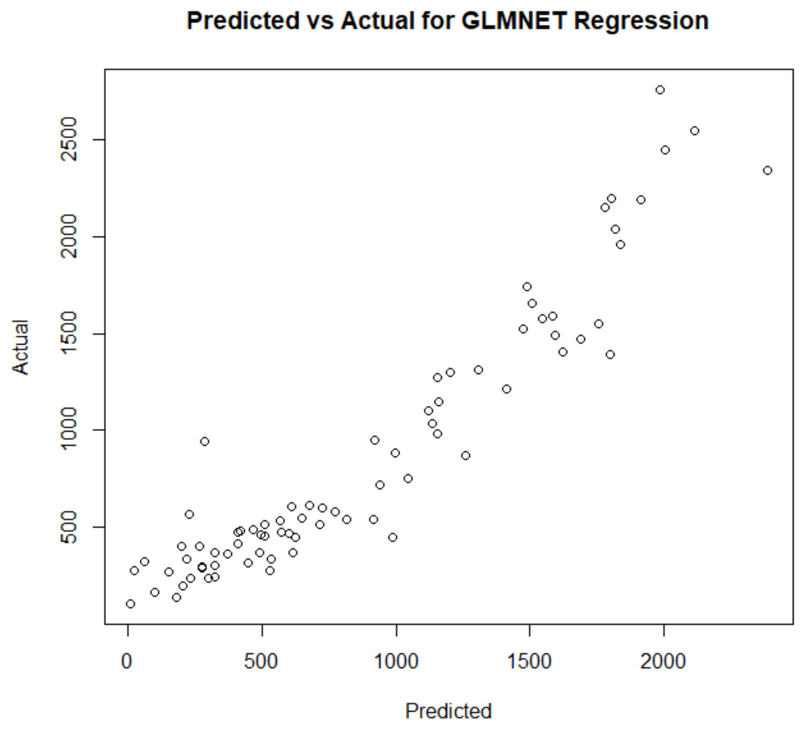

Figure 4. Predicted Class Values vs Actual Values for Elastic Net Regression.

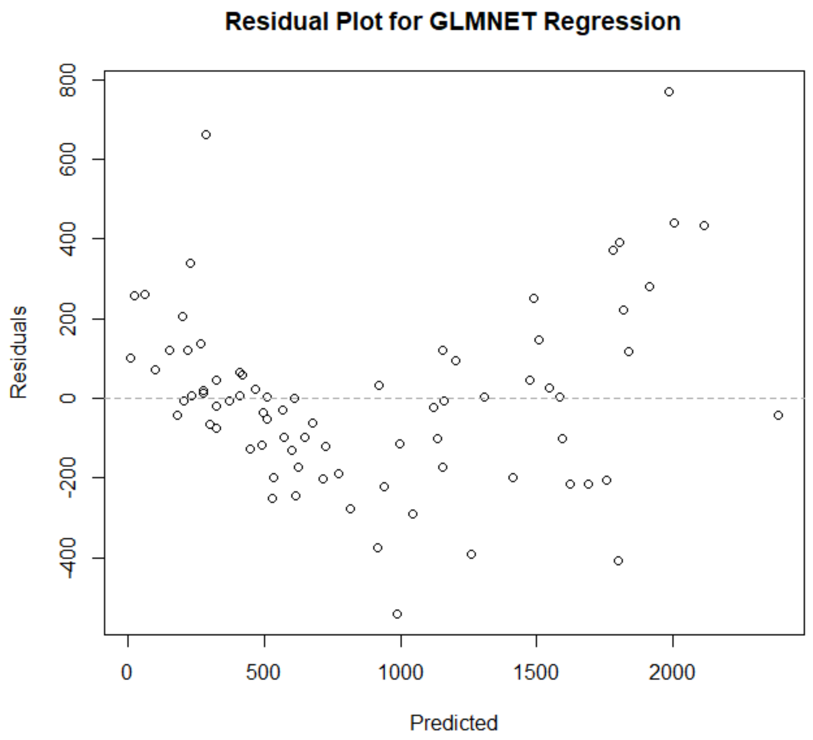

Figure 5. Residual Plot for Elastic Net Regression.

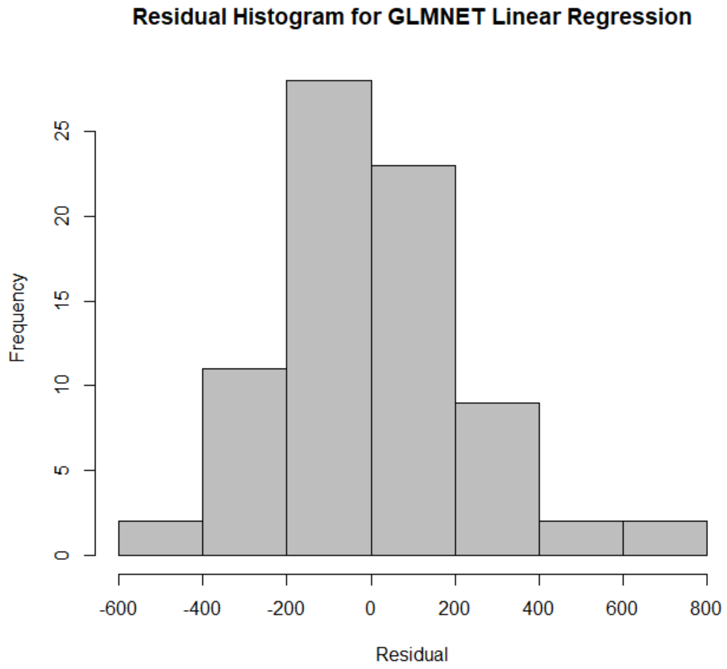

Figure 6. Residual Histogram for Elastic Net Regression.

Ultimately, Table 3 shows that the performance metrics only improved very slightly from OLS. The changed digits are bolded for ease of reading.

Table 3. Performance Metrics for Elastic Net Linear Regression

RMSE	R^2	MAE
284.**112757633739**	0.82**3577225374136**	206.53**3355885329**

4.1.3 GAM Regression

Using the scatter plots of each explanatory variable versus the class variable in the "Exploratory Data Analysis" section of this report, the relevant explanatory variables that may have nonlinear relationships with the class variable were identified. These include: total park area, birth rate for teenage mothers, infant mortality rate, percentage of Hispanics, percentage of white people, percentage of Asian people, and percentage of children in poverty. Thus, the resulting formula used is the following:

$$
\begin{aligned}
y = avgSchoolRating + avgSSLRating + s(totalParkArea) \\
+ has3OrMoreHospitals + s(teenMomRate) \\
+ s(infantMortalityRate) + s(hispanic) + black + s(white) \\
+ s(asian) + other + s(percentChildrenInPoverty)
\end{aligned}
$$

Where the function s is a function that fits a spline between the input and the outcome.

Figure 7 seems to indicate that the GAM model built on all 77 community area data points reduces the nonlinear concavity. In addition, Figure 8 and Figure 9 shows that the range of the residual decreased to $(-300{,}400)$. These all seem to indicate that the GAM model fits the data better.

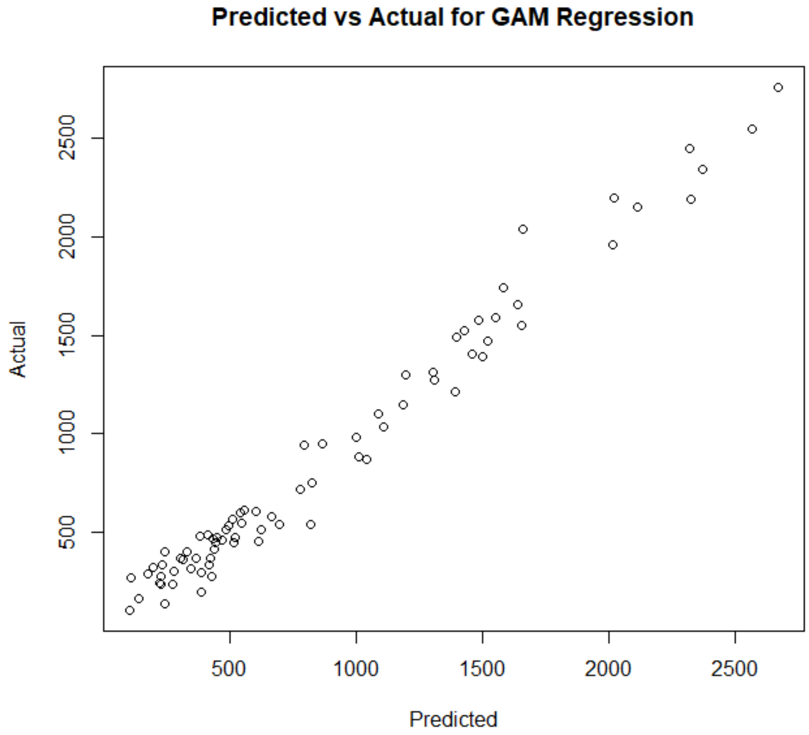

Figure 7. Predicted Class Values vs Actual Values for GAM Regression.

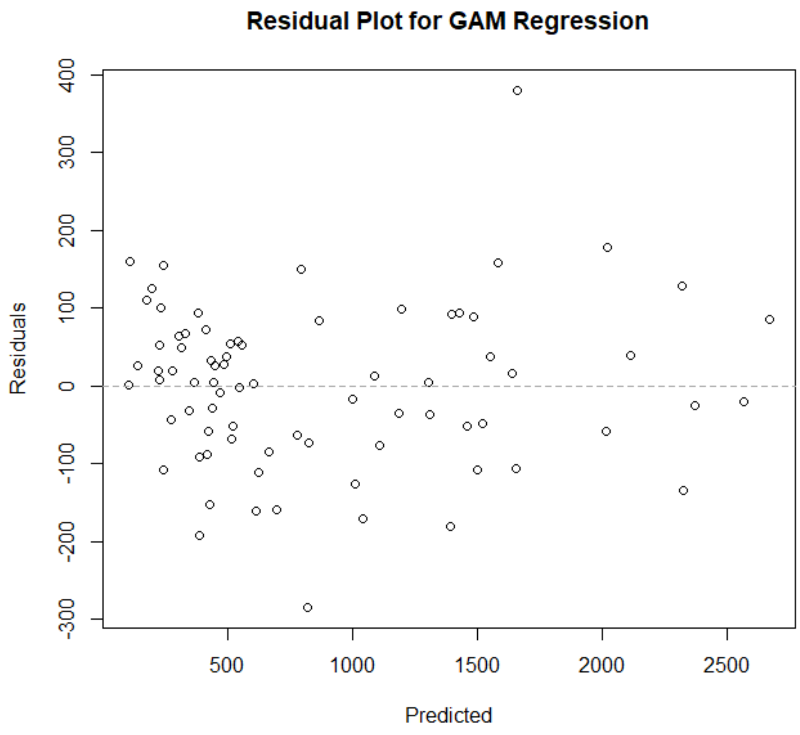

Figure 8. Residual Plot for GAM Regression.

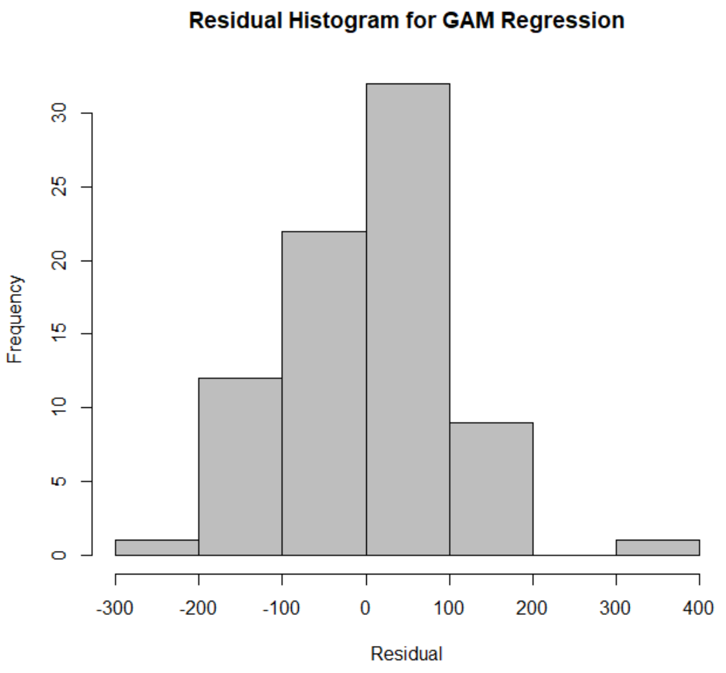

Figure 9. Residual Histogram for GAM Regression.

However, the performance metrics obtained from leave-one-out cross validation shown in Table 4 indicates that the GAM model performance on unseen data is actually worse than the previous two regression algorithms. With leave-one-out cross validation, 76 GAM models are each built from randomly selected 76 community area data points and tested on the single remaining community area.

Table 4. Performance Metrics for GAM Regression.

RMSE	R^2	MAE
308.050825959162	0.**796551460938803**	**188.925268679306**

Table 5 displays the performance metric of the GAM model built with all 77 community area data points. It is clear that by comparing Table 5 and Table 4, leaving even one data point out of the model significantly changes the performance metrics. Thus, the GAM algorithm heavily overfits.

Table 5. Performance Metrics for GAM Regression (Built With All Data Points)

RMSE	R^2	MAE
103.30301646473	0.**976500401871001**	**79.55816922112**

It seems that out of all three regression algorithms, elastic net regression performs the best, though it is only marginally better than OLS regression. Future recommendations include using other nonlinear regression algorithms other than GAM, as its fitted splines may use polynomials with higher than necessary orders, which can cause unnecessary overfitting to occur.

4.2 Clustering Analysis

The K-means clustering analysis is performed on the 12 explanatory variables using R Language's method, "kmeans". Since clustering works the best under isotropic conditions, all of the explanatory variables are normalized and scaled to the range of 0 to 1.

4.2.1 Analysis of Number of Centers

For the first part of this analysis, the optimal number of cluster centers from 1 to 25 is analyzed, with the total within-cluster sum of squares data being collected for each. The result is shown in Figure 10.

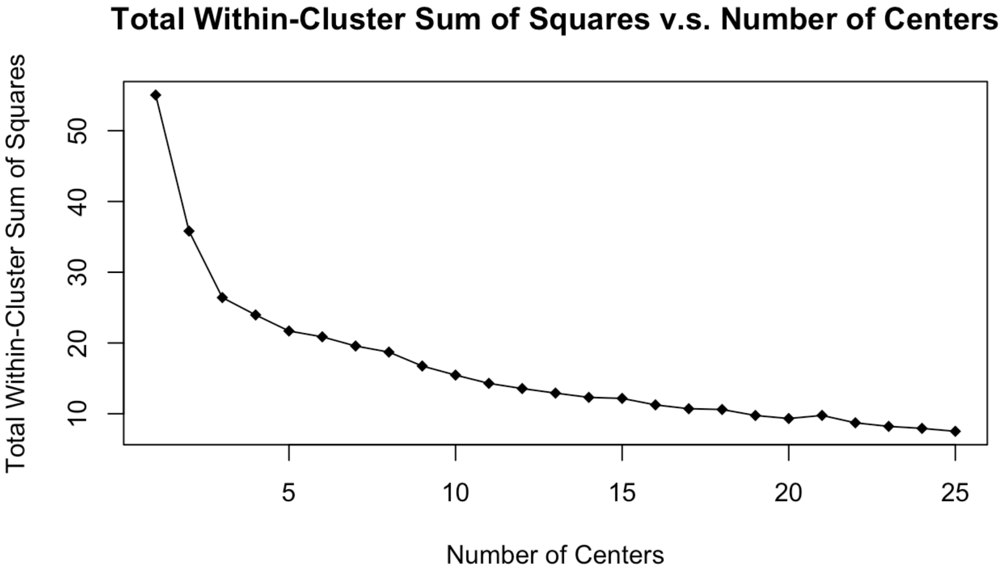

Figure 10. Total within-cluster sum of squares versus the number of centers

As shown in Figure 10, there is a significant drop in the sum of squares until the number of centers reaches 3. The decrease in the sum of squares substantially lowers as the number of centers increases from 3 to 25.

4.2.2 Analysis of Different Starting Points for Centers

Another analysis is performed on the resulting sum of squares when different starting points are randomly chosen for the clustering center. Two trials are executed, and the result is shown in Figure 11.

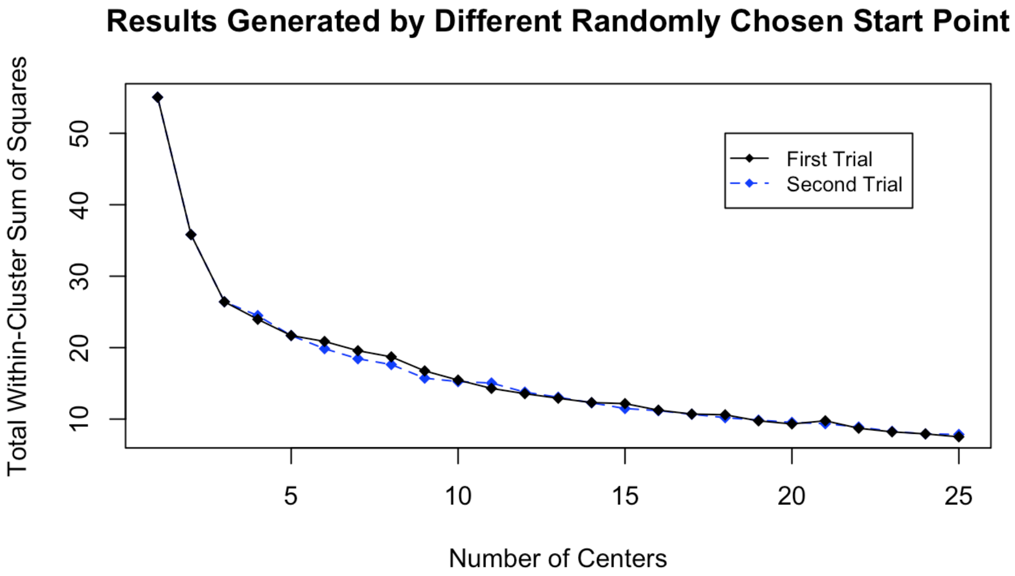

Figure 11. Sum of squares by different randomly chosen start point of center

As shown in the result, different choices of starting points of the clustering centers will result in different sum of squares, which is reasonable due to the randomized nature of the k-means algorithm. Another analysis is then conducted, with 50 different sets of starting points chosen. Figure 12 indicates the average sum of squares obtained from the 50 sets of centers for each number of centers.

