DSC520 Final Project

Amie Davis

11 November, 2019

Data Source:

PDI (Police Data Initiative) Crime Incidents, City of Cincinnati
(PDI__Police_Data_Initiative__Crime_Incidents.csv)

1. Load Libraries

```
#install.packages("lubridate")

library(readr)
library(gdata)
library(ggplot2)
library(lubridate)
library(dplyr)
library(ggmap)
library(caTools)
library(class)
```

2. Load the Data

```
ci_data <-
read_csv("PDI__Police_Data_Initiative__Crime_Incidents_Revised2.csv",
  col_types = cols(
  .default = col_character(),
  UCR = col_double(),
  BEAT = col_character(),
  RPT_AREA = col_character(),
  LONGITUDE_X = col_double(),
  LATITUDE_X = col_double(),
  TOTALNUMBERVICTIMS = col_double(),
  TOTALSUSPECTS = col_double(),
  ZIP = col_character()
) )
```

3. Clean the Data

a) Limit to records with Cincinnati zip codes

```
cin_ci_data <- subset(ci_data, ZIP >= 45211 & ZIP <= 45280, c(INCIDENT_NO,
ZIP, OFFENSE, CPD_NEIGHBORHOOD, SUSPECT_AGE, SUSPECT_RACE, SUSPECT_GENDER,
CLSD, DAYOFWEEK, DATE_FROM, LATITUDE_X, LONGITUDE_X))
```

b) Limit to records with geodetic (lat/long) coordinates

```
cin_geo_data <- subset(cin_ci_data, !is.na(LATITUDE_X), c(INCIDENT_NO, ZIP,
OFFENSE, CPD_NEIGHBORHOOD, SUSPECT_AGE, SUSPECT_RACE, SUSPECT_GENDER, CLSD,
DAYOFWEEK, DATE_FROM, LATITUDE_X, LONGITUDE_X))
```

c) Exclude records missing critical data elements

```
cln_data <- subset(cin_geo_data, !is.na(DATE_FROM), c(INCIDENT_NO, ZIP,
OFFENSE, CPD_NEIGHBORHOOD, SUSPECT_AGE, SUSPECT_RACE, SUSPECT_GENDER, CLSD,
DAYOFWEEK, DATE_FROM, LATITUDE_X, LONGITUDE_X))
```

d) Convert categorical variables to factors

```
cln_data$SUSPECT_AGE <- factor(cln_data$SUSPECT_AGE)
cln_data$SUSPECT_RACE <- factor(cln_data$SUSPECT_RACE)
cln_data$SUSPECT_GENDER <- factor(cln_data$SUSPECT_GENDER)
cln_data$DAYOFWEEK <- factor(cln_data$DAYOFWEEK, levels = c("SUNDAY",
"MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY"))
```

Drop unused factors.

```
x <- drop.levels(cln_data)
```

4. Review the Data

a) Review Structure

```
str(cln_data)

## Classes 'tbl_df', 'tbl' and 'data.frame':    240403 obs. of  12 variables:
##  $ INCIDENT_NO     : chr  "11101449" "11100755" "11102840" "31009187" ...
##  $ ZIP             : chr  "45211" "45211" "45211" "45211" ...
##  $ OFFENSE         : chr  "THEFT" "THEFT" "THEFT" "TELEPHONE HARASSMENT"
...
##  $ CPD_NEIGHBORHOOD: chr  "C. B. D. / RIVERFRONT" "OVER-THE-RHINE" "OVER-
THE-RHINE" "C. B. D. / RIVERFRONT" ...
##  $ SUSPECT_AGE     : Factor w/ 9 levels "18-25","26-30",..: 9 9 9 9 9 1 9
8 8 2 ...
##  $ SUSPECT_RACE    : Factor w/ 8 levels "AMERICAN IINDIAN/ALA",..: NA NA
NA 5 NA 7 NA 5 5 5 ...
##  $ SUSPECT_GENDER  : Factor w/ 6 levels "F - FEMALE","FEMALE",..: NA NA NA
2 NA 2 NA 4 4 4 ...
```

```
## $ CLSD             : chr  "Z--EARLY CLOSED" "F--CLEARED BY ARREST - ADULT"
"Z--EARLY CLOSED" "J--CLOSED" ...
## $ DAYOFWEEK        : Factor w/ 7 levels "SUNDAY","MONDAY",..: 7 6 4 7 3 1
3 3 3 5 ...
## $ DATE_FROM        : chr  "4/16/2011 12:00" "2/25/2011 16:45" "6/15/2011
12:00" "9/11/2010 16:30" ...
## $ LATITUDE_X       : num  39.1 39.1 39.1 39.1 39.1 ...
## $ LONGITUDE_X      : num  -84.5 -84.5 -84.5 -84.6 -84.6 ...
```

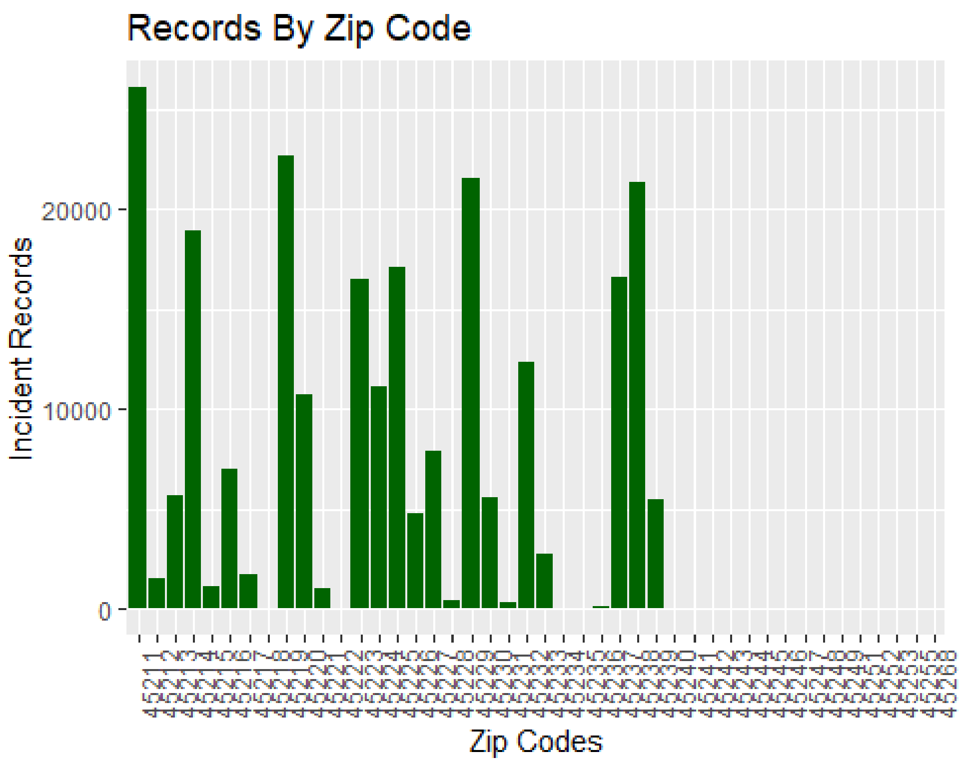
```
#head(cln_data)
summary(cln_data)
```

```
##  INCIDENT_NO             ZIP                OFFENSE
##  Length:240403      Length:240403      Length:240403
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  CPD_NEIGHBORHOOD     SUSPECT_AGE                  SUSPECT_RACE
##  Length:240403      UNKNOWN :159093   BLACK                : 86018
##  Class :character   18-25   : 27623   WHITE                : 19768
##  Mode  :character   31-40   : 15944   UNKNOWN              :  6024
##                     26-30   : 14629   ASIAN/PACIFIC ISLAND:   154
##                     UNDER 18:  9126   AMERICAN INDIAN/ALAS:    53
##                     41-50   :  8729   (Other)              :    29
##                     (Other) :  5259   NA's                 :128357
##              SUSPECT_GENDER       CLSD              DAYOFWEEK
##  F - FEMALE          :     1   Length:240403      FRIDAY  :35208
##  FEMALE              : 28413   Class :character   SATURDAY:34899
##  M - MALE            :     5   Mode  :character   SUNDAY  :33985
##  MALE                : 79783                      MONDAY  :33900
##  NON-PERSON (BUSINESS:    24                      TUESDAY :33810
##  UNKNOWN             :  3820                      (Other) :66743
##  NA's                :128357                      NA's    : 1858
##   DATE_FROM           LATITUDE_X     LONGITUDE_X
##  Length:240403      Min.   :39.05   Min.   :-84.82
##  Class :character   1st Qu.:39.13   1st Qu.:-84.57
##  Mode  :character   Median :39.15   Median :-84.53
##                     Mean   :39.15   Mean   :-84.52
##                     3rd Qu.:39.18   3rd Qu.:-84.49
##                     Max.   :39.36   Max.   :-84.25
##
```