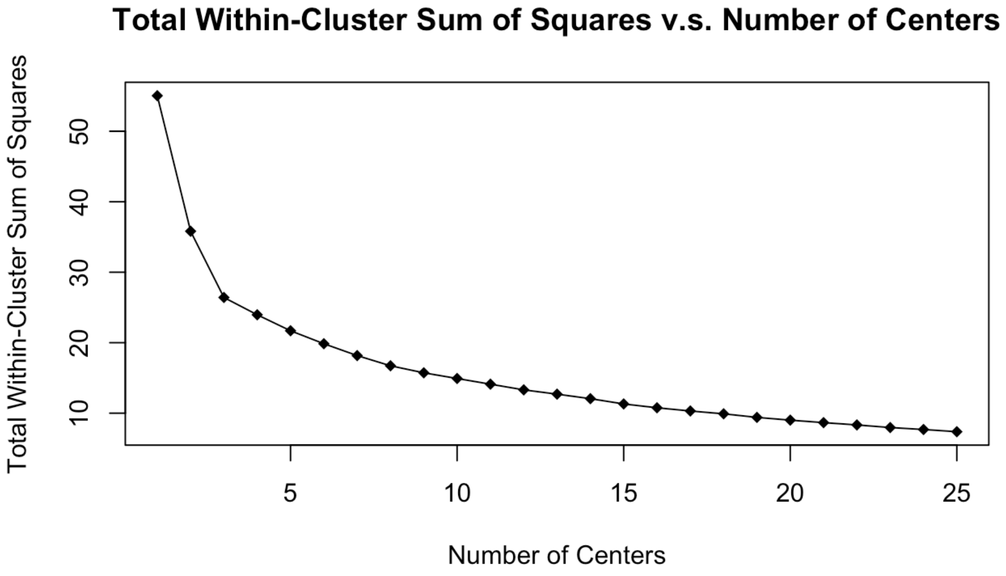

Figure 12. Average sum of squares with 50 sets of starting points

By looking at the above figure, it becomes evident that the significant drop in sum of squares still stops when 3 clustering centers is selected. Therefore, for the clustering analysis, the number of clustering centers is set to 3. The average total within-cluster sum of squares for 3 clustering centers is 26.416911.

4.2.3 Clustering between Each Two Explanatory Variables

After the number of clustering centers is determined, a plot is made to indicate the clustering between each two explanatory variables with 3 clustering centers, as shown in Figure 13.

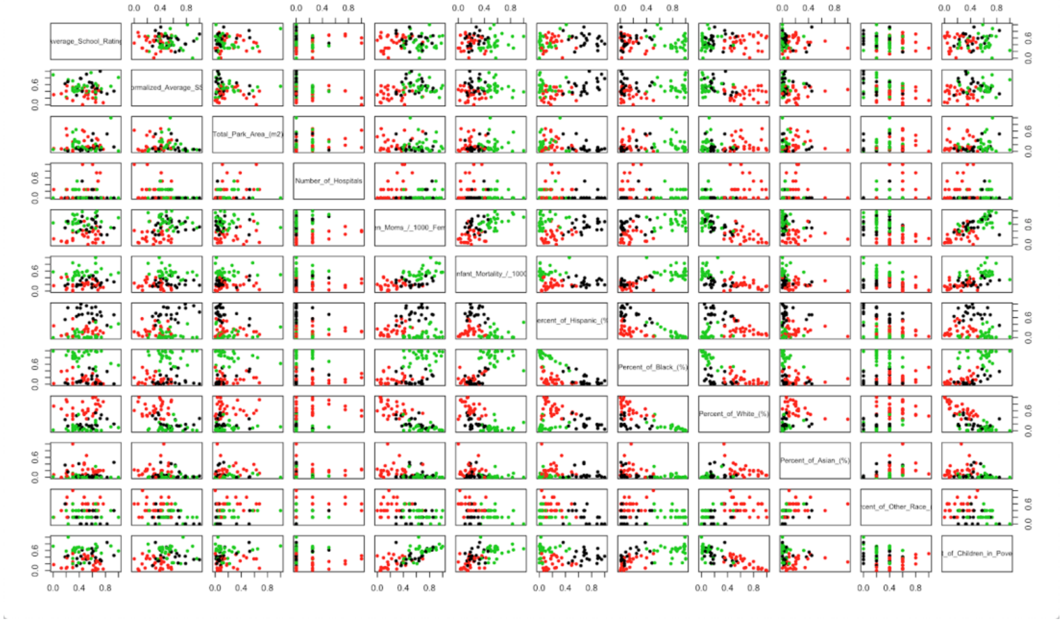

Figure 13. Clustering Relationship between Each Two Explanatory Variables

Figure 13 shows that most of the plots are cleanly divided into the three clusters from the k-means algorithm, specifically at columns 5, 6, 7, 8, 9, and 12, representing teen mom birth rate, infant mortality rate, percent of Hispanic, percent of black, percent of white, and percent of children in poverty, respectively. For example, it can be observed from the plot in column 5 and row 6 (the plot which shows the clustering between teenage mom rate and infant mortality rate), that low teenage mom rate seems to be associated with low infant mortality rate, and high teenage mom rate is associated with high infant mortality rate. More details will be discussed and included in the final report of the project.

4.3 Association Rule Mining Analysis

Association rule mining is still in progress as of current. Current plans are to preprocess and create a CSV file of the explanatory variables in R. This CSV file will contain data that is interpretable with Python’s Apriori’s library. The reason this Python will be used is because the authors of this report are not familiar with association rule mining libraries in R, and because the authors believe they should gain more experience in Python.

5 Bibliography

[1] City of Chicago, "Crimes - 2001 to present," [Online]. Available: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2. [Accessed 23 September 2018].

[2] City of Chicago, "Strategic Subject List," 7 December 2017. [Online]. Available: https://data.cityofchicago.org/Public-Safety/Strategic-Subject-List/4aki-r3np. [Accessed 23 September 2018].

[3] City of Chicago, "Chicago Public Schools - School Profile Information SY1718," 19 October 2018. [Online]. Available: https://data.cityofchicago.org/Education/Chicago-Public-Schools-School-Profile-Information-/w4qj-h7bg. [Accessed 26 October 2018].

[4] Illinois Action for Children, "Population and Poverty Data by Chicago Community Area," [Online]. Available: http://www.actforchildren.org/wp-content/uploads/2018/01/Census-Data-by-Chicago-Community-Area-2017.pdf. [Accessed 23 September 2018].

[5] City of Chicago, "Parks - Chicago Park District Park Boundaries (current)," 28 September 2018. [Online]. Available: https://data.cityofchicago.org/Parks-Recreation/Parks-Chicago-Park-District-Park-Boundaries-curren/ej32-qgdr. [Accessed 26 October 2018].

[6] City of Chicago, "Boundaries - Community Areas (current)," 11 July 2018. [Online]. Available: https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Community-Areas-current-/cauq-8yn6. [Accessed 26 October 2018].

[7] City of Chicago, "Hospitals - Chicago," 28 August 2011. [Online]. Available: https://data.cityofchicago.org/Health-Human-Services/Hospitals-Chicago/ucpz-2r55. [Accessed 4 November 2018].

[8] City of Chicago, "Public Health Statistics - Births to mothers aged 15-19 years old in Chicago, by year, 1999-2009," 28 March 2013. [Online]. Available: https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Births-to-mothers-aged-15/9kva-bt6k. [Accessed 4 November 2018].

[9] City of Chicago, "Public Health Statistics- Infant mortality in Chicago, 2005– 2009," 11 April 2014. [Online]. Available: https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Infant-mortality-in-Chica/bfhr-4ckq. [Accessed 4 November 2018].

[10] City of Chicago, "Parks - Chicago Park District Park Boundaries (current)," [Online]. Available: https://data.cityofchicago.org/Parks-Recreation/Parks-Chicago-Park-District-Park-Boundaries-curren/ej32-qgdr. [Accessed 23 September 2018].

6 Appendix

6.1 Data Collection Method

6.1.1 Class Variable

The class variable is the percent of violent crime per 1000 people in the specified community area. This is obtained via the following formula:

*violentCrimeForCommunityArea * 1000 / populationOfCommunityArea*

In order to obtain the total number of violent crime in a community area, the "Crimes from 2001" dataset is used. In this dataset, each crime is given a type and the community area the crime occurred in, and each crime's type was used to filter for violent types only. Examples include: assault, battery, and homicide. Finally, the population of each community area is obtained from census data.

6.1.2 Average School Rating

The average school rating is the average rating of schools in a certain community area, and it is obtained by formula:

Sum(schoolRatingForCommunityArea) / numSchoolsInCommunityArea

The ratings for each school are from dataset, "Chicago Public Schools - School Profile Information SY1718" [3]. In this dataset, the general information about schools is given, such as names, location, ratings, and student count. The strings representing school levels in the "Overall_Rating" column is translated to numerical scores from 1 to 5 according to the level.

6.1.3 Average SSL Rating

Recall that the SSL is defined as a numerical score with a range of 0 to 500, representing the likelihood of an offender to be involved in a shooting in the near future. 0 is extremely low risk and 500 is extremely high risk. The average SSL rating predictor is simply calculated as the average SSL rating of all strategic subject people in a certain community area. The SSL ratings is from dataset, "Strategic Subject List" [2], which takes samples from the list of arrest data from August 1, 2012 to July 31, 2016.

6.1.4 Total Park Area

To obtain the total park area for each community area, all park shape files [5] and all community area shape files [6] were obtained. Then, using the Raster library in R, the intersection area of each park with each community area is calculated. These intersection areas are then summated for each community area.

6.1.5 Number of Hospitals

Obtaining the number of hospitals for each community area from the hospital data [7] was simple, since each hospital data point included the community area it is located in.

6.1.6 Birth Rate by Teenage Mothers

The dataset of birth rate by teenage mothers has all the rates from 1999 to 2009 [8]. The average of all these birth rates for each community areas is used.

6.1.7 Infant Mortality Rate

Similar to teenage mother birth rate, infant mortality rate for each community area is calculated as the average infant mortality rate from all years in which data was available: from 2005 to 2009 [9]. Two values for one community area from two years had null values, so the average of the non-null values was calculated.

6.1.8 Proportion of Hispanic People

Nothing was needed to be transformed for this predictor [4].

6.1.9 Proportion of Black People

Nothing was needed to be transformed for this predictor [4].

6.1.10 Proportion of White People

Nothing was needed to be transformed for this predictor [4].

6.1.11 Proportion of Asians

Nothing was needed to be transformed for this predictor [4].

6.1.12 Proportion of Other Races

Nothing was needed to be transformed for this predictor [4].

6.1.13 Percent of Children in Poverty

Note that actual poverty rate was unable to be obtained, but percentage of children in poverty was easily obtainable and thus is used instead.

The original dataset of children poverty rates in 2018 has children separated between ages 0 to 5 and ages 6 to 12, and each of these two groups had a poverty rate percentage [4]. A weighted average based on the total population of these two groups was used to obtain the average poverty rate of all children across these two age ranges. This was done in Excel rather than in R, so no code for this data preprocessing exists in the Appendix section.

6.2 Exploratory Data Analysis Plots and Detailed Comments

6.2.1 Class Variable

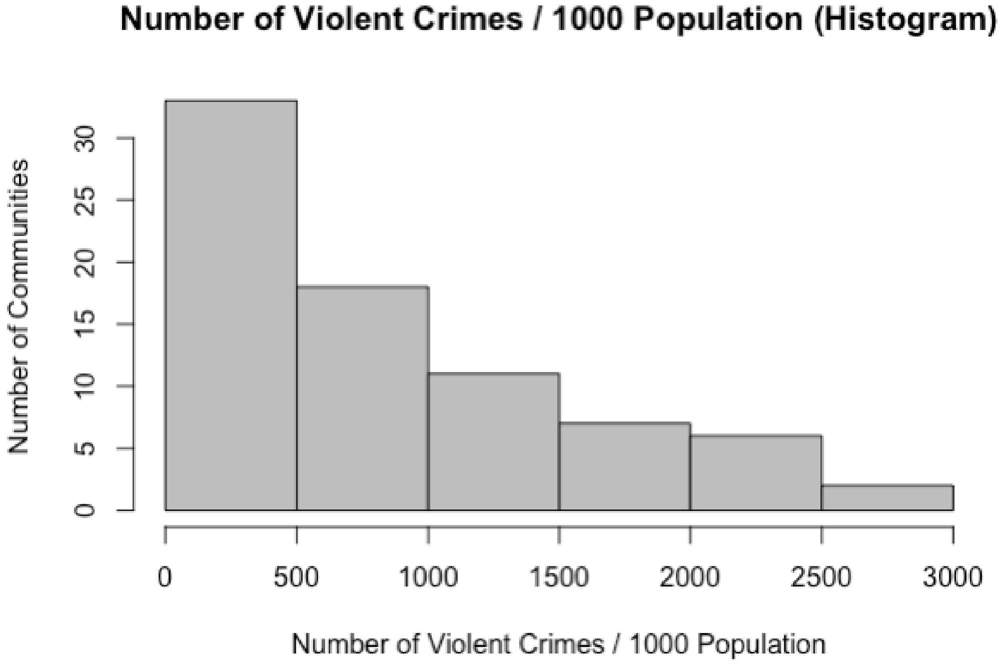

Figure 14. Histogram of Class Variable

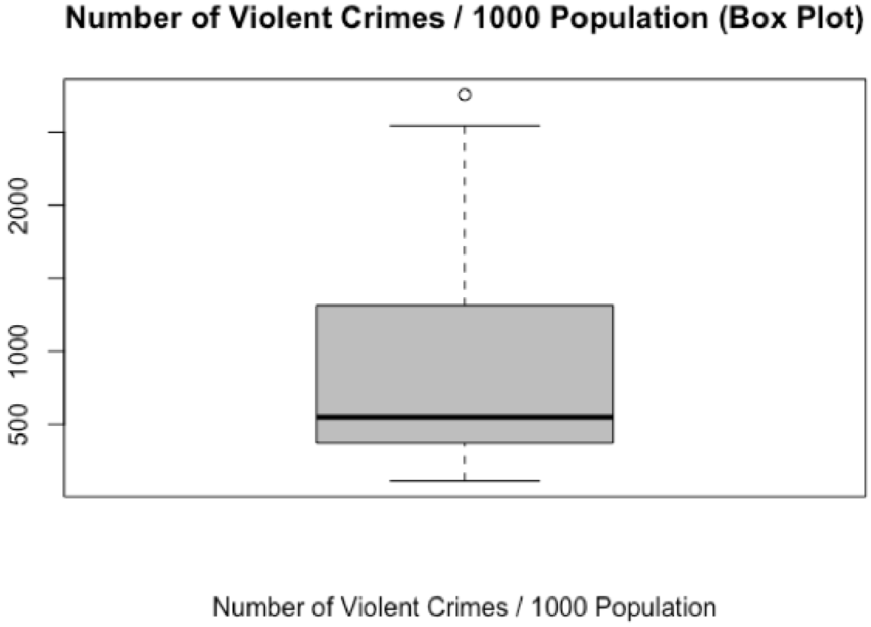

Figure 15. Box Plot of Class Variable

From the histogram, the number of communities exponentially decreases from low to high number of violent crimes. About half of the communities have under 500 cumulative violent crimes per 1000 people since 2001. Two communities, Washington Park and Fuller Park, have more than 2500 cumulative violent crimes per 1000 people. The only one outlier shown in the box plot is also Fuller Park, with 2757 calculated cumulative violent crimes per 1000 people and only 2876 population in 2010. This explains why it does not take too many criminal acts to boost its violent crime rate and why the community area is frequently in the news as one of the most dangerous places to live in Chicago.

6.2.2 Average School Rating

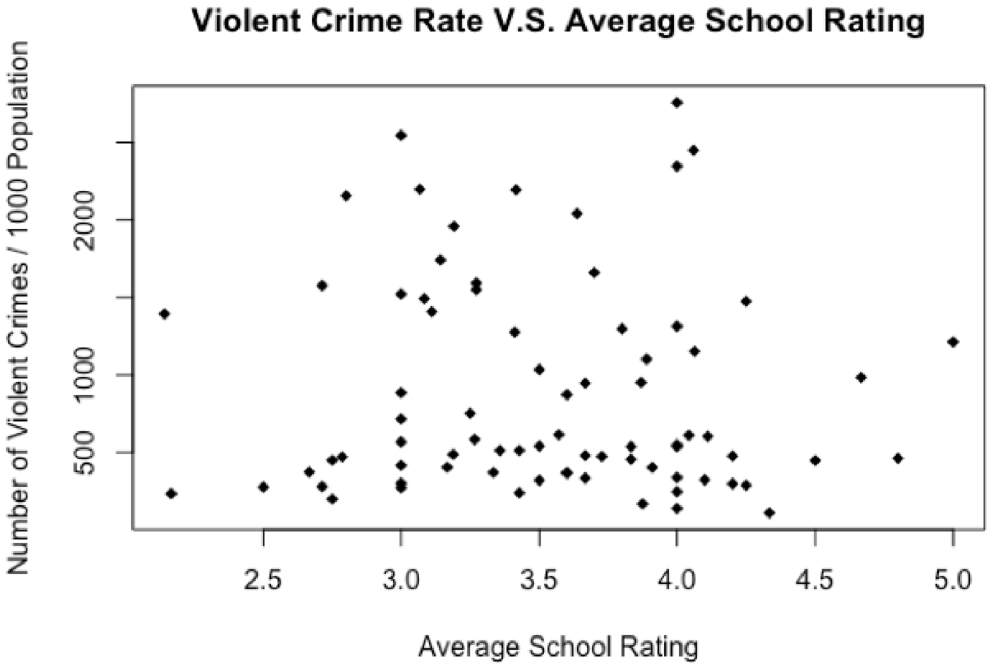

Figure 16. Scatter Plot for Average School Rating

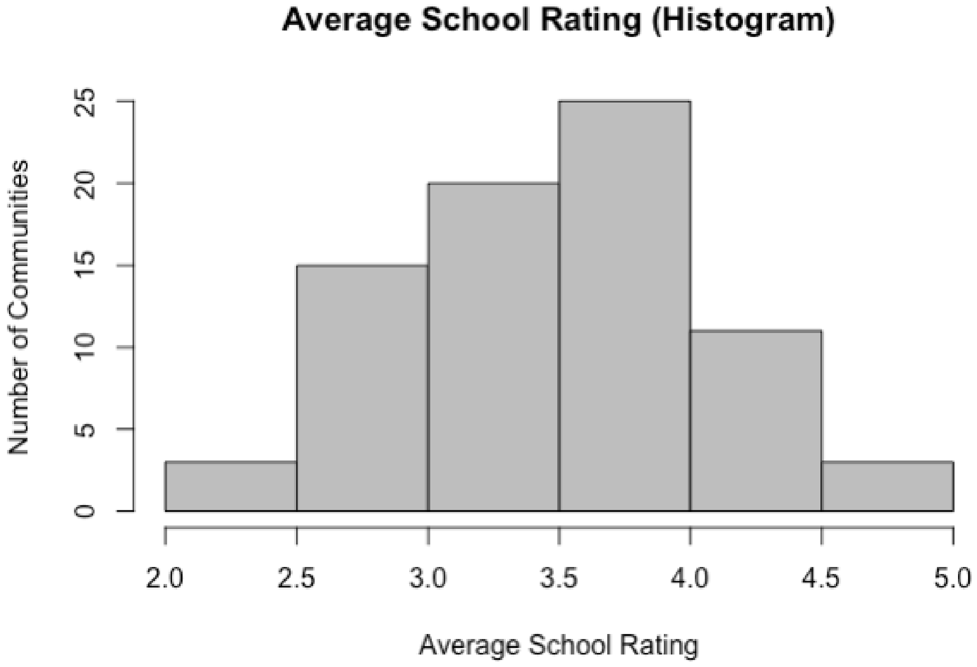

Figure 17. Histogram for Average School Rating

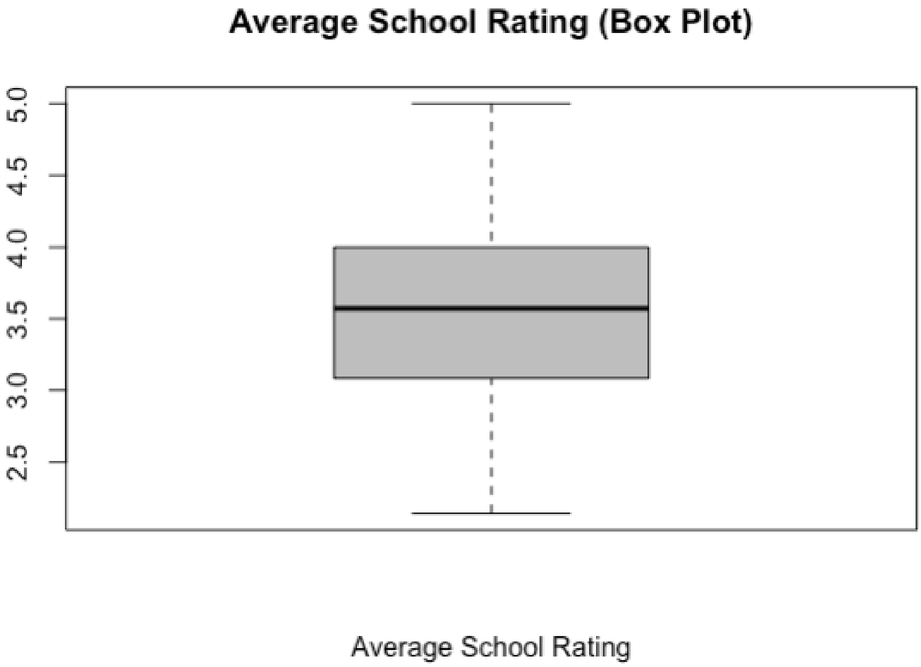

Figure 18. Box Plot for Average School Rating

The histogram and the box plot show that the average school rating is normally distributed. No clear correlation is indicated in the scatter plot, therefore, the average school rating should have no or a very small weight in the prediction model.

6.2.3 Average SSL Rating

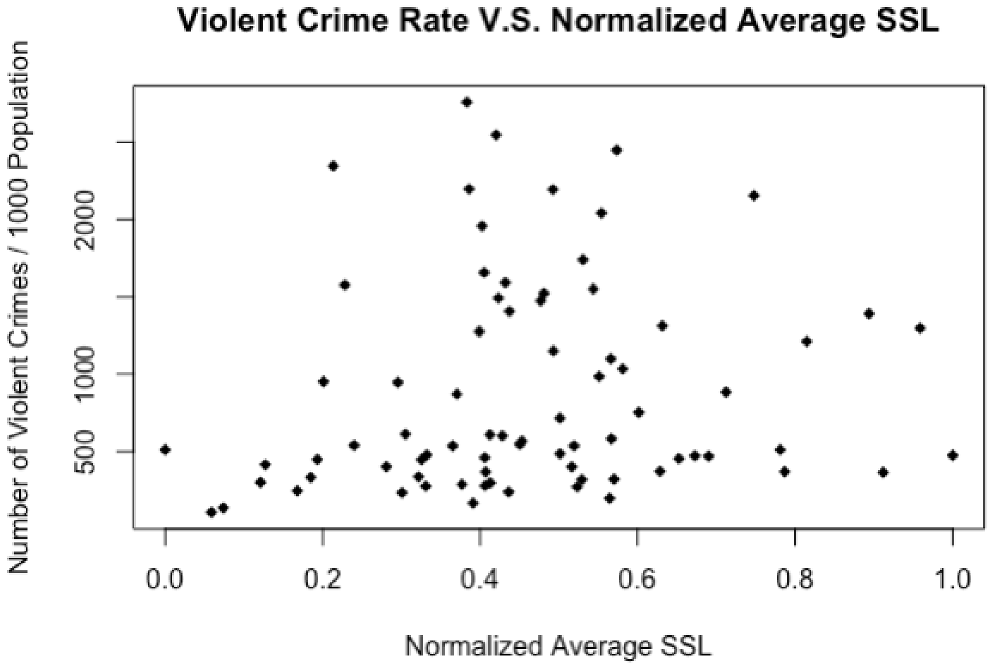

Figure 19. Scatter Plot for Normalized Average SSL Rating

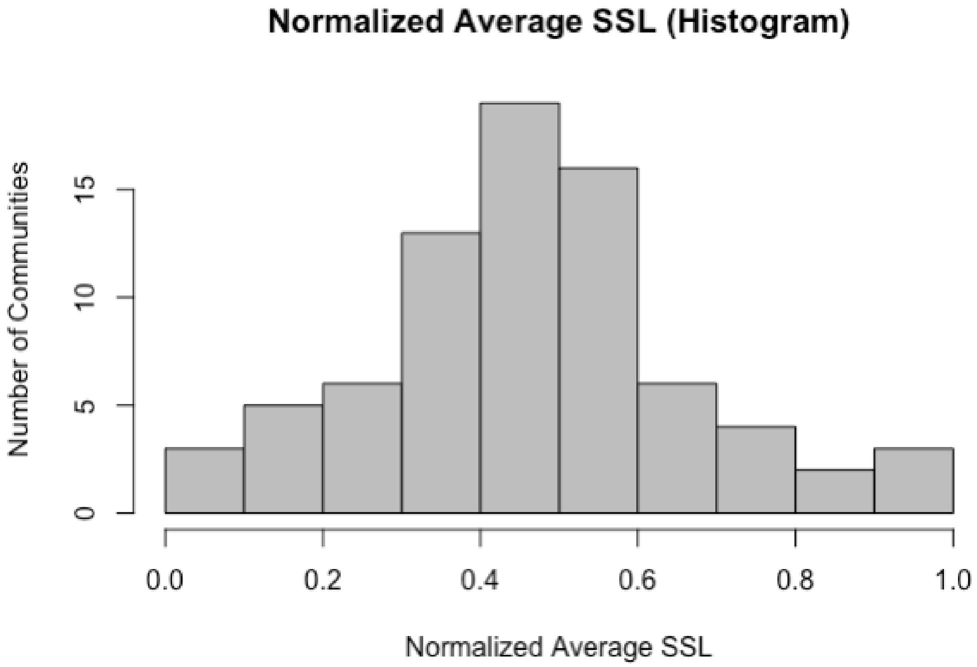

Figure 20. Histogram for Normalized Average SSL Rating

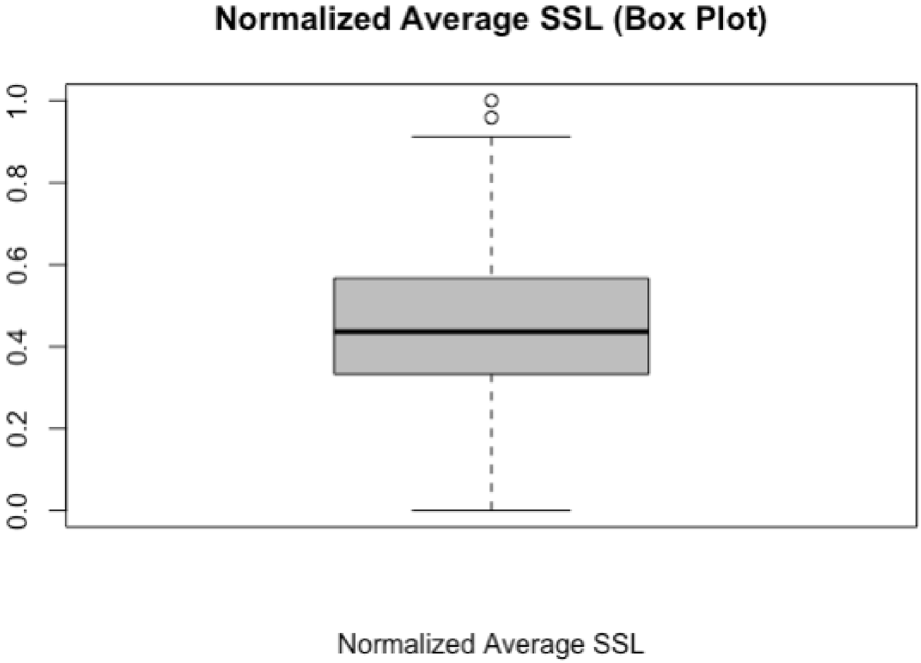

Figure 21. Box Plot for Normalized Average SSL Rating

Note that the average SSL rating is normalized and scaled down from the original 266 to 304 range to 0 to 1 range. It is normally distributed, and there is no evident correlation between the SSL rating and the number of violent crimes.

6.2.4 Total Park Area

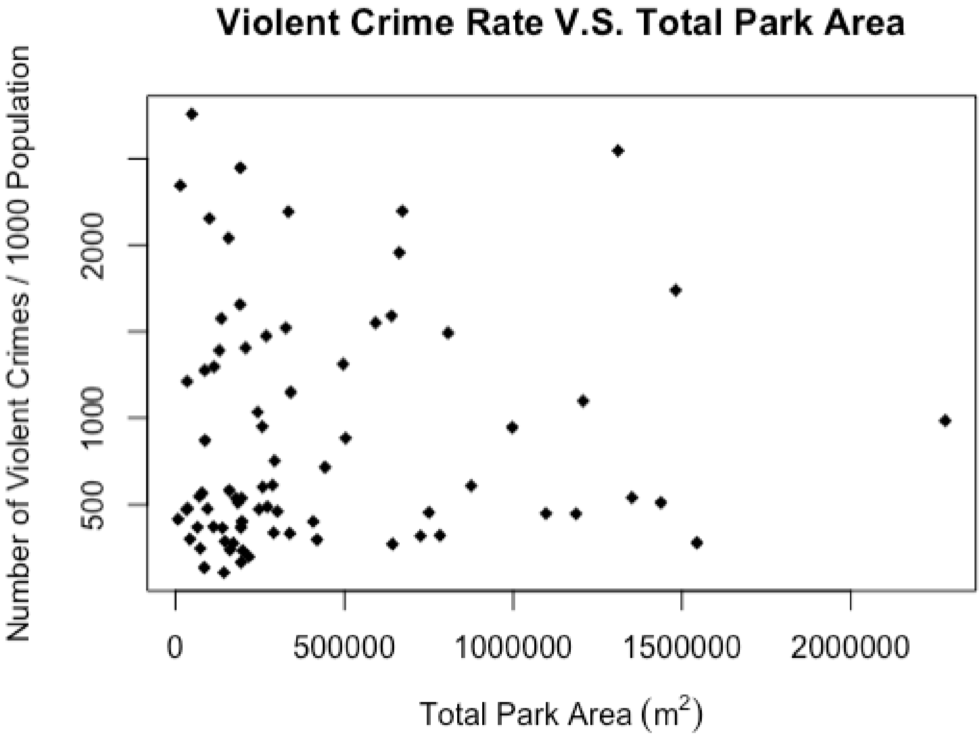

Figure 22. Scatter Plot for Total Park Area

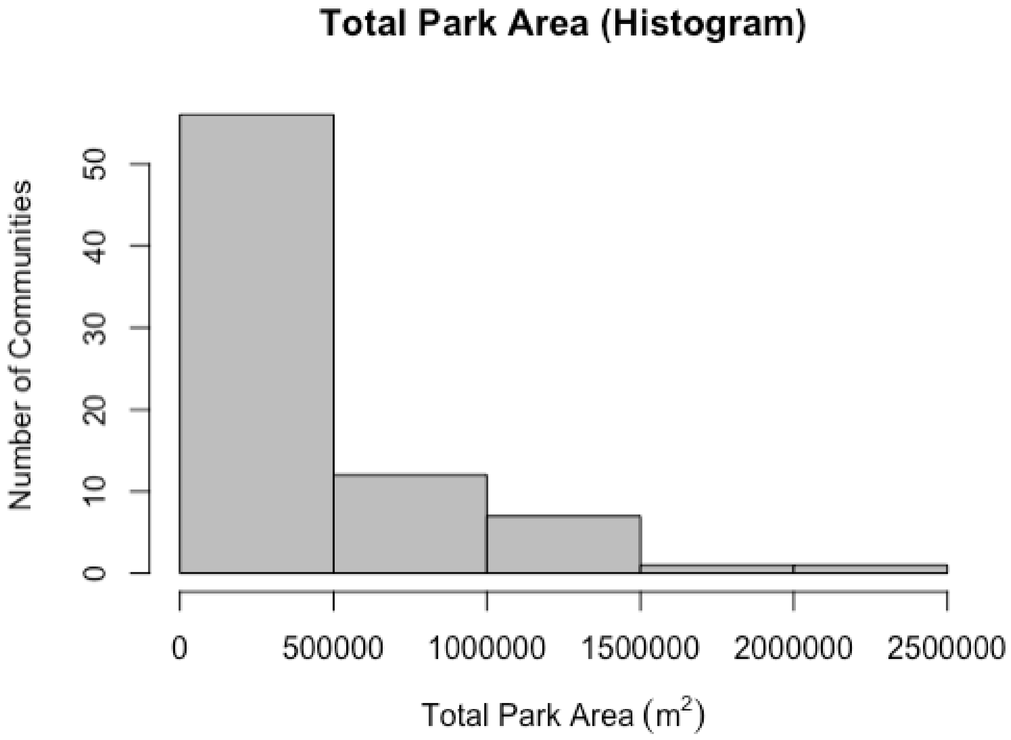

Figure 23. Histogram for Total Park Area

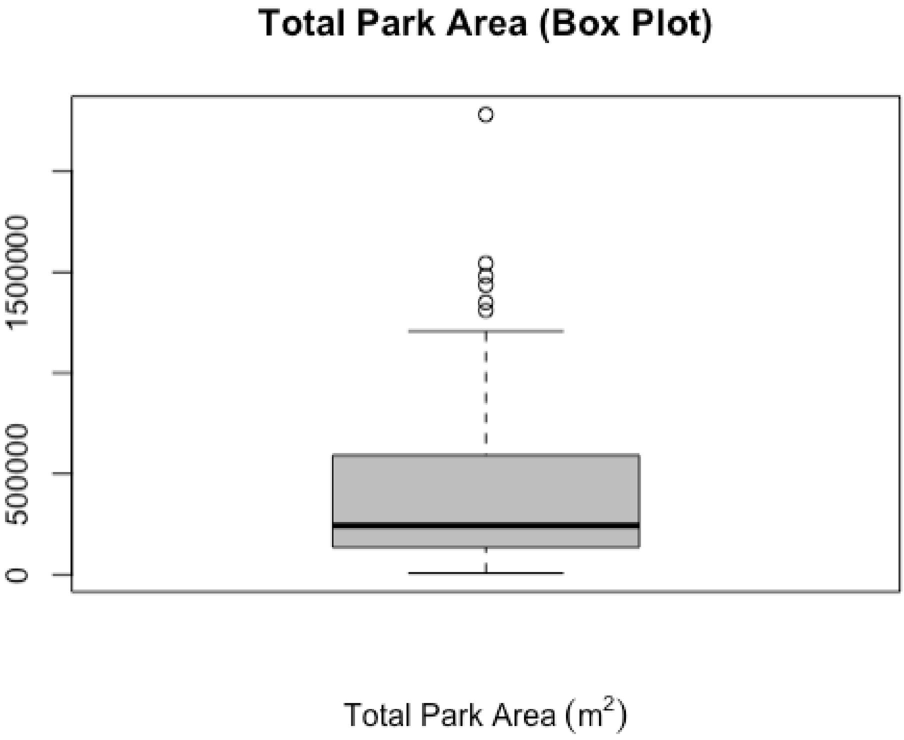

Figure 24. Box Plot for Total Park Area

Over 50 communities have less than 500000 m^2 of park area. A negative exponential relationship can be observed from the scatter plot. However, since the number of samples with large park areas is low, it requires more analysis when building the prediction model to see if there is a real correlation.