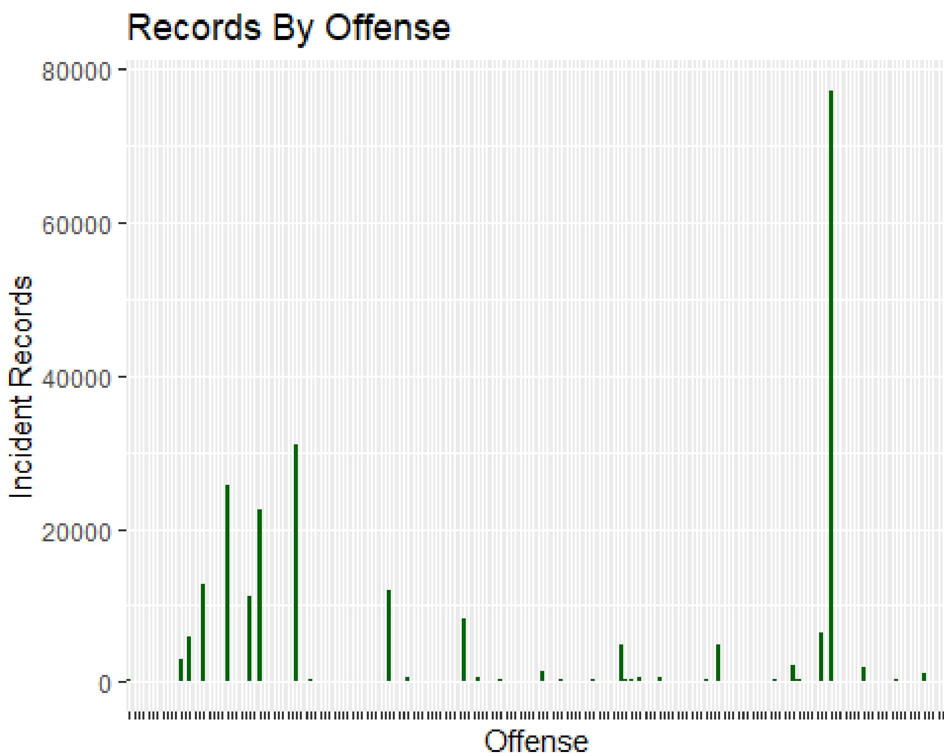
There are more unknown or NA values for suspect's age, race, and gender than there are known values, so I will remove those variables from the dataset.

b) Review Distributions

```r
ggplot(cln_data, aes(as.factor(x=ZIP))) +
    geom_bar(fill="dark green") +
    labs(x="Zip Codes", y="Incident Records", title="Records By Zip Code") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

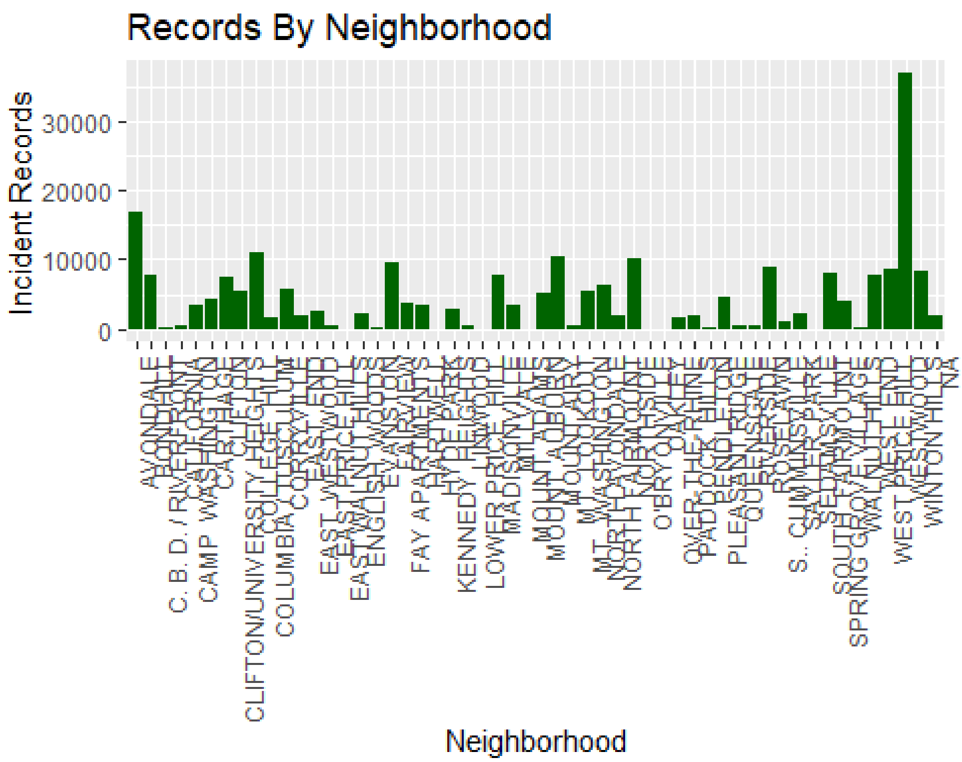



```r
ggplot(cln_data, aes(as.factor(x=OFFENSE))) +
    geom_bar(fill="dark green") +
    labs(x="Offense", y="Incident Records", title="Records By Offense") +
    theme(axis.text.x = element_blank())
```



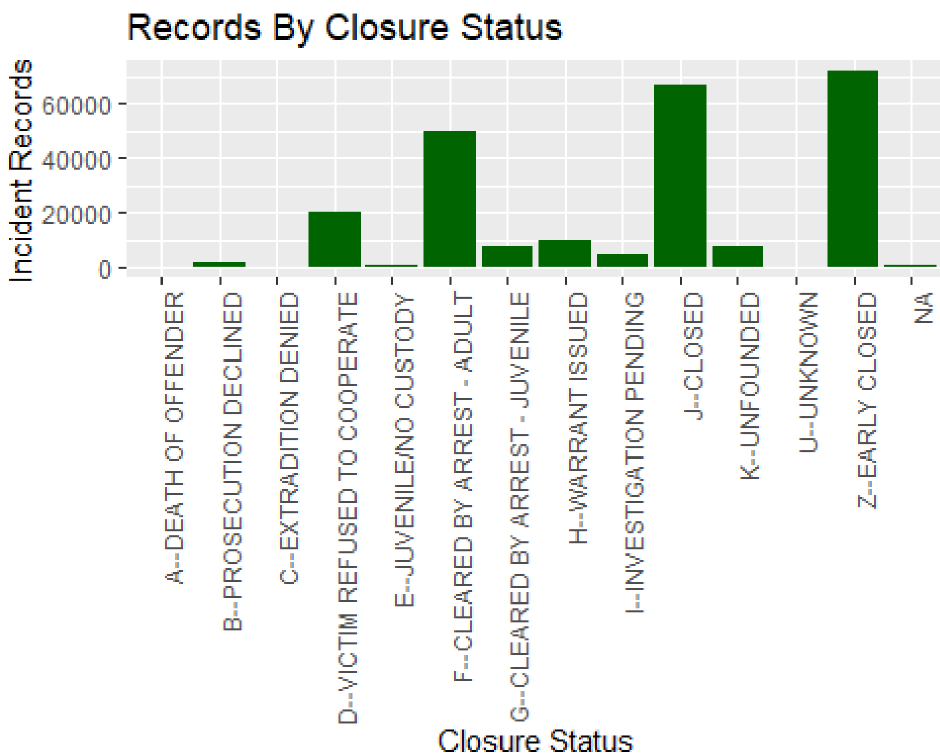
That spike in the chart is for “Theft”. I will look into this further later.