6.2.5 Number of Hospitals

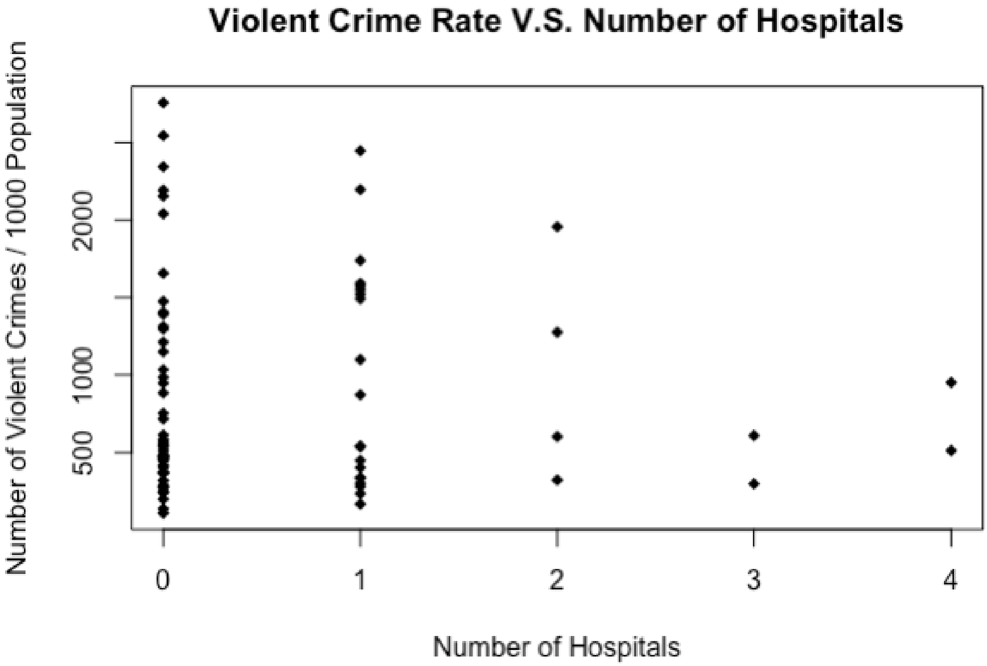

Figure 25. Scatter Plot for Number of Hospitals

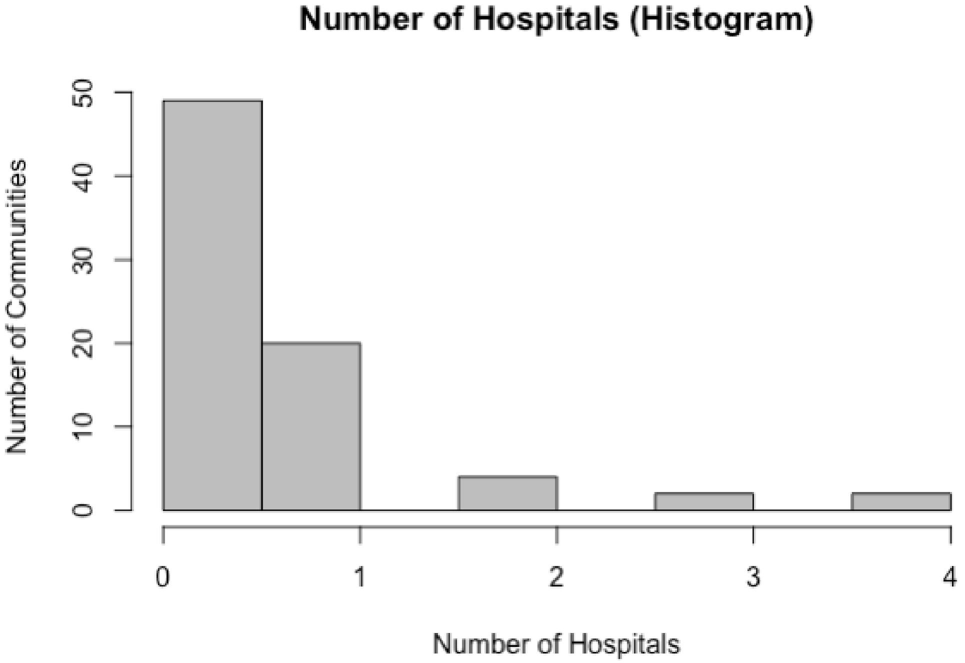

Figure 26. Histogram for Number of Hospitals

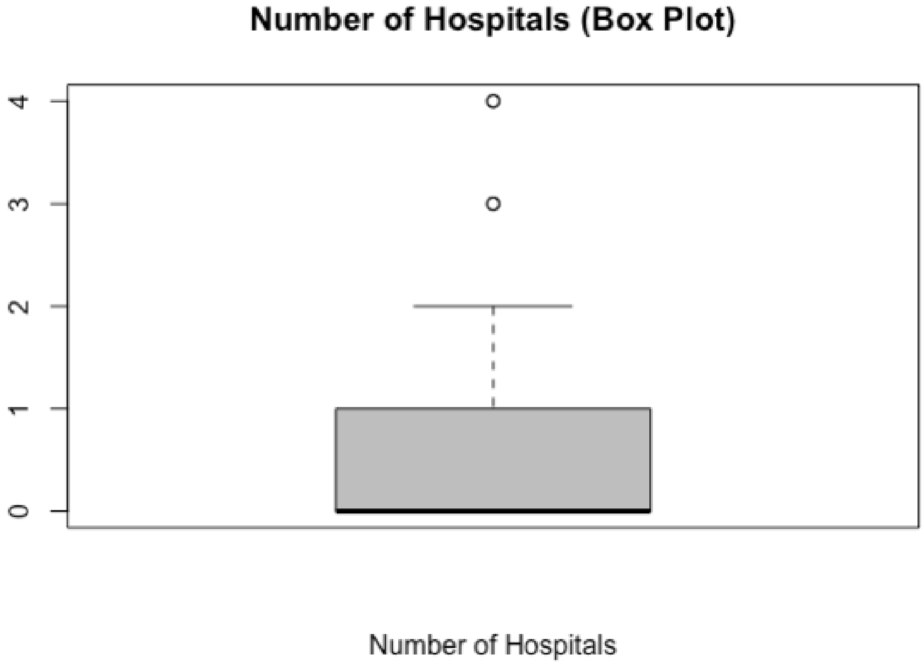

Figure 27. Box Plot for Number of Hospitals

As shown in all the above plots, a majority of the communities do not have hospitals. There are 4 outliers: Uptown and West Town with 3 hospitals, and Lincoln Square and New West Side with 4 hospitals.

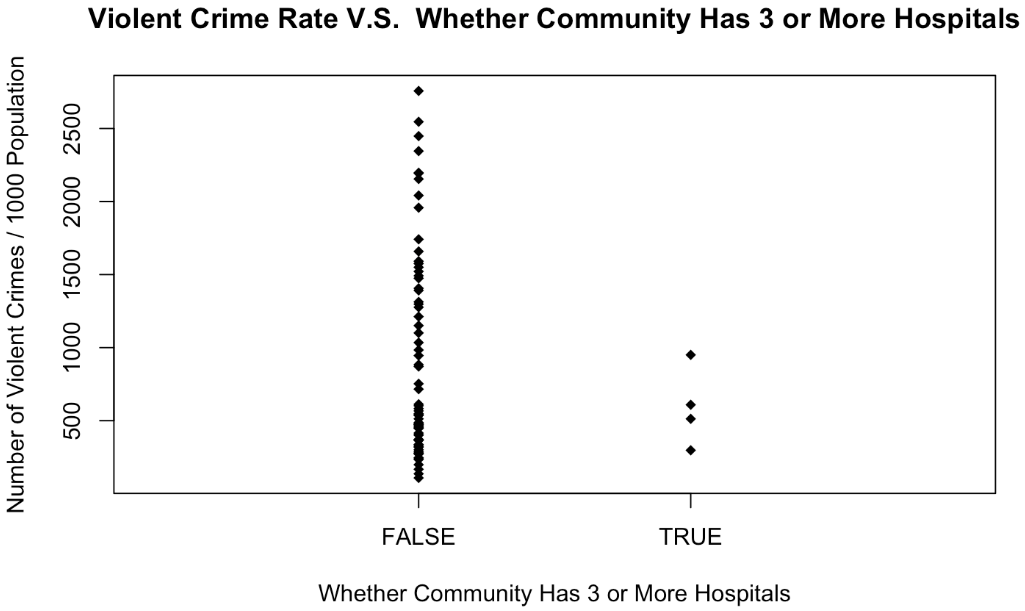

Figure 28. Scatter Plot for Whether Community Has 3 or More Hospitals

Another plot is made with a Boolean of whether the community has 3 or more hospitals as the x axis. All four communities which have 3 or 4 hospitals have lower than 1000 violent crimes per 1000 population.

6.2.6 Birth Rate by Teenage Mothers

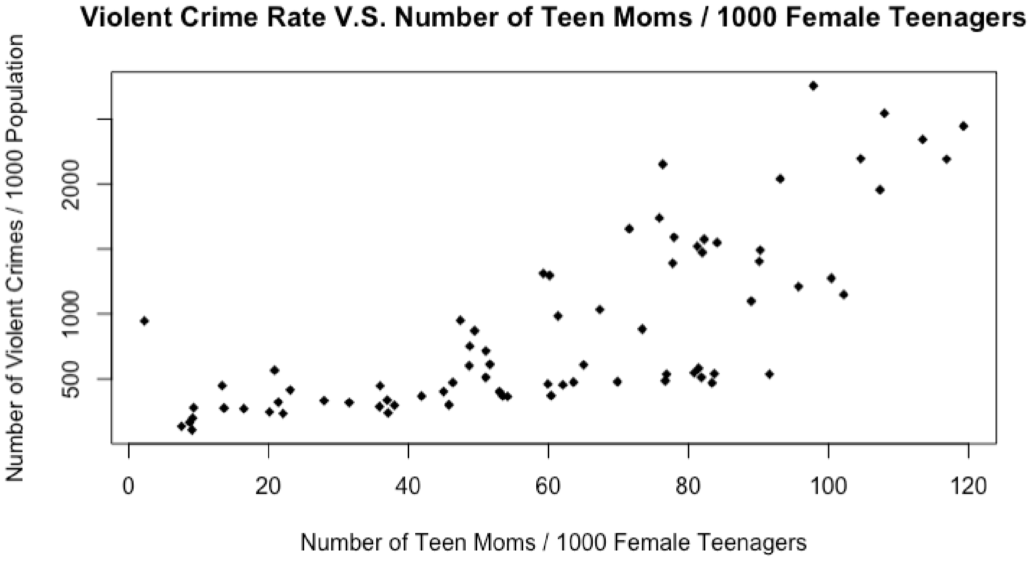

Figure 29. Scatter Plot for Birth Rate by Teenage Mothers

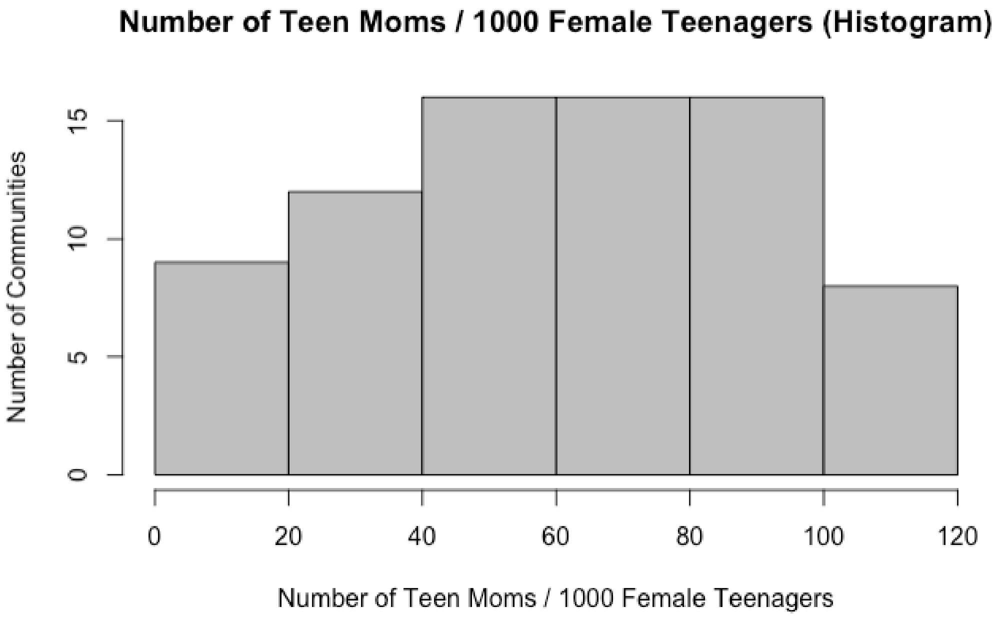

Figure 30. Histogram Plot for Birth Rate by Teenage Mothers

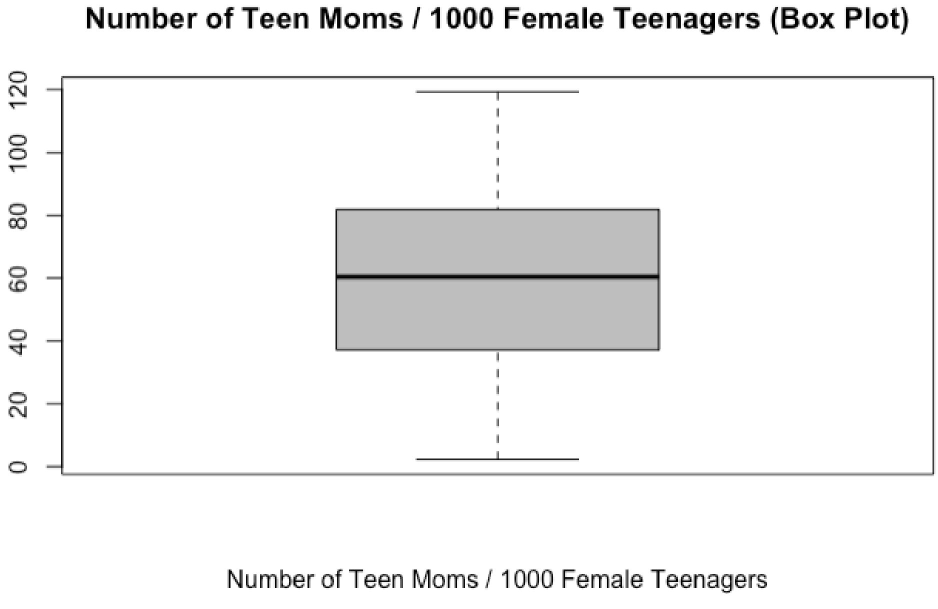

Figure 31. Box Plot for Birth Rate by Teenage Mothers

The number of teenage moms per 1000 female teenagers is also normally distributed in a 0 to 120 range. As indicated in the scatter plot, there is a clear positive relationship between the number of teenage moms and the number of violent crimes.

6.2.7 Infant Mortality Rate

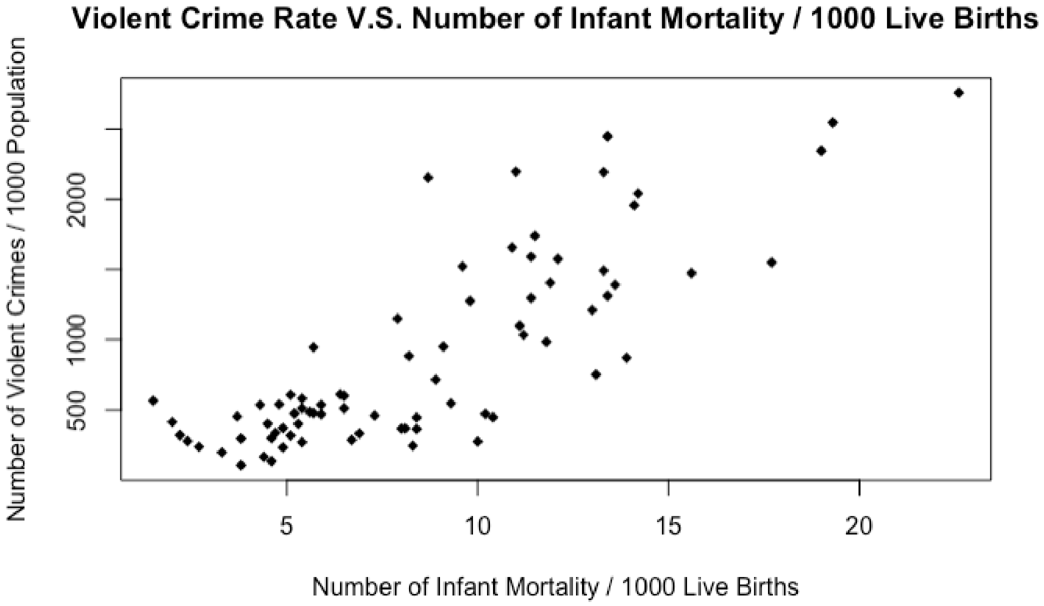

Figure 32. Scatter Plot for Infant Mortality Rate

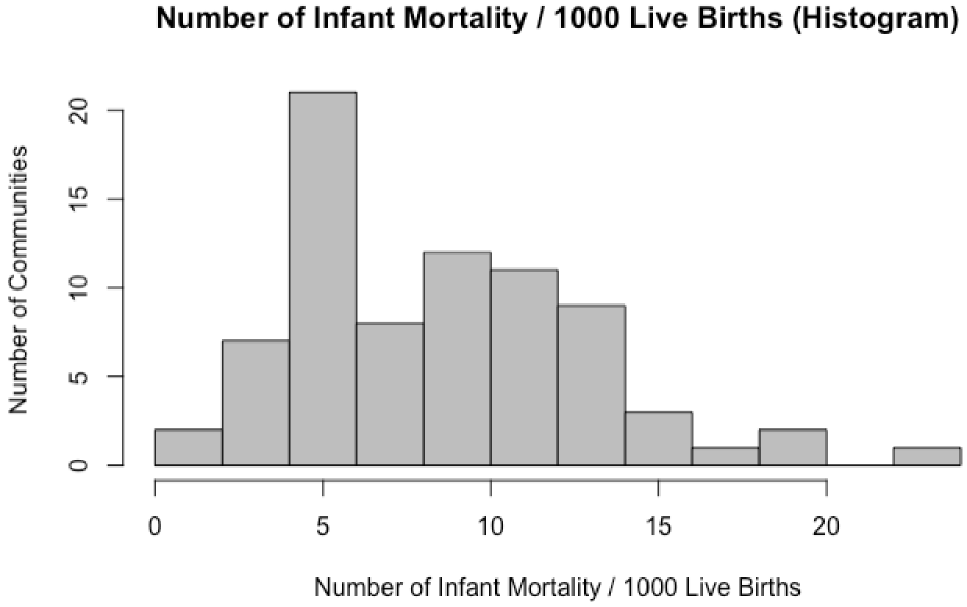

Figure 33. Histogram for Infant Mortality Rate

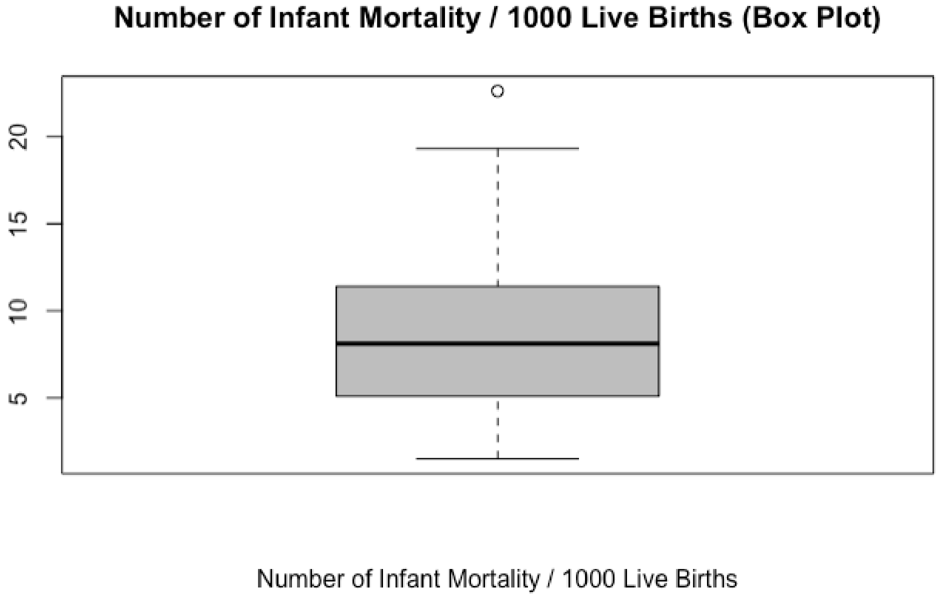

Figure 34. Box Plot for Infant Mortality Rate

From the scatter plot, it can be observed that there is a clear positive relationship between the number of infant mortalities per 1000 live births and the number of violent crimes per 1000 people. The only outlier indicated in the box plot is Fuller

Park, which has 22.6 infant mortalities per 1000 live births. This result matches what is observed from plotting the class variable.

6.2.8 Proportion of Hispanic People

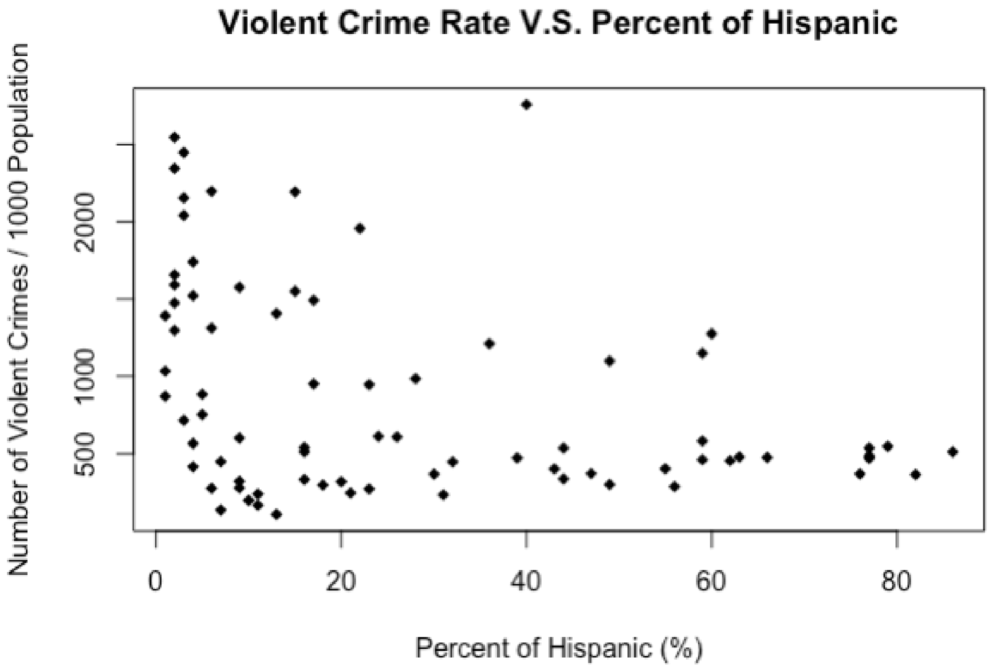

Figure 35. Scatter Plot for Proportion of Hispanic People

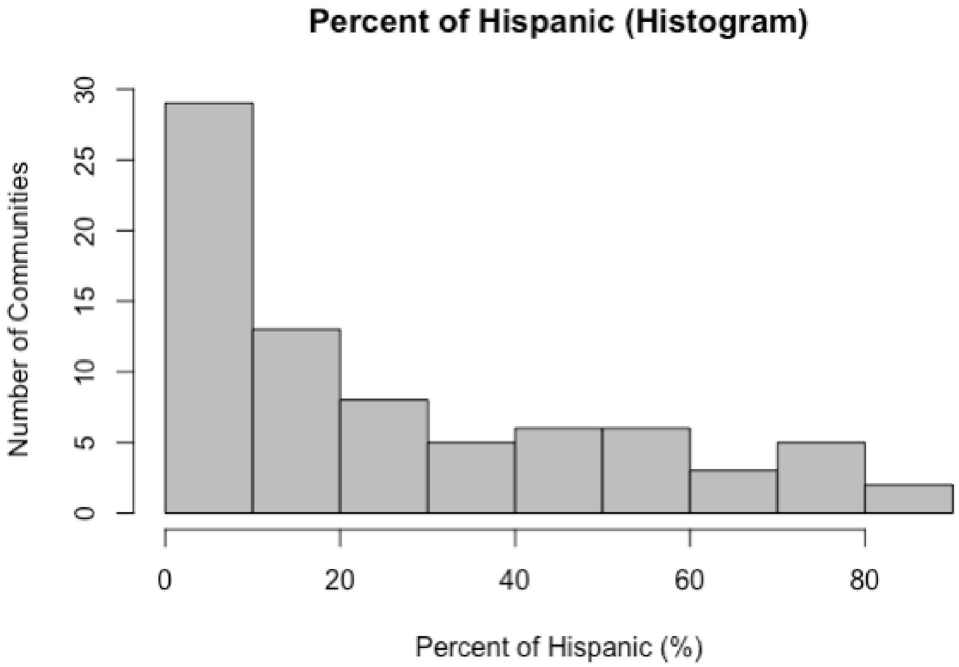

Figure 36. Histogram for Proportion of Hispanic People

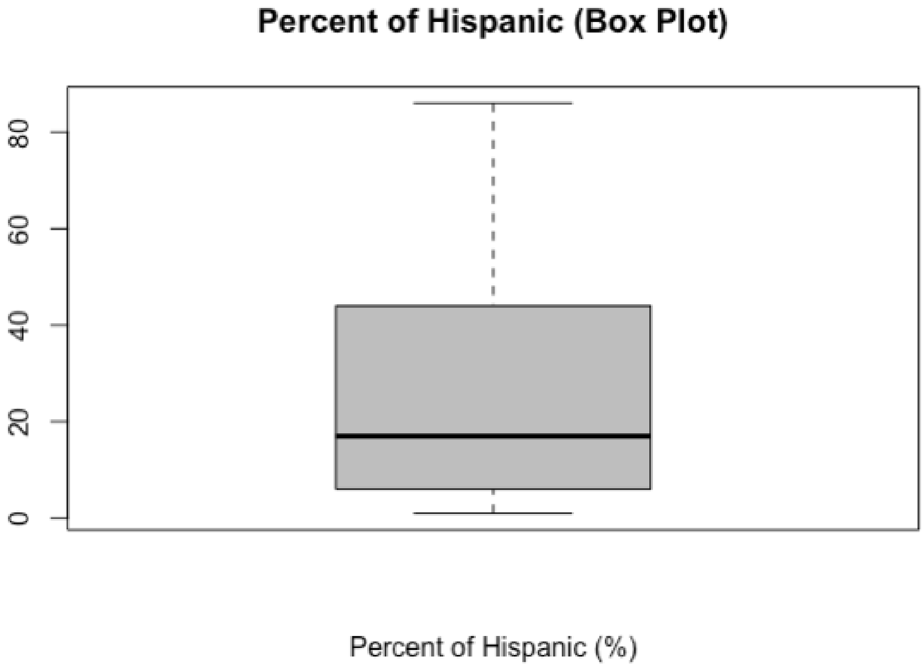

Figure 37. Box Plot for Proportion of Hispanic People

The violent crime rate exponentially decreases as the percentage of hispanic people increases from 0 to 90%, as shown in the scatter plot. The histogram also indicates that the distribution is not normal, but rather exponential as well.

6.2.9 Proportion of Black People

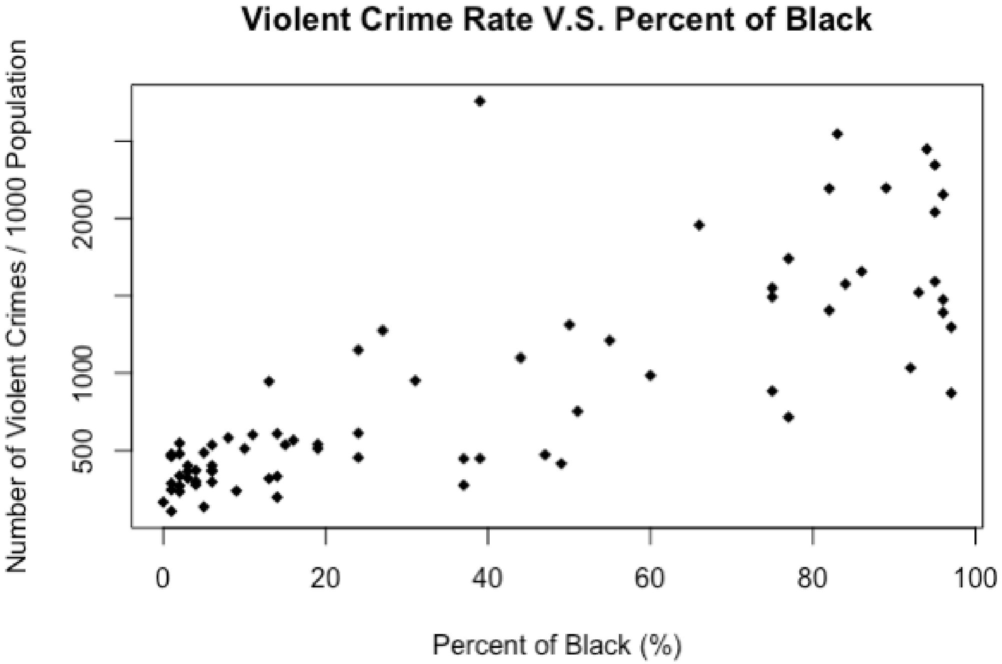

Figure 38. Scatter Plot for Proportion of Black People

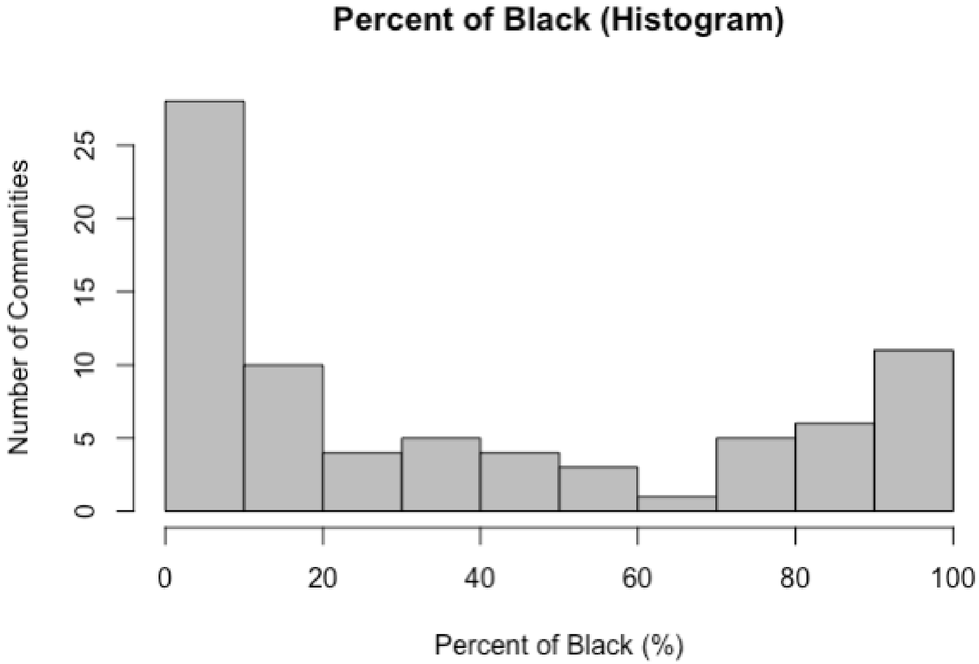

Figure 39. Histogram for Proportion of Black People

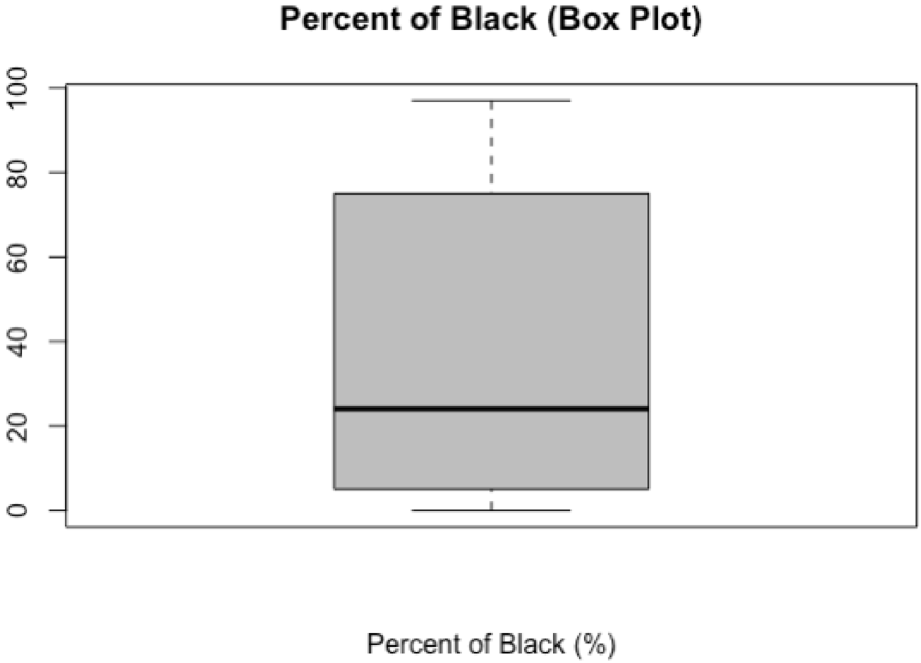

Figure 40. Box Plot for Proportion of Black People

The number of communities decreases and then increases as the percentage of black people increases from 0 to 100%. About 28 communities have 0 to 10% black people; about 10 communities have 10% - 20% and about 10 communities have 90% - 100%. A very low number of communities have 20% - 90% of black people. In addition, there is a clear positive relationship that can be observed with the class variable from the scatter plot.