```
ggplot(cln_data, aes(x=as.factor(CPD_NEIGHBORHOOD))) +
    geom_bar(fill="dark green") +
    labs(x="Neighborhood", y="Incident Records", title="Records By
Neighborhood") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



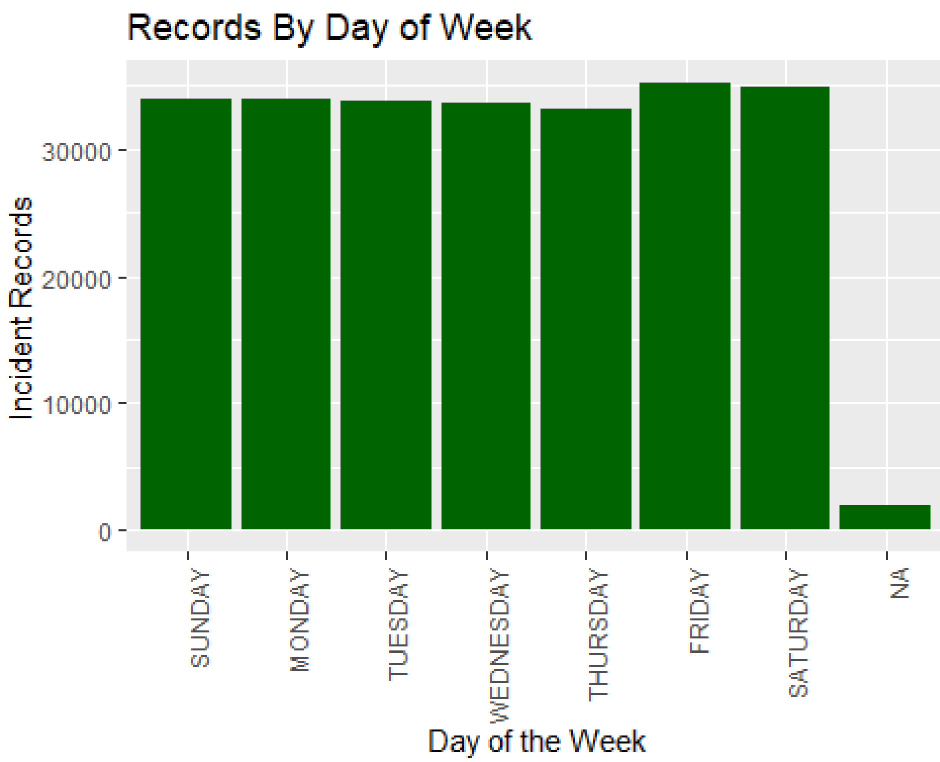
There is another spike here in the “Westwood” neighborhood. I will revisit this later.

```
ggplot(cln_data, aes(x=as.factor(CLSD))) +
    geom_bar(fill="dark green") +
    labs(x="Closure Status", y="Incident Records", title="Records By Closure
Status") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



The large number of incidents designated "EARLY CLOSED" is interesting, especially in light of recent news headlines. I have contacted the dataset owner for more information as to what is meant by "Early Closure."

```
ggplot(cln_data, aes(x=DAYOFWEEK)) +
    geom_bar(fill="dark green") +
    labs(x="Day of the Week", y="Incident Records", title="Records By Day of
Week") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



The distribution by the Day of the Week does not vary much, so I will remove that variable from the dataset.

5. Derived Data

```
# Convert DATE_FROM field to a datetime stamp
date_data <- cln_data %>% mutate(DATE_FROM = mdy_hm(DATE_FROM))

# Split date field into separate columns
der_date_data <- date_data %>% mutate (YEAR = year(DATE_FROM),
                                       MONTH = month(DATE_FROM),
                                       DAY = day(DATE_FROM),
                                       HOUR = hour(DATE_FROM),
                                       MINUTE = minute(DATE_FROM))

# Add Comparison/Fiscal year field (Oct-Sep)
der_date_data$COMP_YEAR <- ifelse(der_date_data$MONTH >= 10,
der_date_data$YEAR+1, der_date_data$YEAR)
der_date_data$COMP_YEAR <- factor(der_date_data$COMP_YEAR)

# Order months for Comparison/Fiscal Year (Oct-Sep)
der_date_data$MONTH <- factor(der_date_data$MONTH, levels =
c("10","11","12","1","2","3","4","5","6","7","8","9"))
```

6. Explore Data

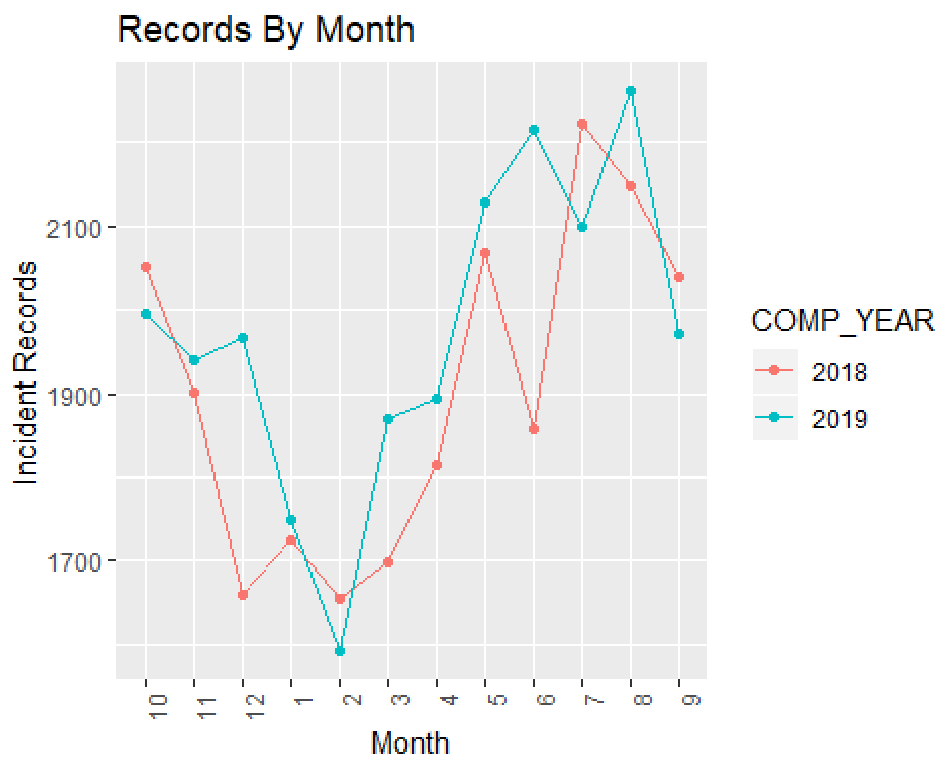
a) Filter to use only two years worth of data.

Using 1Oct2018-30Sep2019 for current year and 1Oct2017-30Sep2018 for last year.

```
recent_data <- subset(der_date_data, (COMP_YEAR == 2018 |
                                      COMP_YEAR == 2019),
 c(INCIDENT_NO, ZIP, OFFENSE, CPD_NEIGHBORHOOD, CLSD, DATE_FROM, LATITUDE_X,
LONGITUDE_X, MONTH, HOUR, COMP_YEAR))
```

b) Plot incidents by month. Compare to previous year.