6.2.10 Proportion of White People

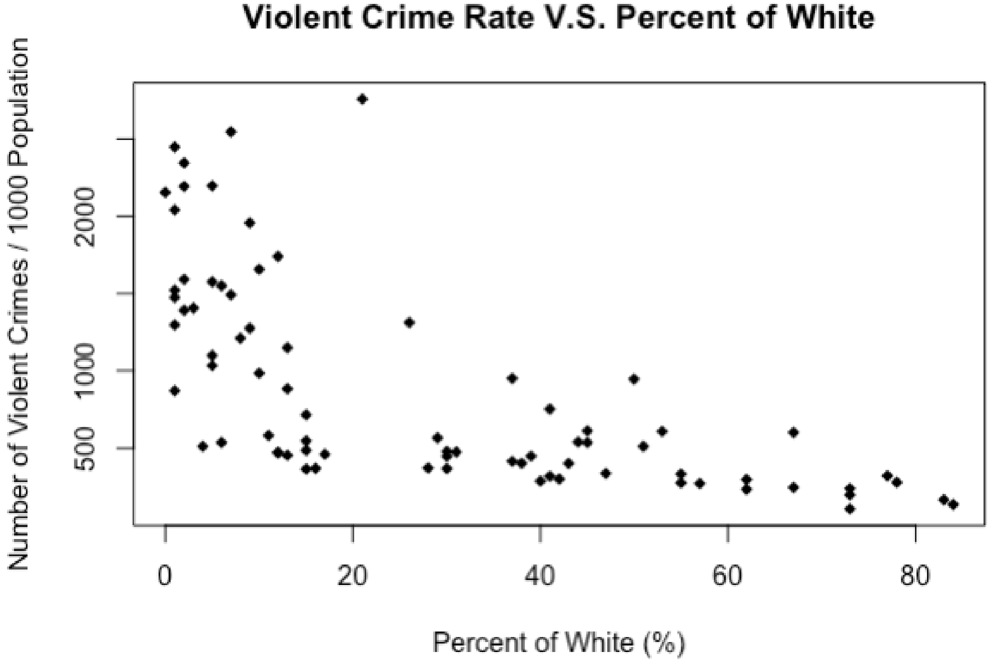

Figure 41. Scatter Plot for Proportion of White People

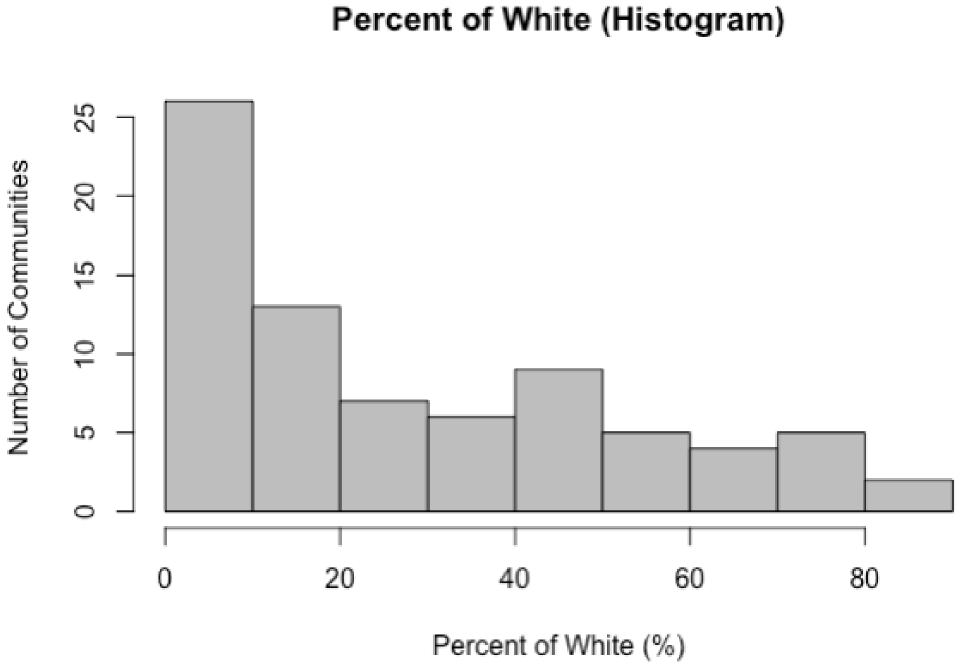

Figure 42. Histogram for Proportion of White People

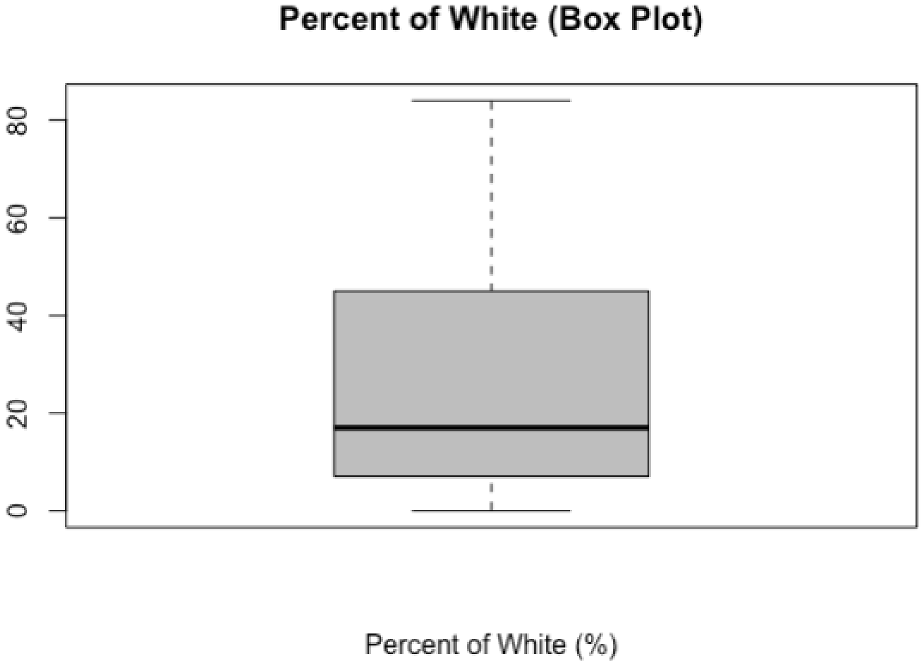

Figure 43. Box Plot for Proportion of White People

The distribution of the histogram is a exponential, in that the number of communities exponentially decreases as the percentage of white people increases. Furthermore, as shown in the scatter plot, there is a negative exponential relationship with the class variable.

6.2.11 Proportion of Asians

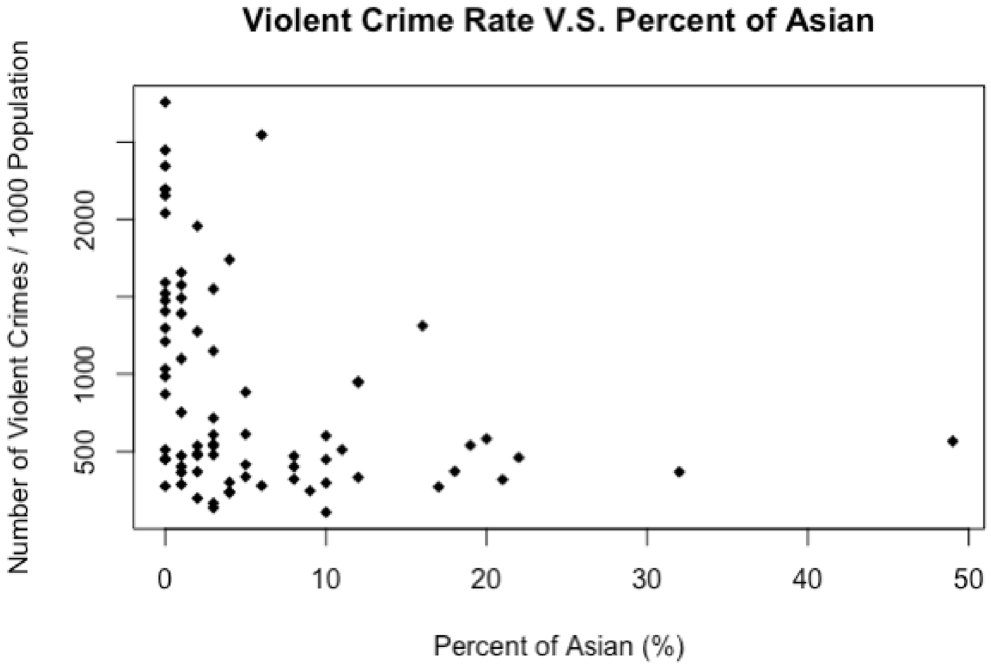

Figure 44. Scatter Plot for Proportion of Asian People

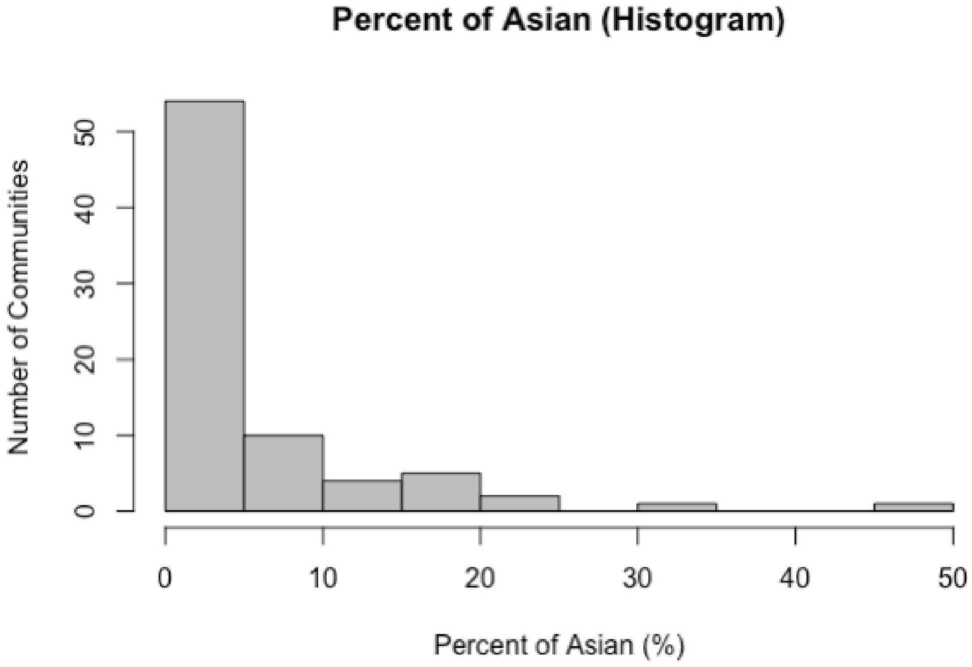

Figure 45. Histogram for Proportion of Asian People

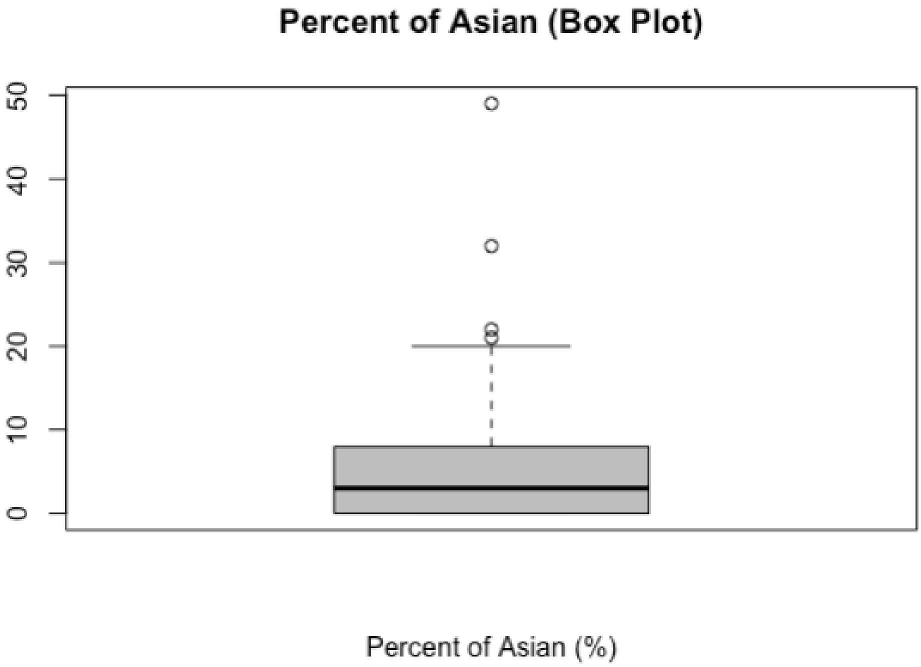

Figure 46. Box Plot for Proportion of Asian People

Similar to percentage of white people, the number of communities exponentially decreases as the percentage of Asian people increases from 0 to 50%. As shown in the scatter plot, there is a negative exponential relationship with the class variable. However, it is not clear due to the low number of samples with a high Asian percentage rate.

6.2.12 Proportion of Other Races

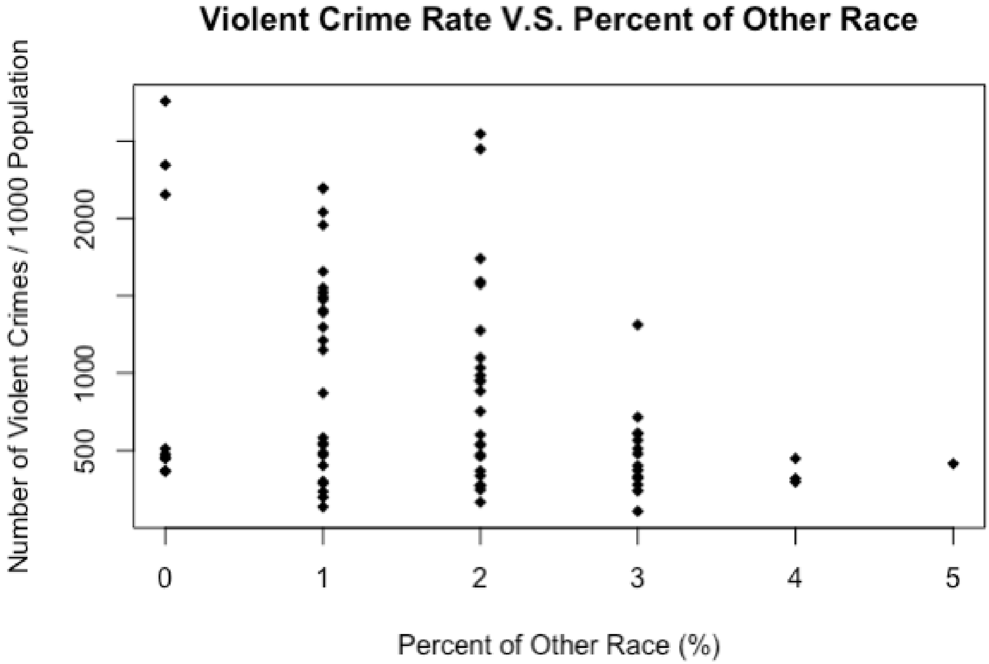

Figure 47. Scatter Plot for Proportion of Other Races

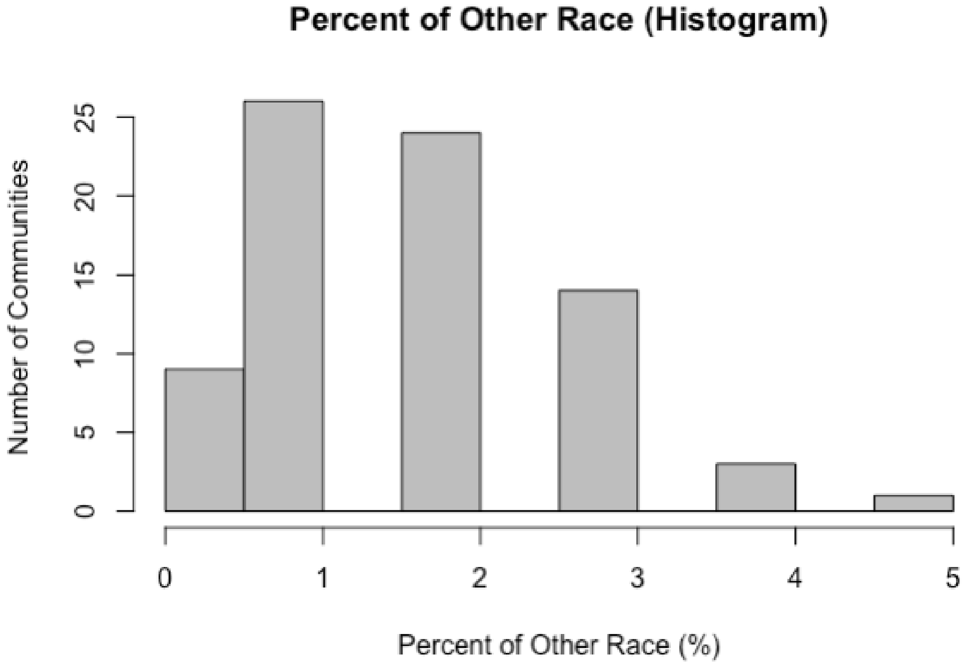

Figure 48. Histogram for Proportion of Other Races

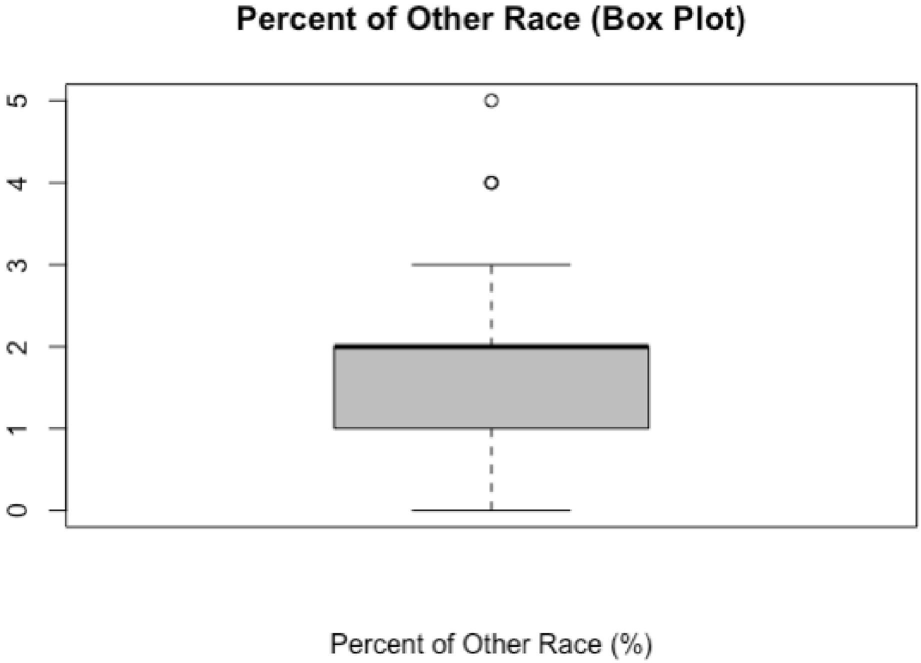

Figure 49. Box Plot for Proportion of Other Races

The percentage of other races does not seem indicate any strong correlation with the class variable. There are also very few communities with a percentage of people of “other races” of 4% or more.