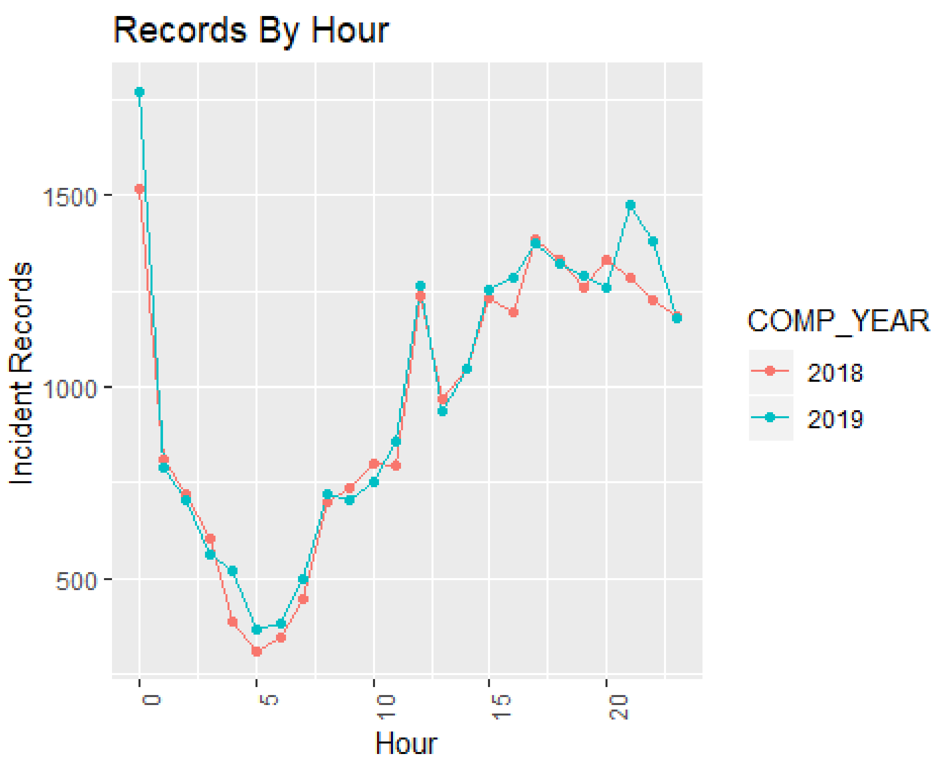
```
month_df <- recent_data %>% group_by(MONTH, COMP_YEAR) %>% tally()

ggplot(month_df, aes(x=MONTH, y=n, color=COMP_YEAR, group=COMP_YEAR)) +
    geom_line() +
    geom_point() +
    labs(x="Month", y="Incident Records", title="Records By Month") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



As you can see, incident reporting is down in the winter months. The least reported is February in both years, while incident reporting is up in the summer months. The greatest is in July both years. You can see that, although this year dropped quite a bit in February, incidents increased this past summer. Reported incidents increased this year in all but three months: October, February, and July. This year's trend is similar to the previous year, but has increased overall.

c) Plot incidents by time of day. Compare to previous year.

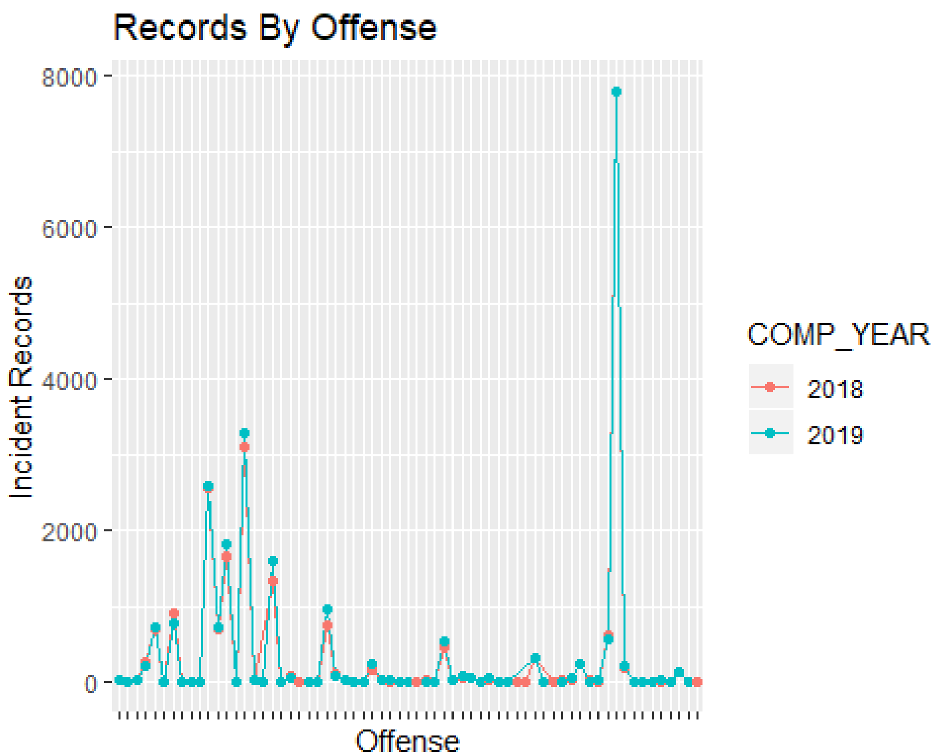
```
hr_df <- recent_data %>% group_by(HOUR, COMP_YEAR) %>% tally()

ggplot(hr_df, aes(x=HOUR, y=n, color=COMP_YEAR, group=COMP_YEAR)) +
    geom_line() +
    geom_point() +
    labs(x="Hour", y="Incident Records", title="Records By Hour") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



The plot indicates incident reporting peaks at midnight and drops dramatically afterward. Incident reporting increases during the daytime until noon. Both years show consistent data.

d) Plot incidents by offense. Compare to previous year.

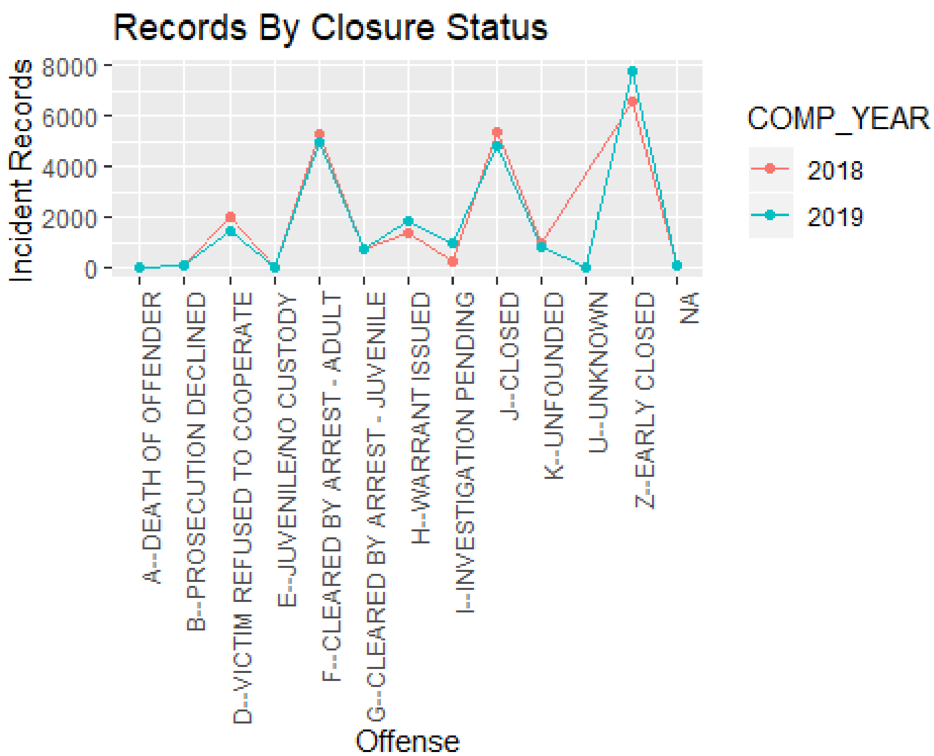
```
off_df <- recent_data %>% group_by(OFFENSE, COMP_YEAR) %>% tally()

ggplot(off_df, aes(x=OFFENSE, y=n, color=COMP_YEAR, group=COMP_YEAR)) +
    geom_line() +
    geom_point() +
    labs(x="Offense", y="Incident Records", title="Records By Offense") +
    theme(axis.text.x = element_blank())
```



Thefts are reported far more than other offenses. Compared to last year, the number of offenses by type is consistent.

e) Plot incidents by closure status. Compare to previous year.

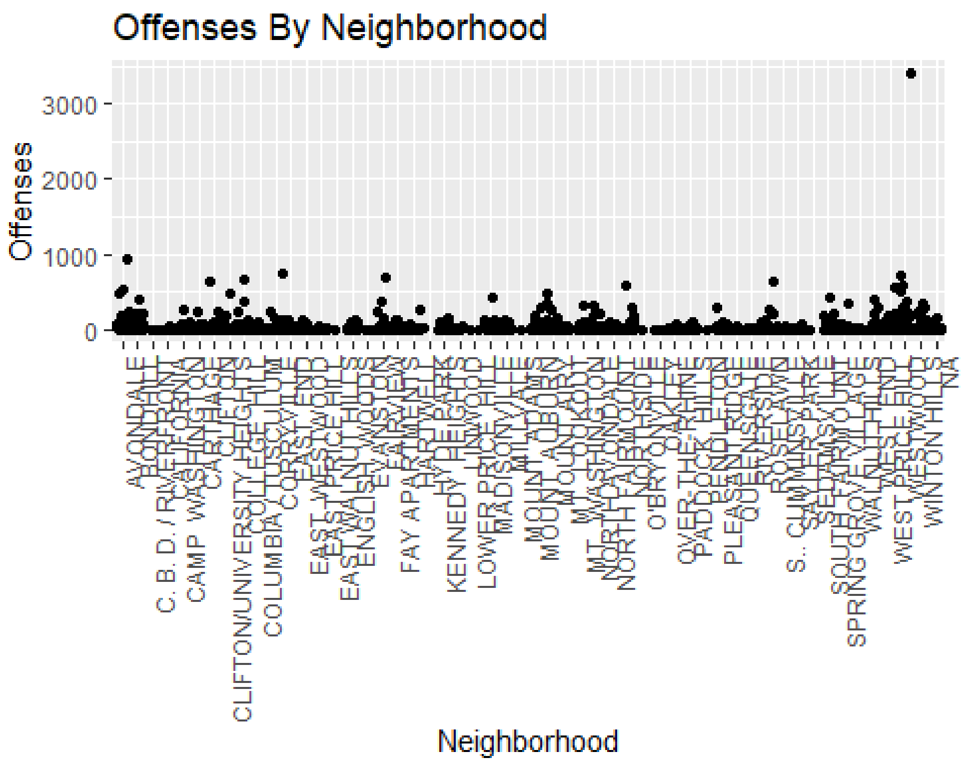
```
clsd_df <- recent_data %>% group_by(CLSD, COMP_YEAR) %>% tally()

ggplot(clsd_df, aes(x=CLSD, y=n, color=COMP_YEAR, group=COMP_YEAR)) +
    geom_line() +
    geom_point() +
    labs(x="Offense", y="Incident Records", title="Records By Closure
Status") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Note the large number of incidents closed with "Early Closed" status. I have contacted the data steward for clarification between "Closed" and "Early Closed" statuses.

f) Look at offenses by neighborhood

```
hood_df <- recent_data %>% group_by(CPD_NEIGHBORHOOD, OFFENSE)%>% tally()

ggplot(hood_df, aes(x=CPD_NEIGHBORHOOD, y=n)) +
  geom_point(position="jitter") +
    labs(x="Neighborhood", y="Offenses", title="Offenses By Neighborhood") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



One type of offense is reported most frequently in a particular neighborhood. This is worth further investigation.

```
west_df <- hood_df %>% filter(CPD_NEIGHBORHOOD == "WESTWOOD")
west_df[order(west_df$n, decreasing = TRUE),]
```

```
## # A tibble: 48 x 3
## # Groups:   CPD_NEIGHBORHOOD [1]
##    CPD_NEIGHBORHOOD OFFENSE                            n
##    <chr>            <chr>                          <int>
##  1 WESTWOOD         THEFT                           3401
##  2 WESTWOOD         CRIMINAL DAMAGING/ENDANGERING    725
##  3 WESTWOOD         ASSAULT                          576
##  4 WESTWOOD         DOMESTIC VIOLENCE                506
##  5 WESTWOOD         BURGLARY                         371
##  6 WESTWOOD         AGGRAVATED ROBBERY               284
##  7 WESTWOOD         AGGRAVATED MENACING              225
##  8 WESTWOOD         BREAKING AND ENTERING            181
##  9 WESTWOOD         TELEPHONE HARASSMENT             159
## 10 WESTWOOD         FELONIOUS ASSAULT                155
## # ... with 38 more rows
```

There is clearly a problem with theft in Westwood.

7. Look for correlarion

a) Create data frame with numeric counts

Since the orginal dataset contained only categorical variables, I need to derive counts to perform any correlation analysis.