6.2.13 Percent of Children in Poverty

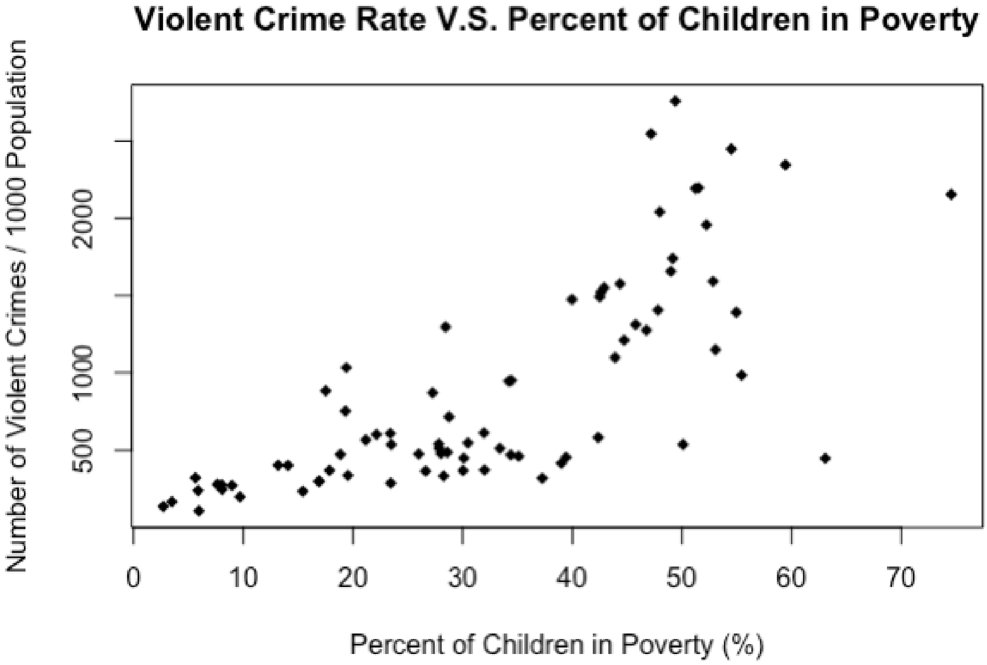

Figure 50. Scatter Plot for Percent of Children in Poverty

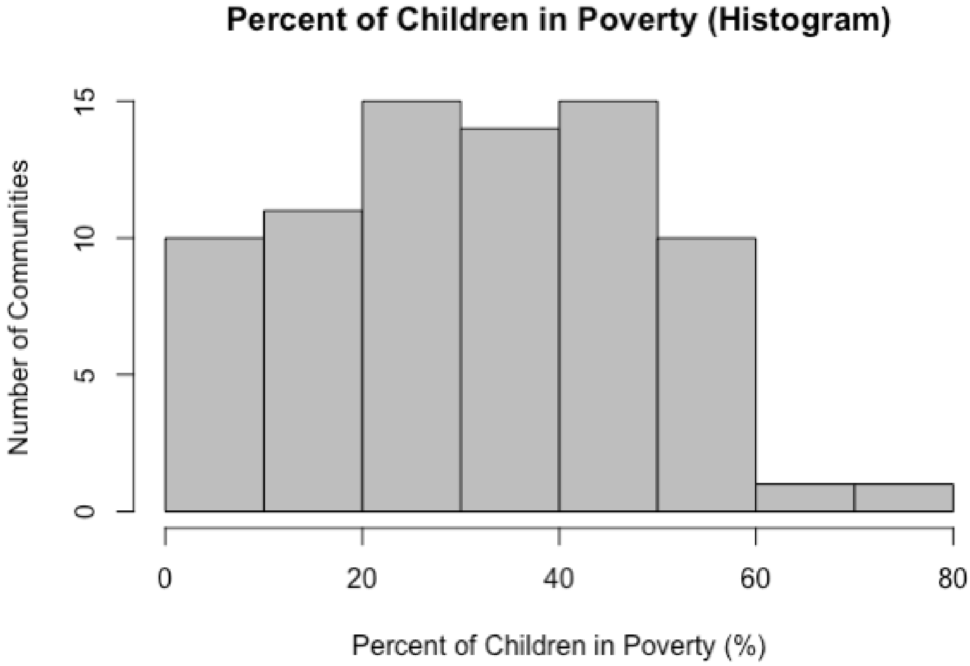

Figure 51. Histogram for Percent of Children in Poverty

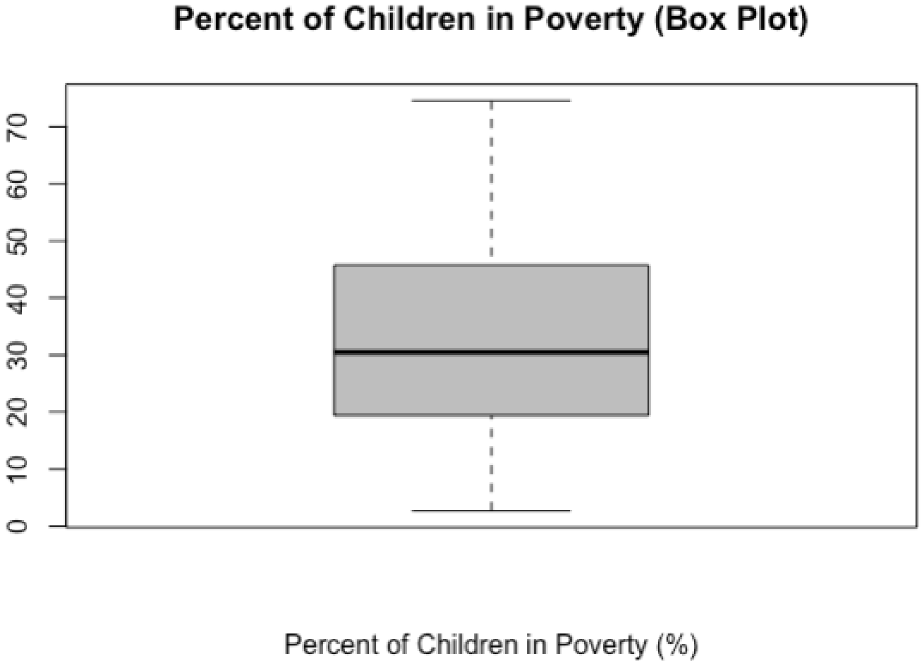

Figure 52. Box Plot for Percent of Children in Poverty

From the scatter plot, it can be observed that there is a clear positive relationship between the percentage of children in poverty and the rate of violent crimes. The percentage of children in poverty ranges from 0 to 80%, concentrated between the range of 0 to 60%.

6.3 Data Collection and Preprocessing Code

6.3.1 Class Variable

```
library(tidyr)
library(dplyr)
library(plyr)

# crime data from 2001
crime <- data.frame(crimes2001$X...ID,
                    crimes2001$Date,
                    crimes2001$Primary.Type,
                    crimes2001$Description,
                    crimes2001$Location.Description,
                    crimes2001$Community.Area)
names(crime) <- c('id', 'date', 'type', 'description', 'location', 'community')
crime <- crime[!(is.na(crime$community) | crime$community=='' |
crime$community=='0'), ]

# dividing offense involving children into violent vs non-violent
crime$type <- ifelse(grepl(('CRIM SEX ABUSE BY FAM MEMBER'), crime$description)
                   | grepl(('CHILD ABUSE'), crime$description)
                   | grepl(('AGG SEX ASSLT OF CHILD FAM MBR'), crime$description)
                   | grepl(('CHILD ABDUCTION'), crime$description)
                   | grepl(('AGG CRIM SEX ABUSE FAM MEMBER'), crime$description)
                   | grepl(('SEX ASSLT OF CHILD BY FAM MBR'), crime$description)
                   | grepl(('CRIM SEX ABUSE BY FAM MEMBER'), crime$description),
                     gsub('OFFENSE INVOLVING CHILDREN', 'VIOLENT OFFENSE INVOLVING
CHILDREN', crime$type),
                     gsub('OFFENSE INVOLVING CHILDREN', 'NON-VIOLENT OFFENSE
INVOLVING CHILDREN', crime$type))

# generating counts
commu_crime <- data.frame(crime$type, crime$community)
names(commu_crime) <- c('type', 'community')

count_commu_crime <- ddply(commu_crime, .(commu_crime$community, commu_crime$type),
nrow)
names(count_commu_crime) <- c("community", "type", "count")

count <- spread(count_commu_crime, key = type, value = count)
count[is.na(count)] <- 0

# population
population <- data.frame(population_chicago$GeogKey,
                         population_chicago$Geog,
                         population_chicago$`Total Population`)
names(population) <- c('community', 'community name', 'population(2010)')

# sum violent crimes and total crimes
names(sum_crime) <- c('community')
sum_crime$violent_crime <- NA
sum_crime$total_crime <- NA
sum_crime$violent_crime <- rowSums(count[, c('ASSAULT', 'BATTERY', 'CRIM SEXUAL
ASSAULT',
                                             'HOMICIDE', 'KIDNAPPING',
                                             'VIOLENT OFFENSE INVOLVING CHILDREN',
                                             'PUBLIC PEACE VIOLATION', 'RITUALISM',
                                             'ROBBERY', 'SEX OFFENSE', 'WEAPONS
VIOLATION')])
sum_crime$total_crime <- rowSums(count[, !(colnames(count) == "community")])

#merge sum_crime and population
sum_crime <- merge(x = sum_crime, y = population, by.x = 'community', by.y =
'community', all = TRUE)
sum_crime <- sum_crime[, c(1, 4, 5, 2, 3)]
```

6.3.2 Average School Rating

```
# Creating new dataframe for school info
schooldf <- data.frame(School_Profile_Information$School_ID,
                       School_Profile_Information$Short_Name,
                       School_Profile_Information$Overall_Rating,
                       School_Profile_Information$Rating_Status,
                       School_Locations$COMMAREA,
                       School_Locations$WARD_15)
community <- data.frame(census_data_by_community_area$communityAreaNumber,
                        census_data_by_community_area$Community)

names(schooldf) <- c('id', 'name', 'rating', 'status', 'community', 'ward')
names(community) <- c('number', 'name')

# Converting ratings to numbers
schooldf[, 'rating'] = toupper(schooldf[, 'rating'])
schooldf$rating_num <- NA
schooldf$rating_num[schooldf$rating=='INABILITY TO RATE'] <- -1
schooldf$rating_num[is.na(schooldf$rating)] <- -1
schooldf$rating_num[schooldf$rating=='LEVEL 3'] <- 1
schooldf$rating_num[schooldf$rating=='LEVEL 2'] <- 2
schooldf$rating_num[schooldf$rating=='LEVEL 2+'] <- 3
schooldf$rating_num[schooldf$rating=='LEVEL 1'] <- 4
schooldf$rating_num[schooldf$rating=='LEVEL 1+'] <- 5

# Calculating average rating by community area
avg <- aggregate(schooldf$rating_num, list(schooldf$community), mean)
names(avg) <- c('community', 'avg_rating')

# Add community number
community[, 'name'] = toupper(community[, 'name'])
community[, 'name'] <- gsub('[^[:alnum:][:space:]]', '', community[, 'name'])

# Join two data frames
avg_commu <- merge(x = community, y = avg, by.x = 'name', by.y = 'community', all
= TRUE)
avg_commu_ordered <- avg_commu[, c(2, 1, 3)]
avg_commu_ordered <- avg_commu_ordered[with(avg_commu_ordered, order(number)), ]
```

6.3.3 Average SSL Rating

```
# Create data frame for ssl and community
ssl <- data.frame(Strategic_Subject_List$`SSL SCORE`,
                  Strategic_Subject_List$`COMMUNITY AREA`)
community <- data.frame(census_data_by_community_area$communityAreaNumber,
                        census_data_by_community_area$Community)
names(ssl) <- c('score', 'community')
names(community) <- c('number', 'name')

# Eliminate rows with blank community
ssl <- ssl[!(is.na(ssl$community) | ssl$community==''), ]

# Standardize community names
ssl[, 'community'] = toupper(ssl[, 'community'])
ssl[, 'community'] <- gsub('[^[:alnum:][:space:]]', '', ssl[, 'community'])

# Calculate the average
avg <- aggregate(ssl$score, list(ssl$community), mean)
names(avg) <- c('community', 'avg_rating')

# Add community number
community[, 'name'] = toupper(community[, 'name'])
community[, 'name'] <- gsub('[^[:alnum:][:space:]]', '', community[, 'name'])

# Join two data frames
avg_commu <- merge(x = community, y = avg, by.x = 'name', by.y = 'community', all
= TRUE)
avg_commu_ordered <- avg_commu[, c(2, 1, 3)]
avg_commu_ordered <- avg_commu_ordered[with(avg_commu_ordered, order(number)), ]
```

6.3.4 Total Park Area

```
##########################################
#parks by community area
##########################################
library(rgdal)
library(sp)
library(dplyr)
library(sf)
library(tidyverse)
library(raster)

#import the shape files
chicagoparks <- readOGR('4A/MSCI 446/R/dataToPreprocess/chicagoparksshapefile',
'geo_export_287c1e81-adfc-4076-bbd4-7ac4b1ca62c2')
chicagocommunityareas <- readOGR('4A/MSCI
446/R/dataToPreprocess/communityareashapefile', 'geo_export_f2c553e7-eb62-4773-
9655-8037a1bdd109', stringsAsFactors = FALSE)

#RUN THIS CODE FOR TOTAL PARK AREA FOR EACH COMMUNITY AREA#
totalParkAreaForCommunityAreas <- rep(0, nrow(chicagocommunityareas))

for(i in 1:nrow(chicagocommunityareas)) {
  totalArea <- 0
  for (j in 1:nrow(chicagoparks)) {
    #get intersection of community area & park
    intersect <- intersect(chicagocommunityareas[i, ], chicagoparks[j, ])
    if (!is.null(intersect)) {
      #if intersection!=null, add to totalArea for current community area
      totalArea <- totalArea + area(intersect)
    }
  }
  totalParkAreaForCommunityAreas[i] <- totalArea
}

#create the dataframe consisting of three columns: communityArea,
communityAreaNumber, and totalParkArea
library(readxl)
censusdata <- read_excel("4A/MSCI 446/R/dataToPreprocess/Census-Data-by-Chicago-
Community-Area-2017 (2).xlsx")
censusdata <- data.frame(censusdata$Community, censusdata$CommunityAreaNumber)
names(censusdata) <- c('Community', 'communityAreaNumber')
censusdata$Community <- toupper(censusdata$Community)
chicagocommunityareas@data$community[75] <- 'O\'HARE' #naming difference

communityAreaNumber <- rep(0, nrow(chicagocommunityareas))

#Need to match totalParkAreaForCommunityAreas to each communityAreaNumber
for(i in 1:nrow(chicagocommunityareas)) {
  communityAreaNumber[i] <-
censusdata[which(censusdata$Community==chicagocommunityareas@data$community[i]),2]
}

totalParkAreaDF <- data.frame(chicagocommunityareas@data$community,
communityAreaNumber, totalParkAreaForCommunityAreas)
names(totalParkAreaDF) <- c('Community', 'communityAreaNumber', 'totalParkArea')
#save the totalParkArea by Community Area Number
write.csv(totalParkAreaDF, 'totalParkAreaByCommunityArea.csv')
```

6.3.5 Number of Hospitals, Teen Mom Birth Rate, Infant Mortality Rate

```
##########################################
#Public Safety Data
##########################################
library(rgdal)
library(sp)
library(dplyr)
library(sf)
library(tidyverse)
library(raster)
```

```
#Code for numHospitalsPerCommunityArea################################
hospitals <- readOGR('4A/MSCI 446/R/dataToPreprocess/Hospitals', 'Hospitals',
stringsAsFactors = FALSE)
numHospitalsPerCommunityArea <- as.data.frame(table(hospitals@data$AREA_NUMBE))
names(numHospitalsPerCommunityArea) <- c('communityAreaNum', 'numHospitals')
numHospitalsPerCommunityArea$communityAreaNum <-
as.numeric(levels(numHospitalsPerCommunityArea$communityAreaNum))
for (i in 1:77) {
  if (sum(numHospitalsPerCommunityArea$communityAreaNum == i) == 0) {
    newDF <- data.frame(i,0)
    names(newDF)<-c('communityAreaNum', 'numHospitals')
    numHospitalsPerCommunityArea <- rbind(numHospitalsPerCommunityArea, newDF)
  }
}
numHospitalsPerCommunityArea <-
numHospitalsPerCommunityArea[order(numHospitalsPerCommunityArea$communityAreaNum),
]
var(numHospitalsPerCommunityArea$numHospitals)

#Code for teenMomRatePerCommunityArea################################
teenMomsData <- read.csv('4A/MSCI
446/R/dataToPreprocess/Public_Health_Statistics_-_Births_to_mothers_aged_15-
19_years_old_in_Chicago__by_year__1999-2009.csv')
teenBirthRates <- data.frame(teenMomsData$Teen.Birth.Rate.1999,
                               teenMomsData$Teen.Birth.Rate..2000,
                               teenMomsData$Teen.Birth.Rate.2001,
                               teenMomsData$Teen.Birth.Rate.2002,
                               teenMomsData$Teen.Birth.Rate.2003,
                               teenMomsData$Teen.Birth.Rate.2004,
                               teenMomsData$Teen.Birth.Rate.2005,
                               teenMomsData$Teen.Birth.Rate.2006,
                               teenMomsData$Teen.Birth.Rate.2007,
                               teenMomsData$Teen.Birth.Rate.2008,
                               teenMomsData$Teen.Birth.Rate.2009)
teenBirthRates <- teenBirthRates[1:nrow(teenBirthRates)-1,]
teenBirthRatesTransposed <- t(teenBirthRates)
rownames(teenBirthRatesTransposed) <- NULL
colnames(teenBirthRatesTransposed) = seq(1:77)
#find the mean teenMomBirthRate for years 1999-2009. Use this as each community
area's "teenMomBirthRate"
teenMomRatePerCommunityAreaVec=c()
for(i in 1:ncol(teenBirthRatesTransposed)){
  teenMomRatePerCommunityAreaVec[i] = mean(teenBirthRatesTransposed[,i], na.rm =
FALSE)
}

teenMomRatePerCommunityArea <- data.frame(teenMomsData[1:nrow(teenMomsData)-1,1],
teenMomRatePerCommunityAreaVec)
names(teenMomRatePerCommunityArea) <- c('communityAreaNum', 'teenMomRate')

#Code for infantMortalityRatePerCommunityArea################################
infantMortalityData <- read.csv('4A/MSCI
446/R/dataToPreprocess/Public_Health_Statistics-
_Infant_mortality_in_Chicago__2005__2009.csv')
infantMortalityData <- infantMortalityData[1:nrow(infantMortalityData)-1,]
infantMortalityRatePerCommunityArea <-
data.frame(infantMortalityData$ ,aerA.ytinummoC‿
infantMortalityData$Average.Infant.Mortality.Rate.2005...2009)
names(infantMortalityRatePerCommunityArea) <- c('communityAreaNum',
'infantMortalityRate')
remove(infantMortalityData)

#write all three to csv
publicHealthData <-
data.frame(infantMortalityRatePerCommunityArea$communityAreaNum,
           numHospitalsPerCommunityArea$numHospitals,
           teenMomRatePerCommunityArea$teenMomRate,
           infantMortalityRatePerCommunityArea$infantMortalityRate)
names(publicHealthData) <- c('communityAreaNum',
                              'numHospitals',
                              'teenMomRate',
                             'infantMortalityRate')
write.csv(publicHealthData, 'publicHealth.csv')
```

6.3.6 Proportion of Different Races, and Percent of Children in Poverty

```
##########################################
#poverty & race by community area
##########################################
#did most of the conversion in excel, and using R to just create a csv of it.
library(readxl)
censusdata <- read_excel("4A/MSCI 446/R/dataToPreprocess/Census-Data-by-Chicago-
Community-Area-2017 (2).xlsx")
censusdata <- data.frame(censusdata$Community,
                         censusdata$CommunityAreaNumber,
                         censusdata$Hispanic,
                         censusdata$Black,
                         censusdata$White,
                         censusdata$Asian,
                         censusdata$Other,
                         censusdata$PercentChildrenInPoverty)
names(censusdata) <- c('Community', 'communityAreaNumber', 'Hispanic', 'Black',
'White', 'Asian', 'Other', 'PercentChildrenInPoverty')
write.csv(censusdata, 'censusdataByCommunityArea.csv')
```

6.3.7 Combining all Datasets into One

```
##########################################
#Combining all the data
##########################################

avgSchoolRating <- read.csv("4A/MSCI
446/R/explanatoryvariables/avg_school_rating_by_community.csv")
avgSSLscore <- read.csv("4A/MSCI
446/R/explanatoryvariables/avg_ssl_score_by_community.csv")
censusData <- read.csv("4A/MSCI
446/R/explanatoryvariables/censusdataByCommunityArea.csv")
typesOfCrimes <- read.csv("4A/MSCI
446/R/explanatoryvariables/crime_count_in_community.csv")
predictedVarDF <- read.csv("4A/MSCI
446/R/explanatoryvariables/total_crime_by_community.csv")
totalParkArea <- read.csv("4A/MSCI
446/R/explanatoryvariables/totalParkAreaByCommunityArea.csv")
publicHealth <- read.csv("4A/MSCI 446/R/explanatoryvariables/publicHealth.csv")

#because totalParkArea dataframe is not sorted by ascending community area number:
totalParkArea <- totalParkArea[order(totalParkArea$communityAreaNumber),]

predTable <- data.frame(totalParkArea$Community,
                        totalParkArea$communityAreaNumber,
                        predictedVarDF$violent_crime * 1000 /
(predictedVarDF$population.2010.),
                        avgSchoolRating$avg_rating,
                        avgSSLscore$avg_rating,
                        totalParkArea$totalParkArea,
                        publicHealth$numHospitals,
                        publicHealth$teenMomRate,
                        publicHealth$infantMortalityRate,
                        censusData[,4:ncol(censusData)])

names(predTable) <- c(
  "community",
  "communityAreaNum",
  "percentViolentCrimePer1000Population",
  "avgSchoolRating",
  "avgSSLRating",
  "totalParkArea",
  "numHospitals",
  "teenMomRate",
  "infantMortalityRate",
  "hispanic",
  "black",
  "white",
  "asian",
  "other",
  "percentChildrenInPov")
)

#write to csv
write.csv(predTable, 'predTable.csv')
```

6.4 Explanatory Data Analysis Code

```
# gather useful columns
explanatory <- data.frame(predTable$communityAreaNum,
                          predTable$percentViolentCrimePer1000Population,
                          predTable$avgSchoolRating,
                          predTable$avgSSLRating,
                          predTable$totalParkArea,
                          predTable$numHospitals,
                          predTable$teenMomRate,
                          predTable$infantMortalityRate,
                          100*predTable$hispanic,
                          100*predTable$black,
                          100*predTable$white,
                          100*predTable$asian,
                          100*predTable$other,
                          100*predTable$percentChildrenInPov)
names(explanatory) <- c('community',
'number_of_violent_crimes_per_1000_population',
                        'Average_School_Rating', 'Normalized_Average_SSL',
'Total_Park_Area_(m2)',
                        'Number_of_Hospitals',
'Number_of_Teen_Moms_/_1000_Female_Teenagers',
'Number_of_Infant_Mortality_/_1000_Live_Births',
                        'Percent_of_Hispanic_(%)', 'Percent_of_Black_(%)',
'Percent_of_White_(%)',
                        'Percent_of_Asian_(%)', 'Percent_of_Other_Race_(%)',
'Percent_of_Children_in_Poverty_(%)')

# normalize SSL (266.0711 - 304.1068)
explanatory$Normalized_Average_SSL <- (explanatory$Normalized_Average_SSL-
min(explanatory$Normalized_Average_SSL))/(max(explanatory$Normalized_Average_SSL)
- min(explanatory$Normalized_Average_SSL))

# num_hospital to binary
explanatory$Whether_Community_Has_3_or_More_Hospitals <- NA
explanatory$Whether_Community_Has_3_or_More_Hospitals <-
explanatory$Number_of_Hospitals >= 3

col_names <- colnames(explanatory)

# scatter plot
for(i in 3:14) {
  plot(explanatory[,i], explanatory$number_of_violent_crimes_per_1000_population,
       main=paste('Violent Crime Rate V.S.', gsub( '\\s*\\([^\\)]+\\)', '',
gsub('_', ' ', col_names[i]))),
       xlab=gsub('_', ' ', col_names[i]), ylab='Number of Violent Crimes / 1000
Population', pch=18)
}

plot(explanatory[,15], explanatory$number_of_violent_crimes_per_1000_population,
     main=paste('Violent Crime Rate V.S. ', gsub( '\\s*\\([^\\)]+\\)', '',
gsub('_', ' ', col_names[15]))),
     xaxt='n', xlim=c(-1,2), xlab=gsub('_', ' ', col_names[15]), ylab='Number of
Violent Crimes / 1000 Population', pch=18)
axis(1, at=0:1, labels=c('FALSE', 'TRUE'))

# histograms
hist(explanatory$number_of_violent_crimes_per_1000_population,
     col='grey',
     main='Number of Violent Crimes / 1000 Population (Histogram)',
     xlab='Number of Violent Crimes / 1000 Population', ylab='Number of
Communities')

for(i in 3:14) {
  hist(explanatory[,i],
       col='grey',
       main=paste(gsub( '\\s*\\([^\\)]+\\)', '', gsub('_', ' ', col_names[i])),
'(Histogram)'),
       xlab=gsub('_', ' ', col_names[i]), ylab='Number of Communities')
}

# box plots
boxplot(explanatory$number_of_violent_crimes_per_1000_population, data=explanatory,
        col='grey',
        main="Number of Violent Crimes / 1000 Population (Box Plot)",
        xlab="Number of Violent Crimes / 1000 Population")

for(i in 3:14) {
  boxplot(explanatory[,i], data=explanatory,
          col='grey',
          main=paste(gsub( '\\s*\\([^\\)]+\\)', '', gsub('_', ' ', col_names[i])),
'(Box Plot)'),
          xlab=gsub('_', ' ', col_names[i]))
}
```

6.5 Numeric Regression Code

```
##########################################
#Numeric Regression
##########################################
library(caret)
library(dplyr)

#Import data: includes explanatory variables AND class variable but also other
columns (e.g. community area name)
data <- read.csv("4A/MSCI 446/R/explanatoryvariables/predTable.csv")
#remove extraneous columns (e.g. community area name)
dataForPred <- dplyr::select(data, -X, -community, -communityAreaNum)
dataForPred <- dataForPred[,1:13]
dataForPred$numHospitals <- ifelse(dataForPred$numHospitals >= 3, 1, 0)
colnames(dataForPred)[5] <- "has3OrMoreHospitals"

#Use 10-fold cross-validation for getting alpha/lambda values for glmnet
tControlObj <- caret::trainControl(
  method = "cv", number = 10,
  verboseIter = TRUE,
  summaryFunction = defaultSummary
)

k <- 77
#leave-1-out cross validation for performance metrics (RMSE, Rsquared, MAE)
splitPlan <- kWayCrossValidation(nrow(dataForPred), k, NULL, NULL)
#initialization of the dataframe that will store the performance metrics
metricsDF <- as.data.frame(matrix(nrow = 3, ncol = 4))
names(metricsDF) <- c("Model", "RMSE", "Rsquared", "MAE")
```

6.5.1 OLS Linear Regression

```
##########################################
#train using linear regression#
modelLM <- train(
  x = dataForPred[,2:13],
  y = dataForPred[,1],
  method = "lm",
  trControl = tControlObj
)
```

```
#Predicted vs Actual Plot
plot(modelLM$finalModel$fitted.values, dataForPred[,1], main='Predicted vs Actual
for Simple Linear Regression', xlab='Predicted', ylab='Actual')
#Residual Plot
plot(modelLM$finalModel$fitted.values, modelLM$finalModel$residuals,
main='Residual Plot for Simple Linear Regression', xlab='Predicted',
ylab='Residuals')
abline(h = 0, col = "darkgrey", lty = 2)
#Residual Histogram Plot
hist(modelLM$finalModel$residuals,
     col='grey',
     main='Residual Histogram for Simple Linear Regression',
     xlab='Residual', ylab='Frequency')

#Leave-1-out Cross Validation to get OLS Performance Metrics
lmPredValues <- data.frame("predicted" = rep(0, nrow(dataForPred)))
for(i in 1:k) {
  split <- splitPlan[[i]]
  model <- lm(percentViolentCrimePer1000Population ~ ., data =
dataForPred[split$train,])
  lmPredValues$predicted[split$app] <- predict(model, newdata =
dataForPred[split$app,])
}
metricsDF[1,] <- c("Linear Regression CV", postResample(lmPredValues$predicted,
dataForPred[,1]))
metricsDF[4,] <- c("Linear Regression", postResample(predict(modelLM,
dataForPred[2:13]), dataForPred[,1]))
```

6.5.2 Elastic Net Regression

```
##########################################
#train using glmnet#
modelGLMNET <- train(
  x = dataForPred[,2:13],
  y = dataForPred[,1],
  method = "glmnet",
  metric = "RMSE",
  tuneGrid = expand.grid(alpha = 0:10/10), #lambda = seq(0.0001, 1, length = 20)
  trControl = tControlObj
)
#obtain the predicted values
predictionGLMNET <- predict(modelGLMNET, dataForPred[, 2:13])

#plots RMSE over different alpha and lambda values.
plot(modelGLMNET,  main='Alpha and Lambda Values for GLMNET')
#Predicted vs Actual Plot
plot(predictionGLMNET, dataForPred[,1], main='Predicted vs Actual for GLMNET
Regression', xlab='Predicted', ylab='Actual')
#Residual Plot
plot(predictionGLMNET, (dataForPred[,1]-predictionGLMNET), main='Residual Plot for
GLMNET Regression', xlab='Predicted', ylab='Residuals')
abline(h = 0, col = "darkgrey", lty = 2)
#Histogram Plot
hist((dataForPred[,1]-predictionGLMNET),
     col='grey',
     main='Residual Histogram for GLMNET Linear Regression',
     xlab='Residual', ylab='Frequency')

#Leave-1-Out Cross Validation to get performance metrics
glmnetPredValues <- data.frame("predicted" = rep(0, nrow(dataForPred)))
for(i in 1:k) {
  split <- splitPlan[[i]]
  model <- glmnet(as.matrix(dataForPred[split$train,2:13]),
dataForPred[split$train,1], alpha = modelGLMNET$bestTune$alpha, lambda =
modelGLMNET$bestTune$lambda)
  glmnetPredValues$predicted[split$app] <- predict(model, s =
modelGLMNET$bestTune$lambda, newx = as.matrix(dataForPred[split$app,2:13]))
}
metricsDF[2,] <- c("Elastic Net CV", postResample(glmnetPredValues$predicted,
dataForPred[,1]))
```

```
metricsDF[5,] <- c("Elastic Net", postResample(predictionGLMNET, dataForPred[,1]))
```

6.5.3 GAM Regression

```
##########################################
#GAM model#
#since caret can only do standard GAM model of y = s(x1) + s(x2) + etc. we will
not be using caret
library(mgcv)
library(vtreat)

#GAM formula, based off scatter plots of each explanatory variable vs class
variable from EDA
GAMformula <- percentViolentCrimePer1000Population ~
                                                  avgSchoolRating +
                                                  avgSSLRating +
                                                  s(totalParkArea) +
                                                  has3OrMoreHospitals +
                                                  s(teenMomRate) +
                                                  s(infantMortalityRate) +
                                                  s(hispanic) +
                                                  black +
                                                  s(white) +
                                                  s(asian) +
                                                  other +
                                                  s(percentChildrenInPov)

#Leave-1-Out Cross Validation To get Performance Metrics
gamPredValues <- data.frame("predicted" = rep(0, nrow(dataForPred)))
for(i in 1:k) {
  split <- splitPlan[[i]]
  model <- gam(GAMformula, data = dataForPred[split$train,], family = gaussian)
  gamPredValues$predicted[split$app] <- predict(model, newdata =
dataForPred[split$app,])
}
metricsDF[3,] <- c("GAM cv", postResample(gamPredValues$predicted,
dataForPred[,1]))

#Building the final model
gamModel <- gam(GAMformula, data = dataForPred, family = gaussian)
finalPredictions <- predict(gamModel, dataForPred[, 2:13])
metricsDF[6,] <- c("GAM", postResample(finalPredictions, dataForPred[,1]))

#Predicted vs Actual Plot
plot(finalPredictions, dataForPred[,1], main='Predicted vs Actual for GAM
Regression', xlab='Predicted', ylab='Actual')
#Residual Plot
plot(finalPredictions, (dataForPred[,1]-finalPredictions), main='Residual Plot for
GAM Regression', xlab='Predicted', ylab='Residuals')
abline(h = 0, col = "darkgrey", lty = 2)
#Residual Histogram
hist((dataForPred[,1]-finalPredictions),
     col='grey',
     main='Residual Histogram for GAM Regression',
     xlab='Residual', ylab='Frequency')
postResample(predict(gamModel, dataForPred[, 2:13]), dataForPred[,1])
```

6.6 Clustering Code

```
# Clustering

# Normalize explanatory variables
normalized <- explanatory
for (col in 1:ncol(normalized)) {
  normalized[,col] <- (normalized[,col]-
min(normalized[,col]))/(max(normalized[,col]) - min(normalized[,col]))
}

# Clustering Euclidean Distance
```

```r
euclidean <- matrix(, nrow = 25, ncol = 2)
for(round in 1:2) {
  for(n in 1:25) {
    cl <- kmeans(normalized[, 3:14], n)
    euclidean[n, round] <- cl$tot.withinss
  }
}

# plot the euclidean distance vs number of centers
plot(euclidean[,2], type='o', col=1, pch=18, lty=1,
     main='Total Within-Cluster Sum of Squares v.s. Number of Centers',
     xlab='Number of Centers', ylab='Total Within-Cluster Sum of Squares')

# plot trials with different sets of starting points
plot(euclidean[,1], type='o', col=4, pch=18, lty=2,
     main='Results Generated by Different Randomly Chosen Start Point',
     xlab='Number of Centers', ylab='Total Within-Cluster Sum of Squares')
lines(euclidean[,2], type="o", pch=18, lty=1, col=1)
legend(18, 50, c("First Trial","Second Trial"), cex=0.8,
       col=c(1,4), pch=18:18, lty=1:2)


# perform k-means again with 50 sets of starting points and take average
for(n in 1:25) {
  cl <- kmeans(normalized[, 3:14], n, nstart=50)
  euclidean[n, round] <- cl$tot.withinss
}

plot(euclidean[,2], type='o', col=1, pch=18, lty=1,
     main='Sum of Squares v.s. Number of Centers',
     xlab='Number of Centers', ylab='Total Within-Cluster Sum of Squares')

# Clustering plots with 3 centres, with 50 sets of randomly chosen starting points
cl <- kmeans(normalized[, 3:14], 3, nstart=50)
plot(normalized[, 3:14], col = cl$cluster, pch=20)
points(cl$centers, col = cl$cluster, pch = 8, cex = 2)
```