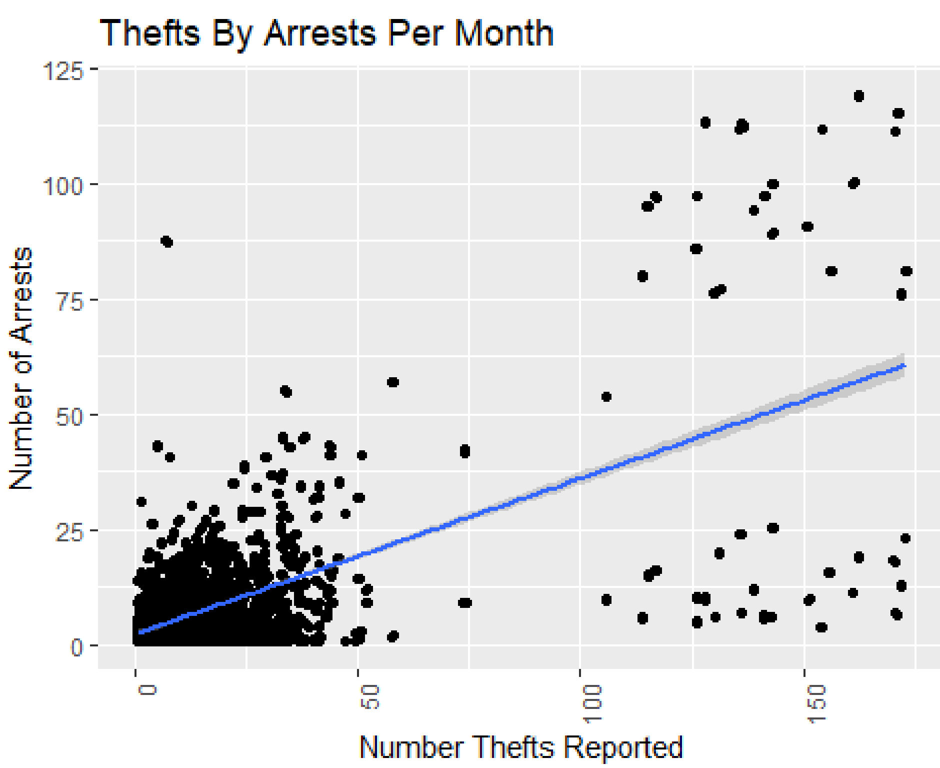
```
# Count thefts by Neighborhood
inc_df <- recent_data %>% group_by(MONTH, COMP_YEAR, CPD_NEIGHBORHOOD,
OFFENSE) %>% tally()
theft_df <- inc_df %>% filter(OFFENSE == "THEFT")
names(theft_df)[5]<-"THEFT_CNT"

# Count arrests by Neighborhood
clsd_df <- recent_data %>% group_by(MONTH, COMP_YEAR, CPD_NEIGHBORHOOD, CLSD)
%>% tally()
arr_df <- clsd_df %>% filter((CLSD == "F--CLEARED BY ARREST - ADULT" | CLSD
== "G--CLEARED BY ARREST - JUVENILE"))
names(arr_df)[5]<-"ARREST_CNT"

# Count closed cases by Neighborhood
clsd_df <- clsd_df %>% filter((CLSD == "J--CLOSED" | CLSD == "Z--EARLY
CLOSED"))
names(clsd_df)[5]<-"CLOSED_CNT"

# Join datasets
tally_df <- merge(theft_df, arr_df,by=c("MONTH", "COMP_YEAR",
"CPD_NEIGHBORHOOD"))
tally_df <- merge(tally_df, clsd_df,by=c("MONTH", "COMP_YEAR",
"CPD_NEIGHBORHOOD"))
summary(tally_df)
```

```
##      MONTH         COMP_YEAR    CPD_NEIGHBORHOOD     OFFENSE
##  10     : 228   2018    :1224   Length:2425        Length:2425
##  8      : 218   2019    :1201   Class :character   Class :character
##  4      : 216   1991    :   0   Mode  :character   Mode  :character
##  7      : 211   1992    :   0
##  5      : 205   1993    :   0
##  2      : 201   1994    :   0
##  (Other):1146   (Other):   0
##    THEFT_CNT        CLSD.x            ARREST_CNT        CLSD.y
##  Min.   :  1.0   Length:2425        Min.   :  1.000   Length:2425
##  1st Qu.:  7.0   Class :character   1st Qu.:  2.000   Class :character
##  Median : 14.0   Mode  :character   Median :  5.000   Mode  :character
##  Mean   : 21.1                      Mean   :  9.426
##  3rd Qu.: 25.0                      3rd Qu.: 11.000
##  Max.   :173.0                      Max.   :119.000
##
```

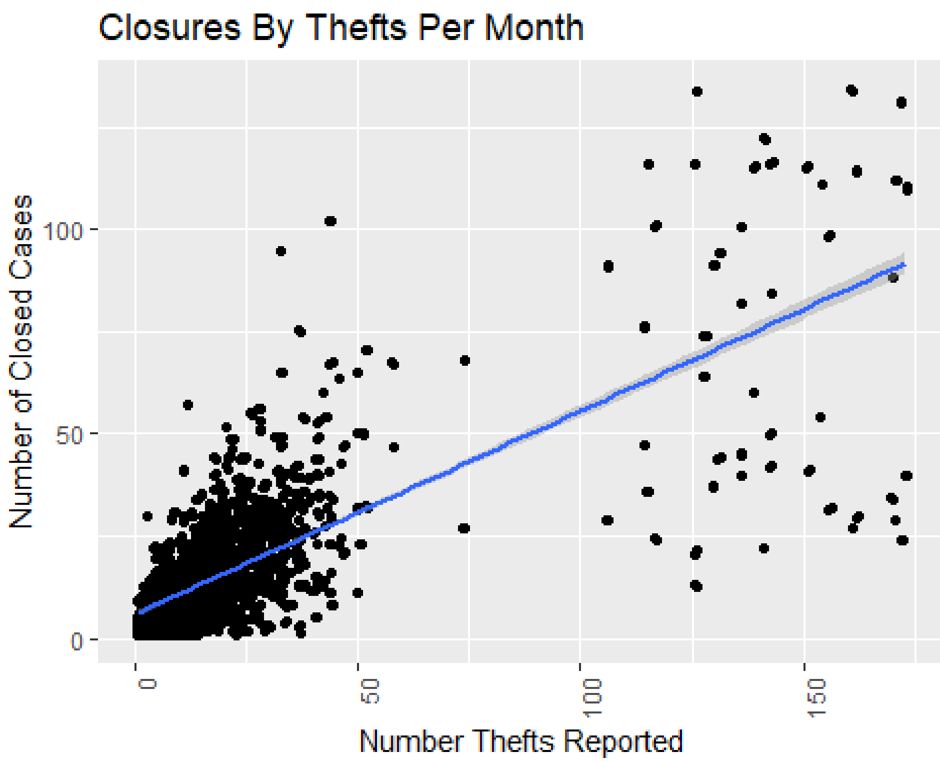
```
##    CLOSED_CNT
##  Min.   :  1.00
##  1st Qu.:  5.00
##  Median : 12.00
##  Mean   : 16.44
##  3rd Qu.: 22.00
##  Max.   :134.00
##
```

b) Look for relationship between the number of thefts and the number of arrests and closed cases

```
ggplot(tally_df, aes(x=THEFT_CNT, y=ARREST_CNT)) +
  geom_point(position="jitter") +
    labs(x="Number Thefts Reported", y="Number of Arrests", title="Thefts By
Arrests Per Month") +
    geom_smooth(method="lm") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

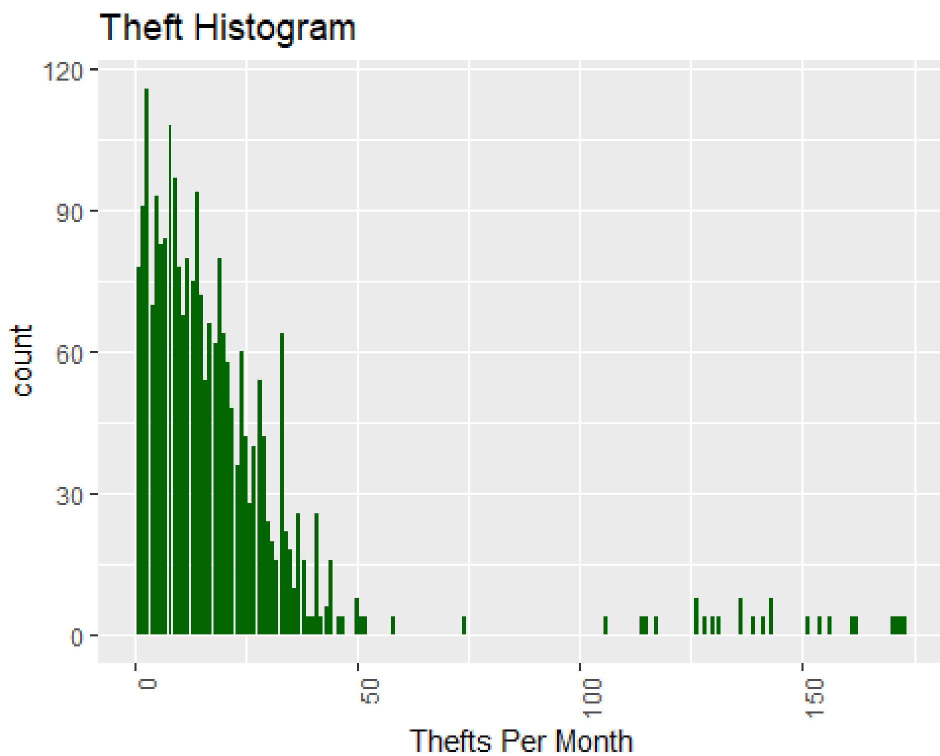



```
ggplot(tally_df, aes(x=THEFT_CNT, y=CLOSED_CNT)) +
  geom_point(position="jitter") +
    labs(x="Number Thefts Reported", y="Number of Closed Cases",
title="Closures By Thefts Per Month") +
    geom_smooth(method="lm") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

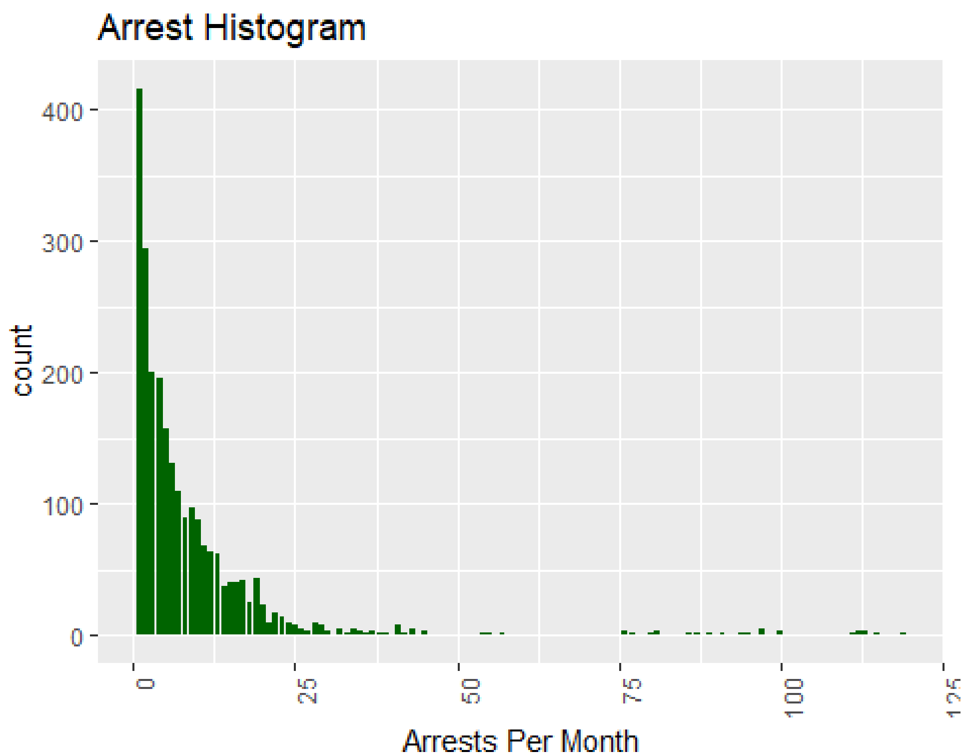


c) Check for normality

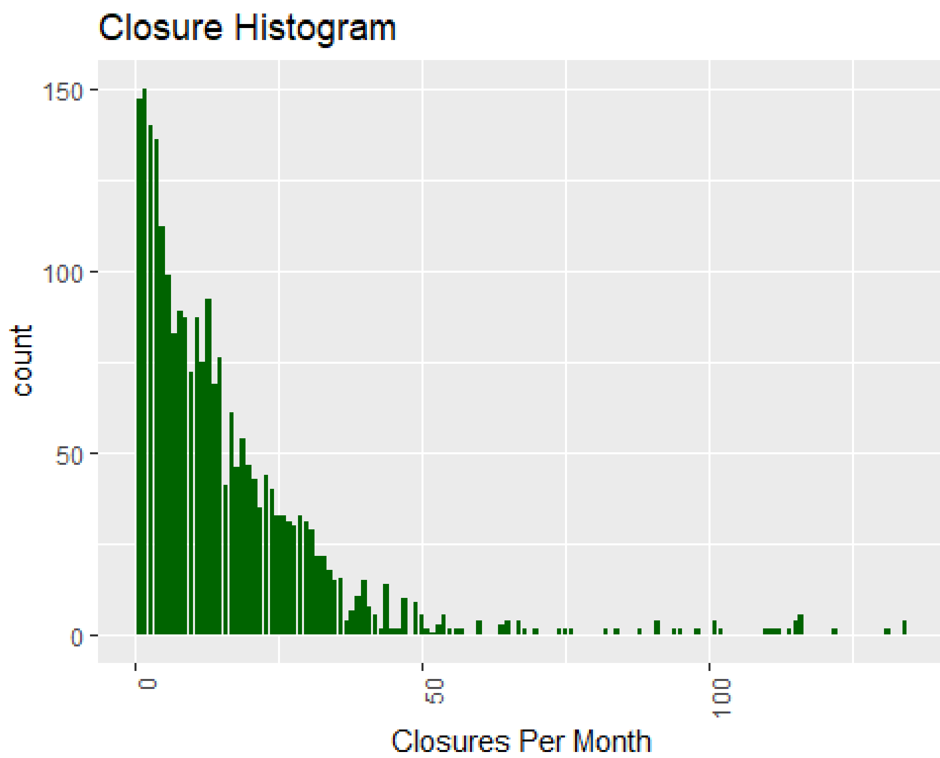
```
ggplot(tally_df, aes(THEFT_CNT)) +
    geom_bar(fill="dark green") +
    labs(x="Thefts Per Month", title="Theft Histogram") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
ggplot(tally_df, aes(ARREST_CNT)) +
    geom_bar(fill="dark green") +
    labs(x="Arrests Per Month", title="Arrest Histogram") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```r
ggplot(tally_df, aes(CLOSED_CNT)) +
    geom_bar(fill="dark green") +
    labs(x="Closures Per Month", title="Closure Histogram") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



d) Test for correlation

Since the distributions are skewed, I will use kendall's tau instead of Person's r to determine correlation.

```
cor.test(tally_df$THEFT_CNT, tally_df$ARREST_CNT, method="kendall")

##
##  Kendall's rank correlation tau
##
## data:  tally_df$THEFT_CNT and tally_df$ARREST_CNT
## z = 17.441, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2478549

cor.test(tally_df$THEFT_CNT, tally_df$CLOSED_CNT, method="kendall")

##
##  Kendall's rank correlation tau
##
## data:  tally_df$THEFT_CNT and tally_df$CLOSED_CNT
## z = 40.505, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
```

```
## sample estimates:
##       tau
## 0.5649966
```

Both show a significant positive relationship. This confirms what is visually displayed in the graphs.

e) Build Linear Model

Given the number of thefts reported by neighborhood, we can predict the number of arrests and closures using linear models.

```
lr_mod1 <- lm(ARREST_CNT ~ THEFT_CNT, tally_df)
summary(lr_mod1)
```

```
##
## Call:
## lm(formula = ARREST_CNT ~ THEFT_CNT, data = tally_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.036  -4.730  -1.431   2.971  82.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.302206   0.294885   7.807 8.63e-15 ***
## THEFT_CNT   0.337623   0.008567  39.410  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.47 on 2423 degrees of freedom
## Multiple R-squared:  0.3906, Adjusted R-squared:  0.3904
## F-statistic:  1553 on 1 and 2423 DF,  p-value: < 2.2e-16
```

```
lr_mod2 <- lm(CLOSED_CNT ~ THEFT_CNT, tally_df)
summary(lr_mod2)
```

```
##
## Call:
## lm(formula = CLOSED_CNT ~ THEFT_CNT, data = tally_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.279  -5.942  -2.389   5.042  74.204
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.974192   0.311279   19.19   <2e-16 ***
## THEFT_CNT   0.495956   0.009043   54.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12.11 on 2423 degrees of freedom
## Multiple R-squared:  0.5538, Adjusted R-squared:  0.5537
## F-statistic:  3008 on 1 and 2423 DF,  p-value: < 2.2e-16
```

Using these linear models, we can predict the number of arrests and closures based on the numbers of thefts reported.

8. Model Data - kNN

a) Plot crime incidents by offense against location data.

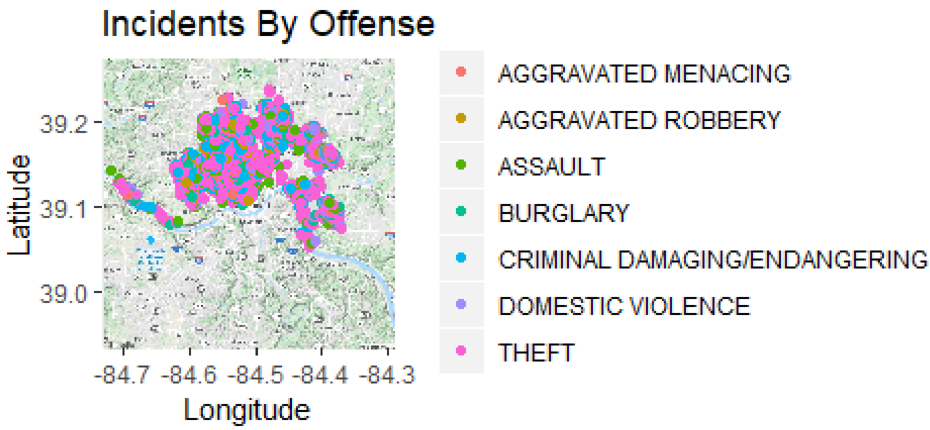
I will expand the dataset to better train the model to include data from 2011 forward. Also, I will only review records where NEIGHBORHOOD and OFFENSE have values. Since there are 65 different categories, I will limit the offenses in the model to the top 6 most frequently reported offenses.

```
model_data <- subset(recent_data, !is.na(OFFENSE) & !is.na(CPD_NEIGHBORHOOD),
 c(OFFENSE, CPD_NEIGHBORHOOD, LATITUDE_X, LONGITUDE_X))
new_mod_data <- subset(model_data, OFFENSE=="THEFT" | OFFENSE=="CRIMINAL
DAMAGING/ENDANGERING" | OFFENSE=="ASSAULT" | OFFENSE=="DOMESTIC VIOLENCE" |
OFFENSE=="BURGLARY" | OFFENSE=="AGGRAVATED ROBBERY" | OFFENSE=="AGGRAVATED
MENACING", c(OFFENSE, CPD_NEIGHBORHOOD, LATITUDE_X, LONGITUDE_X))
```

Using ggmap to use Google maps to display geodetic information. The API key is passed, but is hidden from Markdown.

b) Plot crime incidents by top 6 reported offenses against location data.

```
# Use Cincinnati coordinates as center
cin_map1 <- ggmap(get_googlemap(
                  center = c(lon = -84.512016, lat = 39.103119),
                  zoom = 11, scale = 2,
                  maptype = "terrain",
                  color="color"))
cin_map1 +
  geom_point(data = new_mod_data,
             aes(x=LONGITUDE_X, y=LATITUDE_X, color = OFFENSE)) +
  labs(x="Longitude", y="Latitude", title="Incidents By Offense") +
  theme(legend.title = element_blank())
```



c) Split the data set, randomly into test and train sets.

```
split_off_set <- sample.split(new_mod_data$OFFENSE,SplitRatio=0.8)
train_off_set <- subset(new_mod_data, split_off_set=="TRUE")
test_off_set <- subset(new_mod_data, split_off_set=="FALSE")
```

Separate Labels

Before running the data through a nearest neighbor model, we need to separate the labels from the data.

```
train_off_labels <- train_off_set[,1, drop=TRUE]
test_off_labels <- test_off_set[,1, drop=TRUE]
train_off_data <- train_off_set[,3:4]
test_off_data <- test_off_set[,3:4]
```

d) Build kNN models with training dataset

Now, we can build the models with the training sets, using a variety of k values.

```
knn_off.3<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=3)
knn_off.5<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=5)
knn_off.10<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=10)
knn_off.15<- knn(train = train_off_data, test = test_off_data, cl =
```

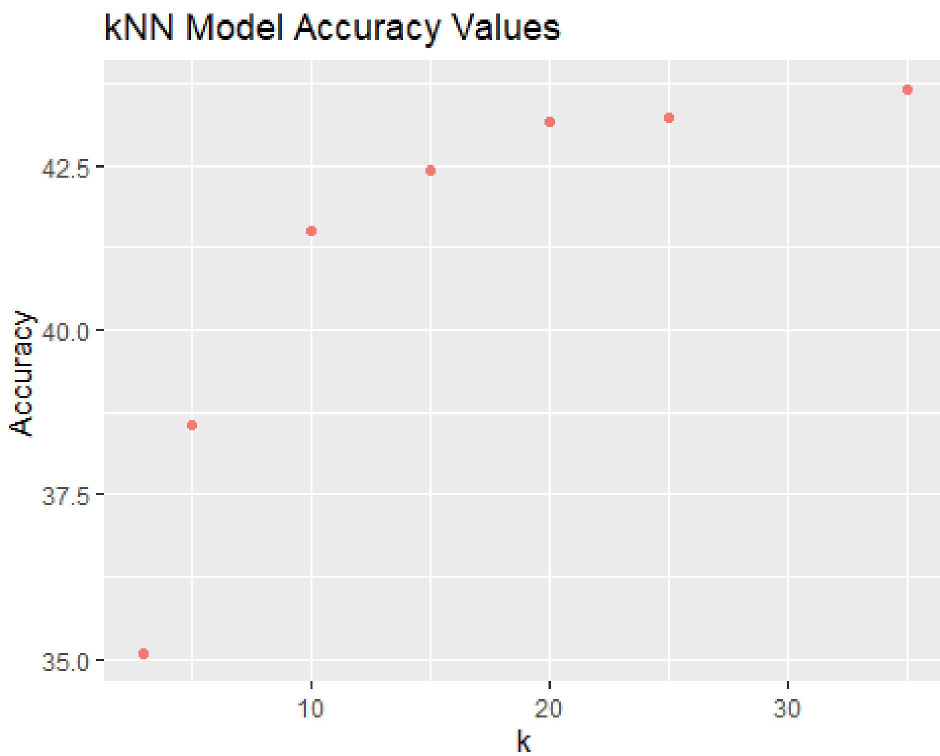
```
train_off_labels, k=15)
knn_off.20<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=20)
knn_off.25<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=25)
knn_off.35<- knn(train = train_off_data, test = test_off_data, cl =
train_off_labels, k=35)
```

e) Test kNN model with test dataset

```
# Accuracy for offense model
ACC_off.3 <- 100 * sum(test_off_labels == knn_off.3)/NROW(test_off_labels)
ACC_off.5 <- 100 * sum(test_off_labels == knn_off.5)/NROW(test_off_labels)
ACC_off.10 <- 100 * sum(test_off_labels == knn_off.10)/NROW(test_off_labels)
ACC_off.15 <- 100 * sum(test_off_labels == knn_off.15)/NROW(test_off_labels)
ACC_off.20 <- 100 * sum(test_off_labels == knn_off.20)/NROW(test_off_labels)
ACC_off.25 <- 100 * sum(test_off_labels == knn_off.25)/NROW(test_off_labels)
ACC_off.35 <- 100 * sum(test_off_labels == knn_off.35)/NROW(test_off_labels)

# Add accuracy values to a new data frame
k <- c(3,5,10,15,20,25,35)
ACC <- c(ACC_off.3, ACC_off.5, ACC_off.10, ACC_off.15, ACC_off.20,
ACC_off.25, ACC_off.35)
ACC_df <- data.frame(k, ACC, stringsAsFactors=FALSE)
```

Plot accuracy values

```
# Convert data types for data frame
ACC_df$k <- as.numeric(ACC_df$k)
ACC_df$ACC <- as.numeric(ACC_df$ACC)

ggplot(ACC_df, aes(x=k, y=ACC, col="light orange")) +
  geom_point() +
  labs(title="kNN Model Accuracy Values", y="Accuracy") +
    theme(legend.position = "none")
```



The best I will get with this model is around 43% accuracy with k=25 clusters.

References

https://www.littlemissdata.com/blog/maps?format=amp https://www.latlong.net D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf