R Notebook

Code

NYPD Crime Statistics Project Analysis

The New York City police department, more commonly known as the NYPD, is one of the oldest and biggest police departments in the US. It is broken down into 77 precincts, which each span a relatively small amount of territory. As a whole, there are roughly 40,000 police officers and has an annual budget of 5.6 billion dollars. Unsurprisingly, the NYPD responds to about half a million complaints each year. As part of an initiave to be more transparent, the NYPD releases most of its crime statistics, including complaints, arrests, shootings, and court summons. This data is found here: https://www1.nyc.gov/site/nypd/stats/crime-statistics/citywide-crime-stats.page (https://www1.nyc.gov/site/nypd/stats/crime-statistics/citywide-crime-stats.page)

The data I am using for this project is the incident level complaint dataset and can be found here: https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i (https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i)

On the website, this dataset is stated to include "all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) from 2006 to the end of last year (2017)".

This dataset contains 35 variables and approximately six million observations. A full description of each variable can be found on the website but I go through the relevant variables below.

Throughout this project, I will be exploring the various facets of this data set and be attempting to answer the following questions:

1. Is there a borough that contains a higher perctage of a specific crime?
2. What is the spread of gender of the victims and suspects?
3. Is it more likely to occur during the day or at night?
4. Can the time that it takes for a victim to report a crime be predicted based on specific variables?
5. Can crime statistics be forecasted?

Required Libraries

In order to use the ggmap library, you must register for a Google Maps API key and run register_google(key).

Hide

```
library(lubridate)
library(tidyverse)
library(ggmap)
library(forecast)
library(RSQLite)
library(corrplot)
register_google("")
```

Data Loading

Hide

```
complaints<-read_csv("NYPD_Complaint_Data_Historic.csv")
nyc_base <-
  ggmap::get_map(location = c(lon = -73.95, lat = 40.7), zoom = 11)
```

Data Cleaning and Manipulation

The complaints dataset had many variables and I did not end up using them all. With the variables that remained, most are understandable. However, I've included descriptions of variables that have illegible shorthand.

CMPLNT_NUM - unique complaint id

CMPLNT_FR_* - date/time that incident started

CMPLNT_TO_* - date/time that incident finished

RPT_DT - date incident was reported

ADDR_PCT_CD - precinct number

PD_CD, PD_DESC - internal description and code of type of incident

Because this data was taken from a csv file, it did not need a significant amount of cleaning. I converted all of the NA values, as well as values that were out of place, of the demographics of the suspect and victim to "Unknown"

In addition, I converted the dates and times to datetime objects using the lubridate package and truncated the latitude and longitude values. The values that were recorded were accurate to one meter which I believe was unneccessary.

I included several new variables that would make visualization easier. First, I added both the duration of the incident and the length of time that it took for the incident to be reported. Finally, I added a boolean that recorded whether the incident occurred at night or in the day.

Hide

```
cleanedComplaints<-NA
cleanedComplaints <-
  complaints[, c(1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 15, 24, 25, 26, 28, 29, 33, 34, 35)]

  cleanedComplaints$SUSP_AGE_GROUP[is.na(cleanedComplaints$SUSP_AGE_GROUP)] <-
  "UNKNOWN"
    cleanedComplaints$SUSP_AGE_GROUP[cleanedComplaints$SUSP_AGE_GROUP=="U"] <-
  "UNKNOWN"
  cleanedComplaints$SUSP_RACE[is.na(cleanedComplaints$SUSP_RACE)] <-
  "UNKNOWN"
    cleanedComplaints$SUSP_RACE[cleanedComplaints$SUSP_RACE=="U"] <-
  "UNKNOWN"
    cleanedComplaints$SUSP_SEX[cleanedComplaints$SUSP_SEX=="U"|is.na(cleanedComplaints$SUSP_SE
X)] <-
  "UNKNOWN"
  cleanedComplaints$VIC_AGE_GROUP[is.na(cleanedComplaints$VIC_AGE_GROUP)] <-
  "UNKNOWN"
    cleanedComplaints$VIC_AGE_GROUP[cleanedComplaints$VIC_AGE_GROUP=="U"] <-
  "UNKNOWN"
  cleanedComplaints$VIC_RACE[is.na(cleanedComplaints$VIC_RACE)] <-
  "UNKNOWN"
    cleanedComplaints$VIC_RACE[cleanedComplaints$VIC_RACE=="U"] <-
  "UNKNOWN"

  cleanedComplaints$VIC_SEX[cleanedComplaints$VIC_SEX=="U"|cleanedComplaints$VIC_SEX=="E"|cleane
dComplaints$VIC_SEX=="D"|is.na(cleanedComplaints$VIC_SEX)] <-
  "UNKNOWN"


  cleanedComplaints$CMPLNT_FR_DT <- mdy(cleanedComplaints$CMPLNT_FR_DT)
  cleanedComplaints$CMPLNT_TO_DT <- mdy(cleanedComplaints$CMPLNT_TO_DT)

  cleanedComplaints$Longitude <-
  round(cleanedComplaints$Longitude, 2) + .005
  cleanedComplaints$Latitude <-
  round(cleanedComplaints$Latitude, 2) + .005

  cleanedComplaints$location <-
  paste(cleanedComplaints$Latitude, cleanedComplaints$Longitude)


  cleanedComplaints$CMPLNT_FR <-
  paste(cleanedComplaints$CMPLNT_FR_DT,
  cleanedComplaints$CMPLNT_FR_TM)
  cleanedComplaints$CMPLNT_TO <-
  paste(cleanedComplaints$CMPLNT_TO_DT,
  cleanedComplaints$CMPLNT_TO_TM)

  cleanedComplaints$CMPLNT_FR <- ymd_hms(cleanedComplaints$CMPLNT_FR)
  cleanedComplaints$CMPLNT_TO <- ymd_hms(cleanedComplaints$CMPLNT_TO)

  cleanedComplaints$CMPLNT_FR[is.na(cleanedComplaints$CMPLNT_FR)] <-
  as_datetime(0)
```

```
cleanedComplaints$CMPLNT_TO[is.na(cleanedComplaints$CMPLNT_TO)] <-
cleanedComplaints$CMPLNT_FR[is.na(cleanedComplaints$CMPLNT_TO)]

cleanedComplaints$RPT_DT <- mdy(cleanedComplaints$RPT_DT)

cleanedComplaints$INT_OCCUR <-
int_length(interval(cleanedComplaints$CMPLNT_FR, cleanedComplaints$CMPLNT_TO)) /
(60 * 60 * 24)
cleanedComplaints$INT_REPORT <-
int_length(interval(date(cleanedComplaints$CMPLNT_FR), cleanedComplaints$RPT_DT)) /
(60 * 60 * 24)

cleanedComplaints$NIGHT <-
ifelse(
cleanedComplaints$CMPLNT_FR_TM >= hms::as.hms(20 * 60 * 60) |
cleanedComplaints$CMPLNT_FR_TM < hms::as.hms(8 * 60 * 60),
1,
0
)
```

Data Storage

For this project I decided to use a SQL database.

Hide

```
complaints_db <- dbConnect(SQLite(), dbname = "complaints.sqlite")
dbWriteTable(
conn = complaints_db,
name = "complaints",
value = cleanedComplaints,
row.names = FALSE,
header = TRUE,
overwrite = TRUE
)
```

Data Retrieval

After the database was created, I queried it several times. suspGender and vicGender retrieves the gender of the suspect/victim involved in each incident. desc retrieves incidents where the total count of the type of incident is greater than 100000. Finally, precCount retrieves the precinct of each incident.

Hide

```
suspGender <-
  dbGetQuery(
  complaints_db,
  "SELECT DISTINCT CMPLNT_NUM,SUSP_SEX FROM complaints WHERE SUSP_SEX = 'M' OR SUSP_SEX = 'F' GR
OUP BY CMPLNT_NUM ORDER BY CMPLNT_NUM"
  )

  vicGender <-
  dbGetQuery(
  complaints_db,
  "SELECT DISTINCT CMPLNT_NUM,VIC_SEX FROM complaints WHERE VIC_SEX = 'M' OR VIC_SEX = 'F' GROUP
BY CMPLNT_NUM ORDER BY CMPLNT_NUM"
  )

  desc <-
  dbGetQuery(
  complaints_db,
  "SELECT DISTINCT CMPLNT_NUM, NIGHT, PD_DESC FROM complaints WHERE PD_DESC IN (SELECT PD_DESC F
ROM complaints GROUP BY PD_DESC HAVING COUNT(*)>100000)  GROUP BY CMPLNT_NUM"
  )

  precCount <-
  dbGetQuery(
  complaints_db,
  "SELECT DISTINCT CMPLNT_NUM, ADDR_PCT_CD FROM complaints GROUP BY CMPLNT_NUM"
  )
  precCount$Borough[precCount$ADDR_PCT_CD %in% c(1:34)] <- "M"
  precCount$Borough[precCount$ADDR_PCT_CD %in% c(40:52)] <- "BRNX"
  precCount$Borough[precCount$ADDR_PCT_CD %in% c(60:94)] <- "BRK"
  precCount$Borough[precCount$ADDR_PCT_CD %in% c(100:115)] <- "QUE"
  precCount$Borough[precCount$ADDR_PCT_CD %in% c(120:125)] <- "ST"


  dbDisconnect(complaints_db)
```

Data Visualization

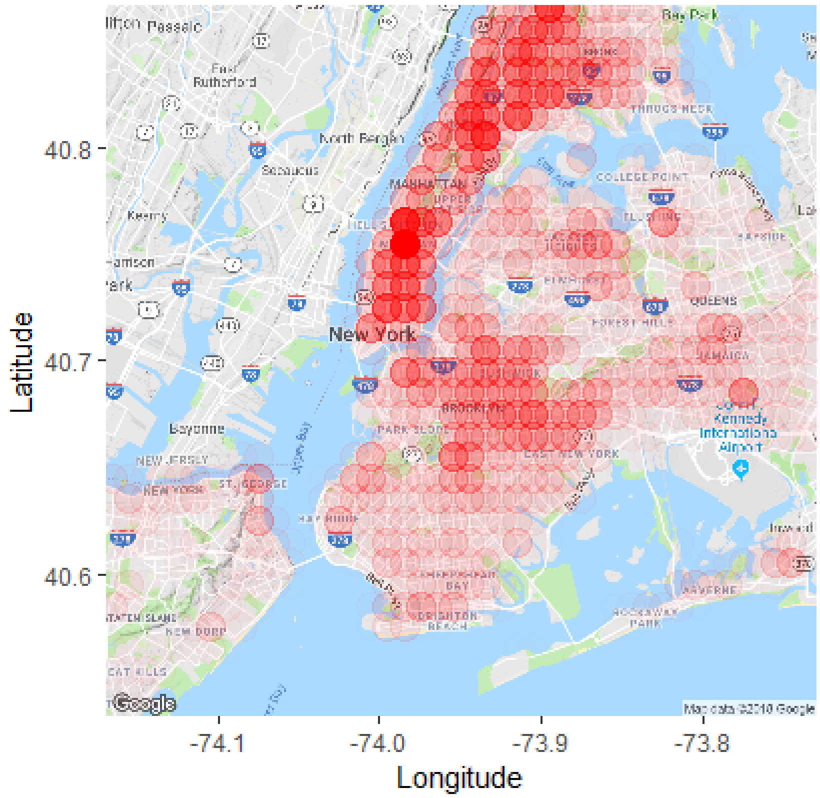
Since each incident contained Latitude and Longitude data, I was able to use ggmap to create a heat map of types of crime, which were then overlayed with a street map of NYC. The opacity of each circle represents the amount of incidents that were reported in that area, relative to the maximum amount of (incidents per area). For instance, if points A, B, and C had 1,2, and 3 incidents respectively, then the circle at point A would have opacity 1/3, the circle at point B would have opacity 2/3, and the circle at point C would have opacity 3/3.

I created these maps for all incidents, indicidents contains the word “harassment” in the description, indicidents contains the word “larceny” in the description, and indicidents contains the word “assault” in the description.

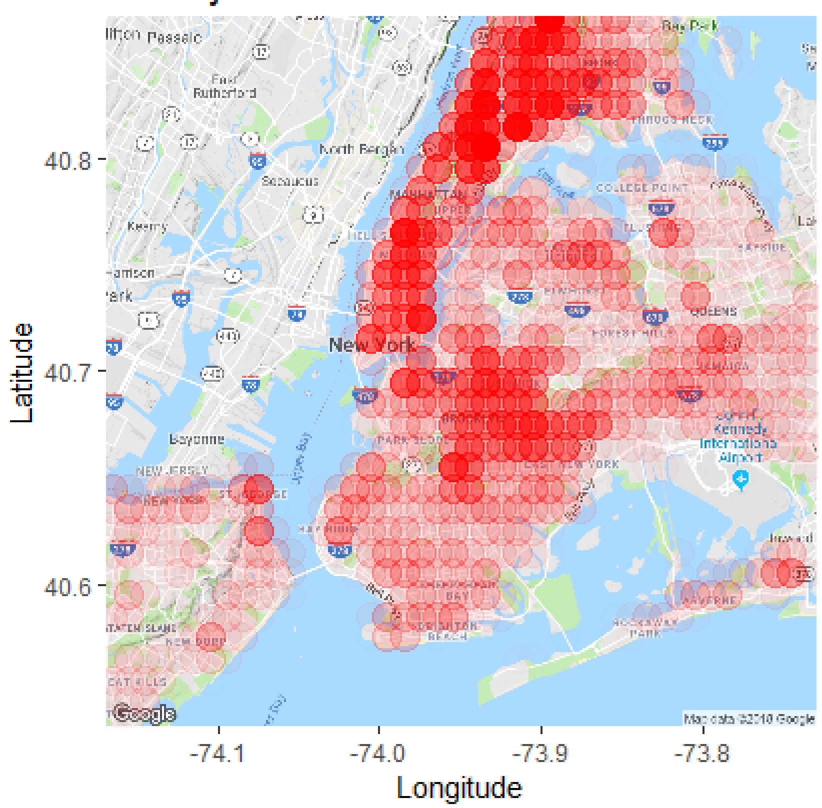
Based on these plots, over the last 11 years, the highest density of incidents occurred in Manhattan, while the highest desnity of harassment incidents occurred in Bronx.

These plots show neighborhoods that contain the highest density of incidents, which could mean that more support is needed in these areas.

Density of All Incidents

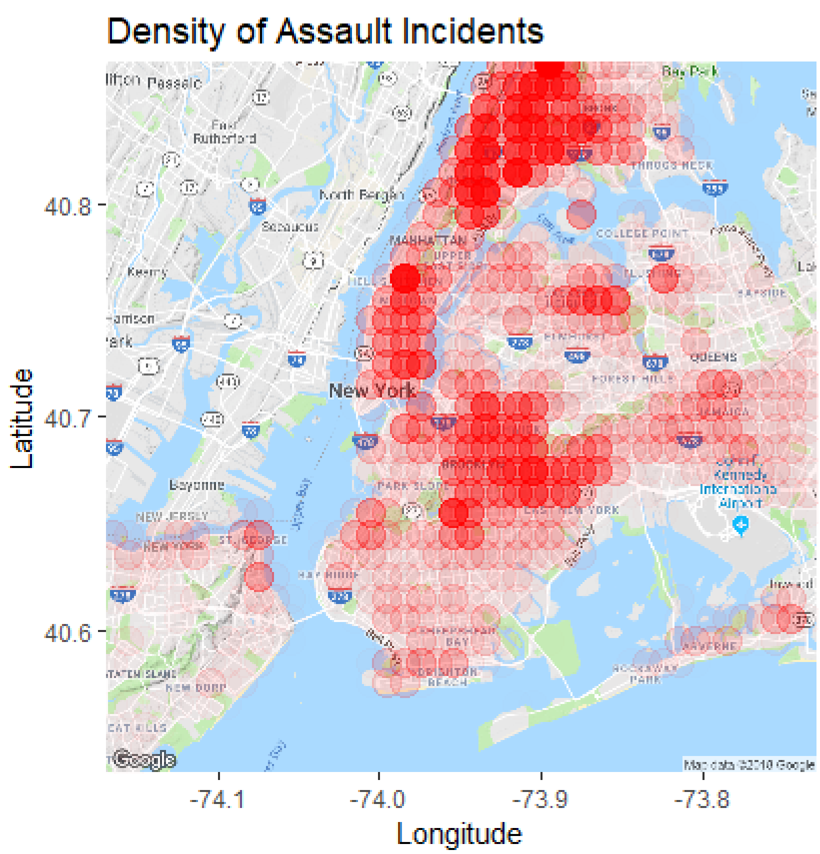


Density of Harassment Incidents



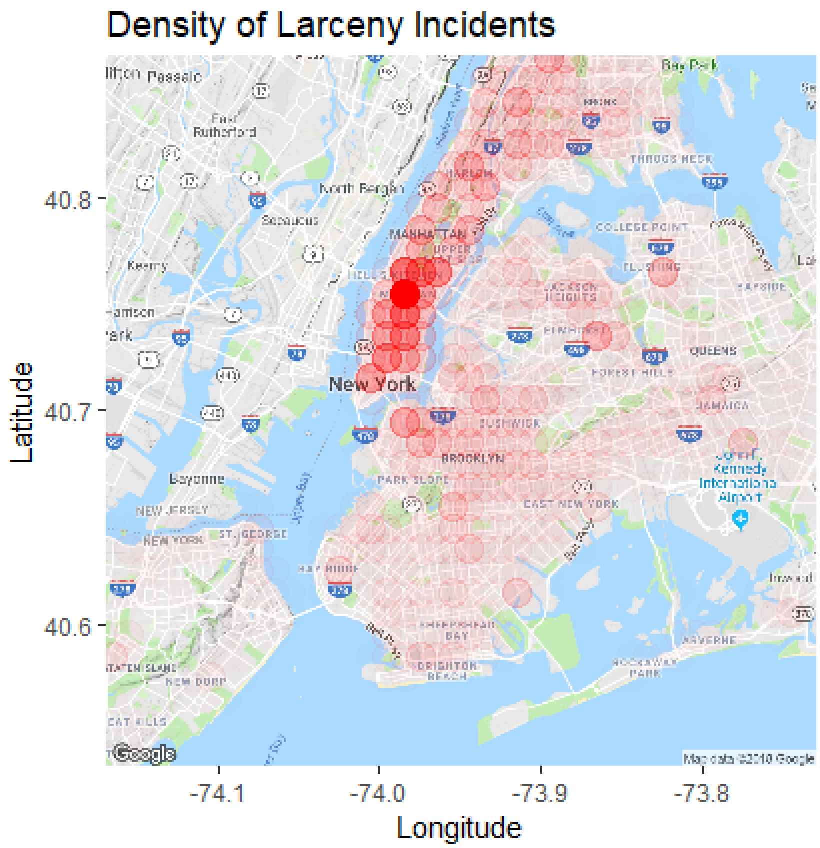
Hide

```
ggmap(nyc_base) + geom_point(
data = complaintClusters,
aes(x = complaintClusters$longitude, y = complaintClusters$latitude),
color = "red",
size = 5,
alpha = complaintsLocCount$locCount / 81090
) + ggtitle("Density of All Incidents") + xlab("Longitude") + ylab("Latitude")
ggmap(nyc_base) + geom_point(
data = harassmentComplaintClusters,
aes(x = harassmentComplaintClusters$longitude, y = harassmentComplaintClusters$latitude),
color = "red",
size = 5,
alpha = harassmentComplaintsLocCount$locCount / 8226
) + ggtitle("Density of Harassment Incidents") + xlab("Longitude") + ylab("Latitude")

ggmap(nyc_base) + geom_point(
data = assaultComplaintClusters,
aes(x = assaultComplaintClusters$longitude, y = assaultComplaintClusters$latitude),
color = "red",
size = 5,
alpha = assaultComplaintsLocCount$locCount / 7307
) + ggtitle("Density of Assault Incidents") + xlab("Longitude") + ylab("Latitude")
```



Hide

```
ggmap(nyc_base) + geom_point(
data = larcenyComplaintClusters,
aes(x = larcenyComplaintClusters$longitude, y = larcenyComplaintClusters$latitude),
color = "red",
size = 5,
alpha = larcenyComplaintsLocCount$locCount / 51451
) + ggtitle("Density of Larceny Incidents") + xlab("Longitude") + ylab("Latitude")
```



Histograms

The following are histograms, using data retrieved from the database.

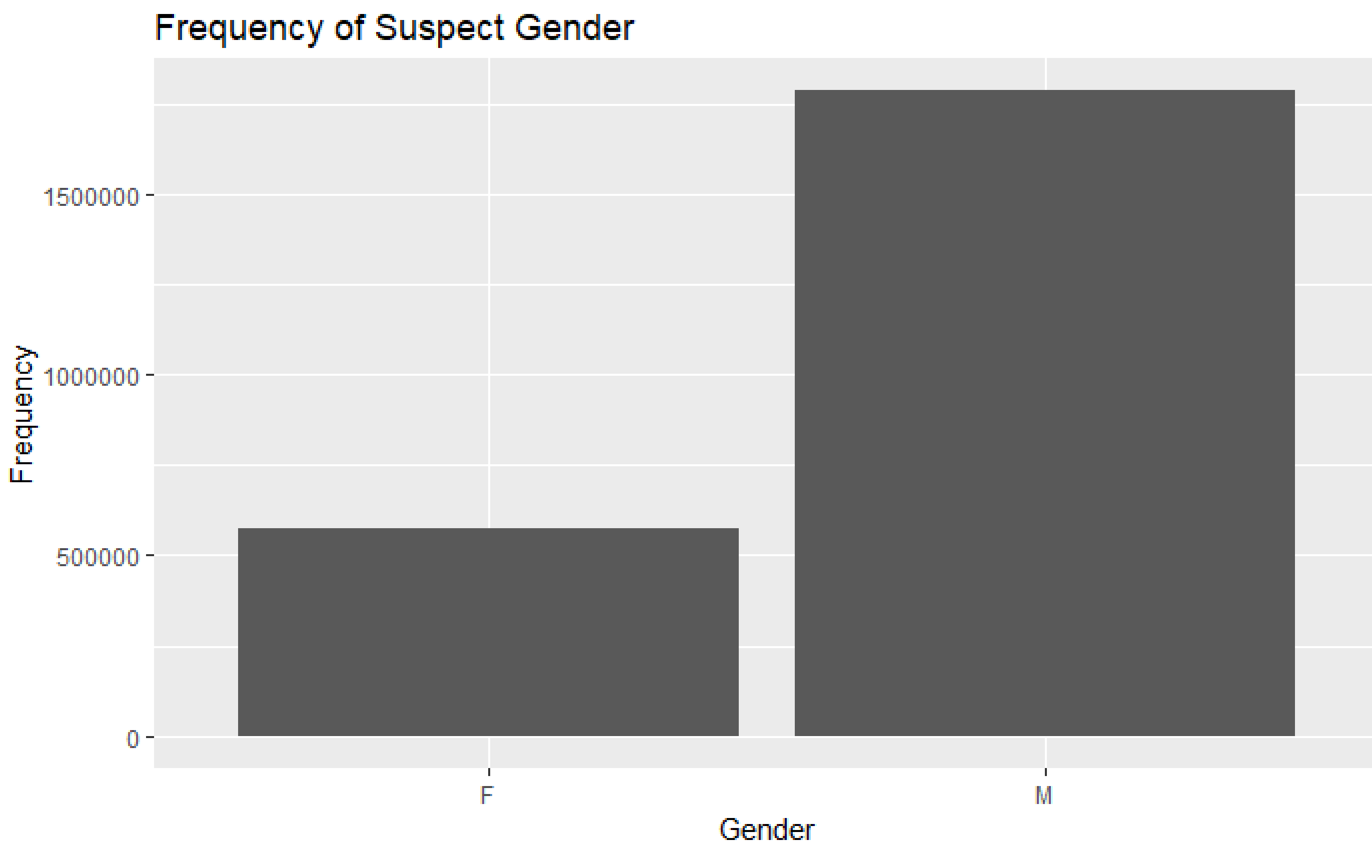
The difference between the number of male and female suspects was not surprising but the similarity between the number of male and female victims was interesting.

I was very surprised when I discovered that there were more incicidents that occurred during the day (between 8AM and 8 PM).

Finally, given the populations of each borough, the final histogram was not surprising. For reference, Brooklyn (BRK) has the highest population at 2.5 million, Bronx (BRNX), Queens (QU), and Manhattan (M) all have around 1.5 million, and Staten Island (St) has 400 thousand.

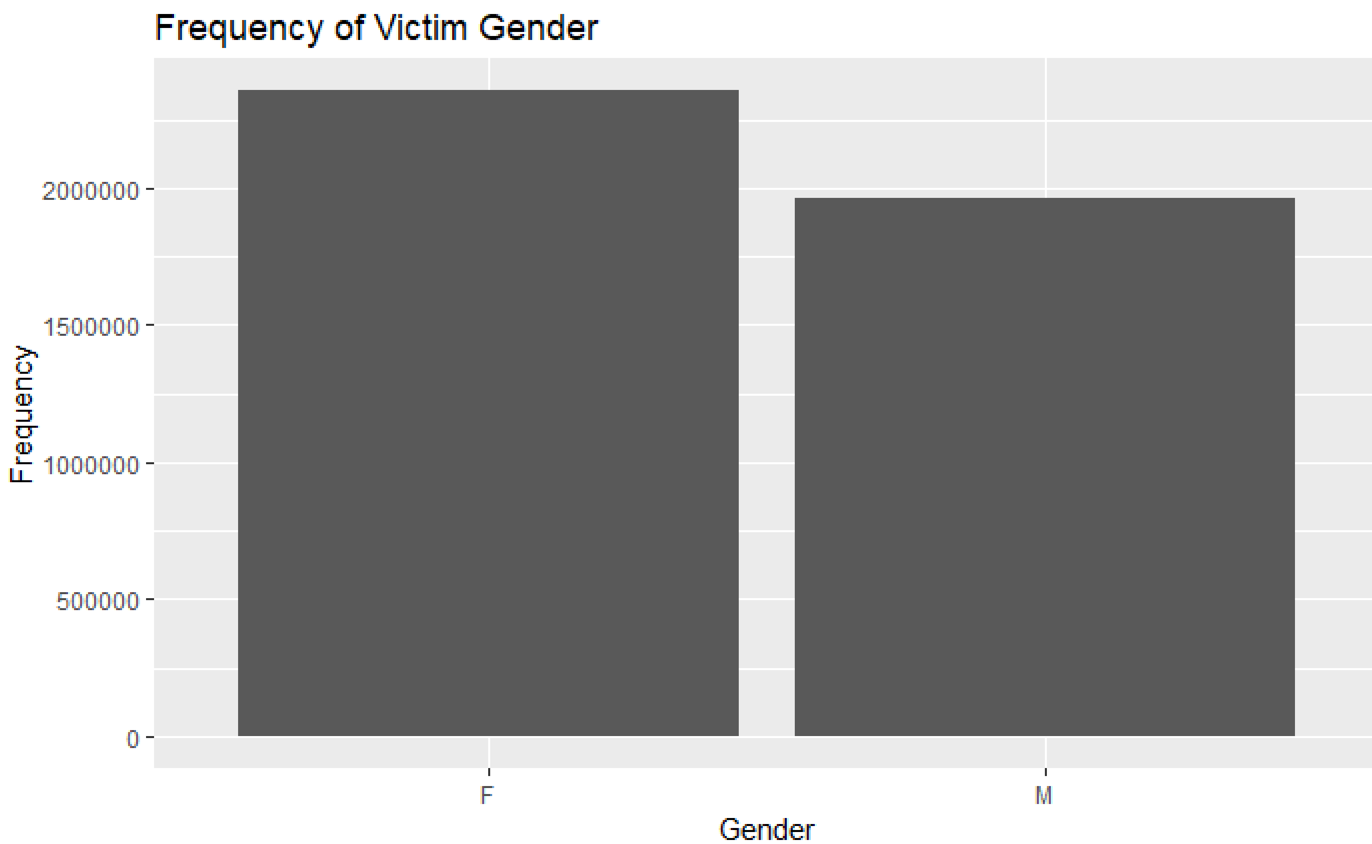
Hide

```
ggplot(data = suspGender) + geom_bar(mapping = aes(x = suspGender$SUSP_SEX, y = ..count..)) +
  ggtitle("Frequency of Suspect Gender") + xlab("Gender") + ylab("Frequency")
```



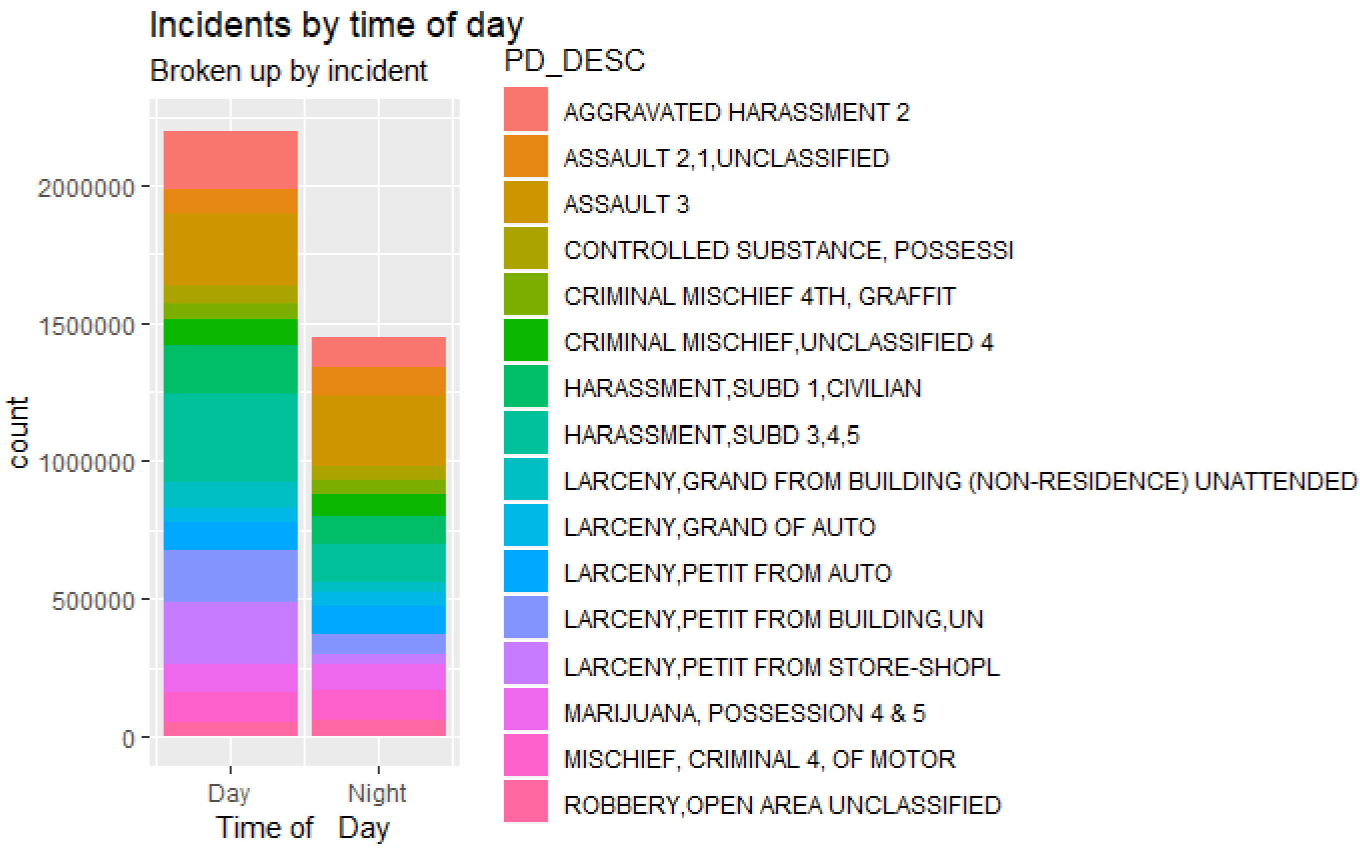
Hide

```
ggplot(data = vicGender) + geom_bar(mapping = aes(x = vicGender$VIC_SEX, y =..count..)) +
  ggtitle("Frequency of Victim Gender") + xlab("Gender") + ylab("Frequency")
```



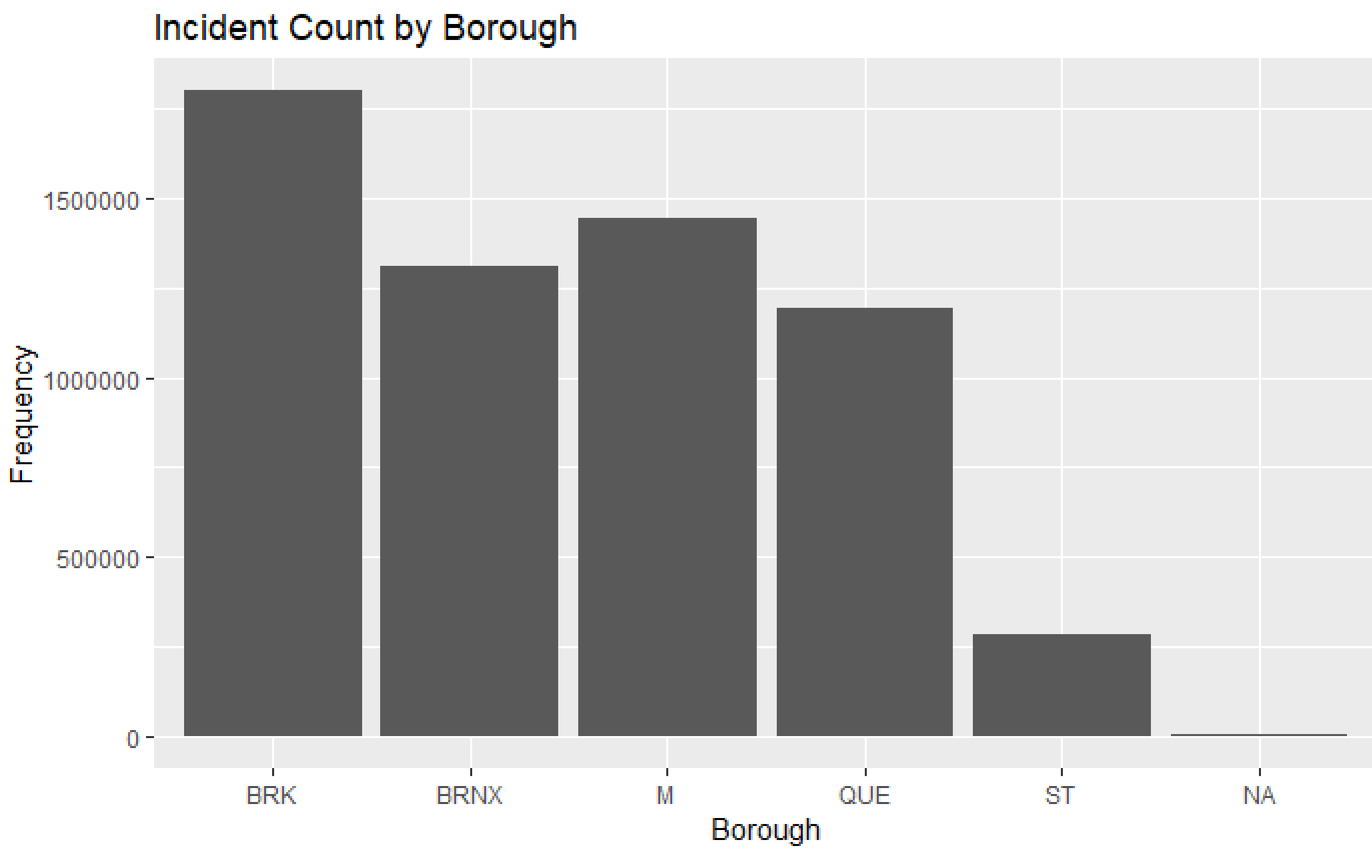
Hide

```
ggplot(data = desc, aes(x = NIGHT, fill = PD_DESC)) + geom_bar() +
  scale_x_continuous("Time of   Day", breaks = c(0, 1), labels = c("Day", "Night")) +
  ggtitle("Incidents by time of day", "Broken up by incident")
```



Hide

```
ggplot(data = precCount) + geom_bar(mapping = aes(x = precCount$Borough, y =..count..)) +
  ggtitle("Incident Count by Borough") + xlab("Borough") + ylab("Frequency")
```



Data Prediction

The purpose of the following is to determine whether the average report time can be predicted. To start, I created a testing dataset and a training data set. The training set contained about 70% of the incidents recording in the cleanedComplaints dataframe. Then, I ran a logistic regression model using some relavant variables in the dataset on the training set. After determining which factors were stastically significant, Using this model, I predicted the outcomes of the incidents in the testing set. The accuracy was 86% with a false positive rate of 14%.

This model allows for the prediction of whether the report time is going to be less than or greater than 3 days. As time elapses, it becomes harder to remember details about the incident. So, if there was a location that had a higher chance of delayed reported, it would be better to be proactive, rather than reactive.

Hide

```
# Boolean - was the duration betwen the incident occurring and the report being filed longer tha
n three days?
cleanedComplaints$wait <- cleanedComplaints$INT_REPORT > 3
# Separated the data into training and testing data
trainComplaints <-
cleanedComplaints[sample(1:nrow(cleanedComplaints), floor(.7 * nrow(cleanedComplaints))), ]
testComplaints <-
anti_join(cleanedComplaints, trainComplaints, by = "CMPLNT_NUM")
# Linear Model
trialDelayReport.glm <-
glm(
wait ~ ADDR_PCT_CD + VIC_SEX + PD_CD  + CRM_ATPT_CPTD_CD +
SUSP_RACE + SUSP_SEX,
data = cleanedComplaints,
family = binomial
)
summary(trialDelayReport.lm)
```

```
Call:
lm(formula = wait ~ ADDR_PCT_CD + VIC_SEX + PD_CD + CRM_ATPT_CPTD_CD + 
    SUSP_RACE + SUSP_SEX, data = cleanedComplaints, family = binomial)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.2647 -0.1554 -0.1239 -0.0761  1.0338 

Coefficients:
                                    Estimate Std. Error  t value Pr(>|t|)    
(Intercept)                        1.275e-01  3.718e-03   34.286  < 2e-16 ***
ADDR_PCT_CD                        5.760e-05  3.996e-06   14.411  < 2e-16 ***
VIC_SEXM                          -2.411e-02  3.268e-04  -73.800  < 2e-16 ***
VIC_SEXUNKNOWN                    -9.319e-02  3.560e-04 -261.796  < 2e-16 ***
PD_CD                              1.082e-04  6.469e-07  167.183  < 2e-16 ***
CRM_ATPT_CPTD_CDCOMPLETED          2.315e-02  1.048e-03   22.092  < 2e-16 ***
SUSP_RACEASIAN / PACIFIC ISLANDER -1.352e-02  3.691e-03   -3.663 0.000249 ***
SUSP_RACEBLACK                    -5.050e-02  3.532e-03  -14.297  < 2e-16 ***
SUSP_RACEBLACK HISPANIC           -3.607e-02  3.624e-03   -9.955  < 2e-16 ***
SUSP_RACEOTHER                     4.013e-02  1.009e-01    0.398 0.690723    
SUSP_RACEUNKNOWN                   7.961e-03  3.566e-03    2.233 0.025579 *  
SUSP_RACEWHITE                    -4.322e-03  3.565e-03   -1.212 0.225420    
SUSP_RACEWHITE HISPANIC           -3.717e-02  3.550e-03  -10.471  < 2e-16 ***
SUSP_SEXM                         -3.139e-02  5.092e-04  -61.643  < 2e-16 ***
SUSP_SEXUNKNOWN                   -2.612e-02  7.325e-04  -35.653  < 2e-16 ***
---
Signif. codes:  0 �***� 0.001 �**� 0.01 �*� 0.05 �.� 0.1 � � 1

Residual standard error: 0.3343 on 6031198 degrees of freedom
  (5592 observations deleted due to missingness)
Multiple R-squared:  0.01771,	Adjusted R-squared:  0.01771 
F-statistic:  7767 on 14 and 6031198 DF,  p-value: < 2.2e-16
```

Hide

```
delayReport.glm <-
glm(wait ~ ADDR_PCT_CD + VIC_SEX + PD_CD ,
data = cleanedComplaints,
family = binomial)
summary(delayReport.lm)
```

```

Call:
lm(formula = wait ~ ADDR_PCT_CD + VIC_SEX + PD_CD, data = cleanedComplaints,
    family = binomial)

Residuals:
     Min       1Q   Median       3Q      Max
-0.22186 -0.14980 -0.12645 -0.08367  0.95871

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.123e-01  4.270e-04  263.08   <2e-16 ***
ADDR_PCT_CD     7.140e-05  3.991e-06   17.89   <2e-16 ***
VIC_SEXM       -1.925e-02  3.251e-04  -59.23   <2e-16 ***
VIC_SEXUNKNOWN -8.169e-02  3.421e-04 -238.80   <2e-16 ***
PD_CD           1.047e-04  6.404e-07  163.52   <2e-16 ***
---
Signif. codes:  0 �***� 0.001 �**� 0.01 �*� 0.05 �.� 0.1 � � 1

Residual standard error: 0.3352 on 6031215 degrees of freedom
  (5585 observations deleted due to missingness)
Multiple R-squared:  0.01246,   Adjusted R-squared:  0.01246
F-statistic: 1.902e+04 on 4 and 6031215 DF,  p-value: < 2.2e-16
```

Hide

```
testComplaints$predictedProb <-
predict(delayReport.lm, testComplaints, type = "response")
testComplaints$predictedLongWait <-
ifelse(testComplaints$predictedProb > .5, 1, 0)
numRight <- nrow(filter(testComplaints, wait == predictedLongWait))
numRows <- nrow(testComplaints)
accuracy <- numRight / numRows
sprintf("Accuracy rate = %f", 100 * accuracy)
```

```
[1] "Accuracy rate = 86.782305"
```

Hide

```
numFalsePos <- nrow(filter(testComplaints, wait > predictedLongWait))
numFalseNeg <- nrow(filter(testComplaints, wait < predictedLongWait))
falsePos <- numFalsePos / numRows
falseNeg <- numFalseNeg / numRows
sprintf("Number of observations = %f", numRows)
```

```
[1] "Number of observations = 1811042.000000"
```

Hide

```
sprintf("Number of false positives = %f", numFalsePos)
```

```
[1] "Number of false positives = 237682.000000"
```

Hide

```
sprintf("Number of false negatives = %f", numFalseNeg)
```

```
[1] "Number of false negatives = 0.000000"
```

Hide

```
sprintf("False Positives = %f", falsePos)
```

```
[1] "False Positives = 0.131240"
```

Hide

```
sprintf("False Negatives = %f", falseNeg)
```

```
[1] "False Negatives = 0.000000"
```

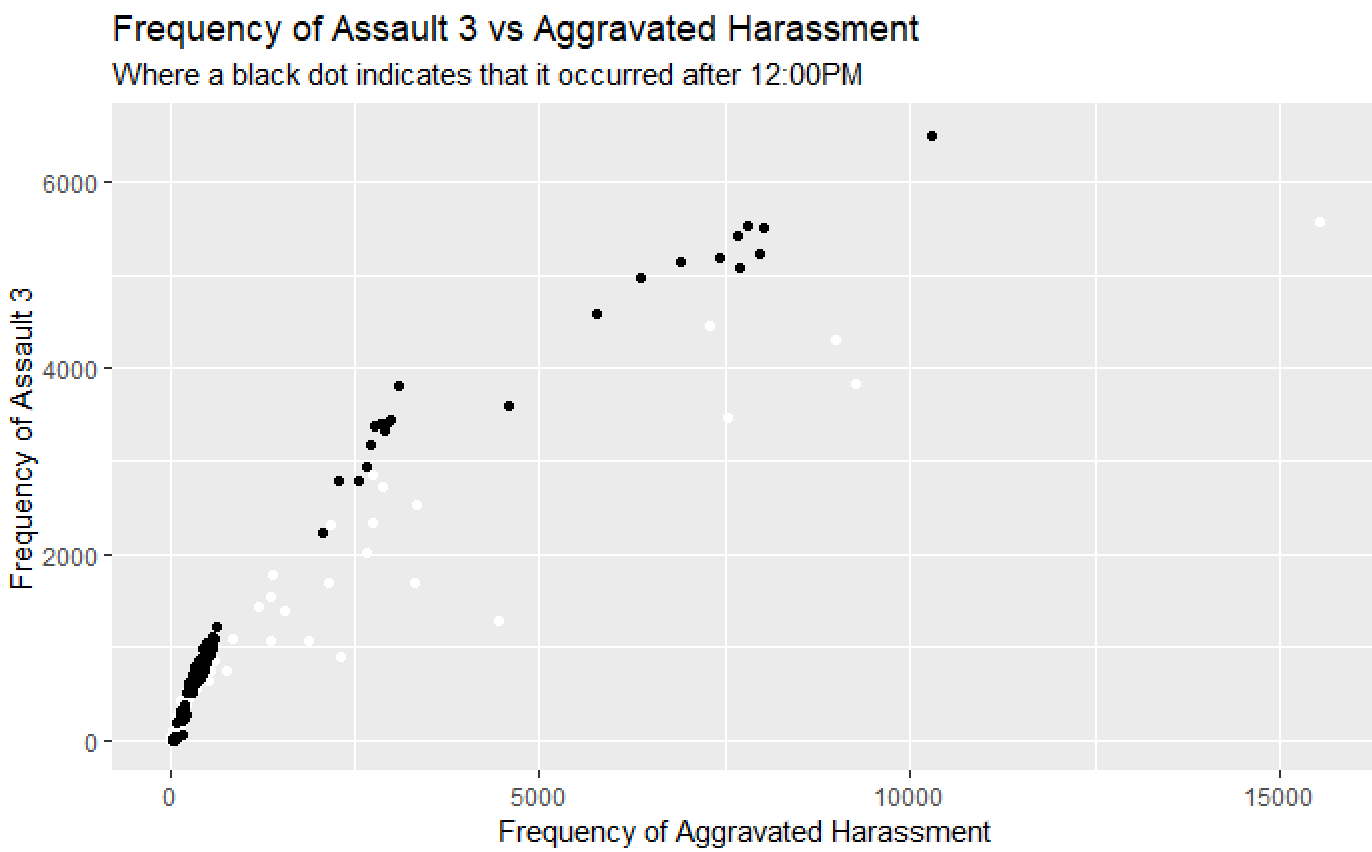
In order to narrow my cleanedComplaints dataframe, I created a new dataframe that would allow me to decide how many of the most common types of incidents I wanted to analyze. In the end, I decided to go with the five most frequent incidents: aggravated harassment, assault 3, assault 2, possession of a controlled substance, and graffiti.

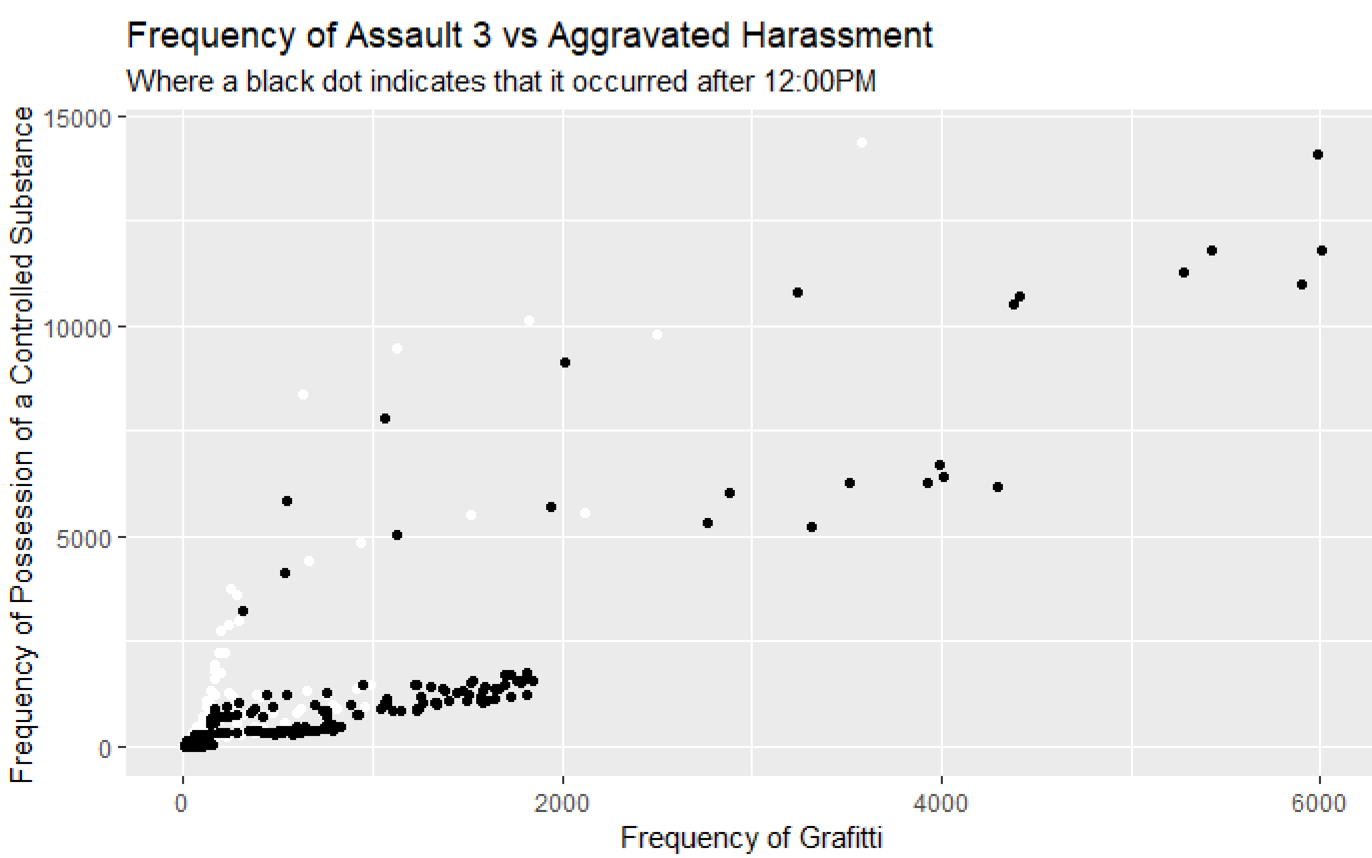
The topComplaints dataframe is used for the rest of this project.

Hide

```
  typesComplaintCounts <-
  summarise(group_by(desc, PD_DESC), IncidentCount = length(PD_DESC))
  typesComplaintCounts <-
  typesComplaintCounts[order(typesComplaintCounts$IncidentCount, decreasing = TRUE),]
  top5Complaints <- flatten(typesComplaintCounts[1:5, 1])
  topComplaints <-
  filter(cleanedComplaints, PD_DESC %in% top5Complaints)
  topComplaints<-filter(topComplaints,!is.na(topComplaints$CMPLNT_FR_TM))
```

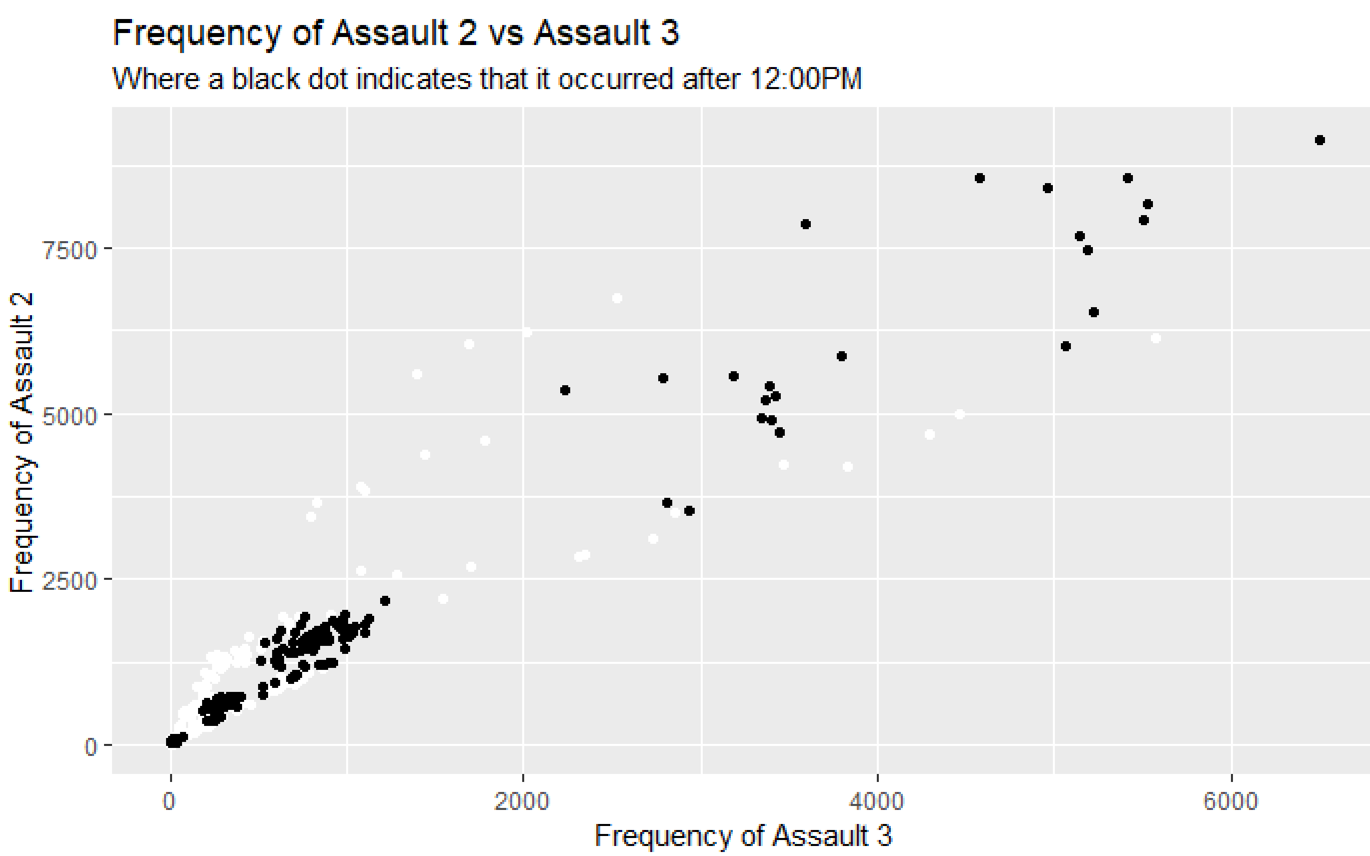
The purpose of the following is to determine whether there is a correlation between the number of different incidents that occur at a same time. I expected that there was not going to be a correlation because it seemed that, for instance, the people who were involved in a graffiti incident would be independent from those involved in an aggravated harassment incident. However, based on these results, it appears that there is a positive correlation between all types of incidents, which may require more in depth research. The correlation plot represents the Pearson’s correlation coefficient between each type of incident.





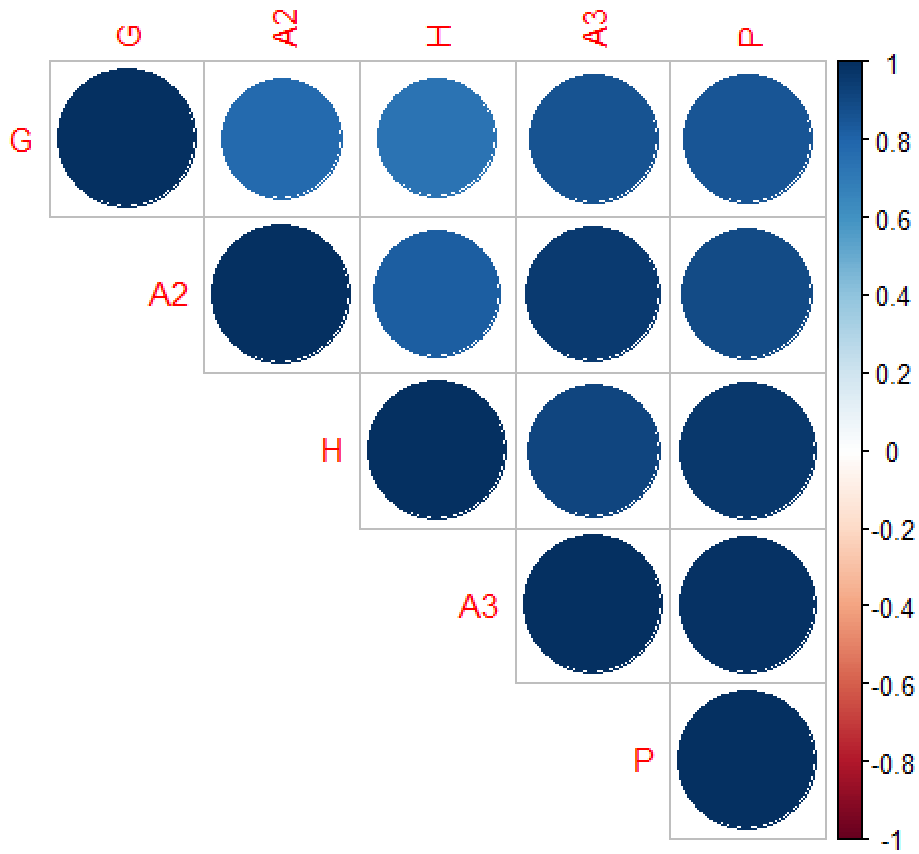
Hide

```
ggplot(complaintsTime_df, aes(AGGRAVATED_HARASSMENT, ASSAULT_3)) + geom_point(col =
 colr) + xlab("Frequency of Aggravated Harassment") + ylab("Frequency of Assault 3") +
 ggtitle(
 "Frequency of Assault 3 vs Aggravated Harassment",
 "Where a black dot indicates that it occurred after 12:00PM"
 )
 ggplot(complaintsTime_df,
 aes(GRAFITTI, POSSESSION_CONTROLLED_SUBSTANCE)) + geom_point(col = colr) + xlab("Frequency of
Grafitti") +
 ylab("Frequency of Possession of a Controlled Substance") + ggtitle(
 "Frequency of Assault 3 vs Aggravated Harassment",
 "Where a black dot indicates that it occurred after 12:00PM"
 )

 ggplot(complaintsTime_df,
 aes(ASSAULT_3, ASSAULT_2)) + geom_point(col = colr) + xlab("Frequency of Assault 3") +
 ylab("Frequency of Assault 2") + ggtitle(
 "Frequency of Assault 2 vs Assault 3",
 "Where a black dot indicates that it occurred after 12:00PM"
 )
```



Hide

```
 cor_df <- cor(complaintsTime_df[, c(2:6)])
 rownames(cor_df) <- c("H", "A2", "A3", "P", "G")
 colnames(cor_df) <- c("H", "A2", "A3", "P", "G")
 corrplot(cor_df, order = "hclust", type = "upper")
```



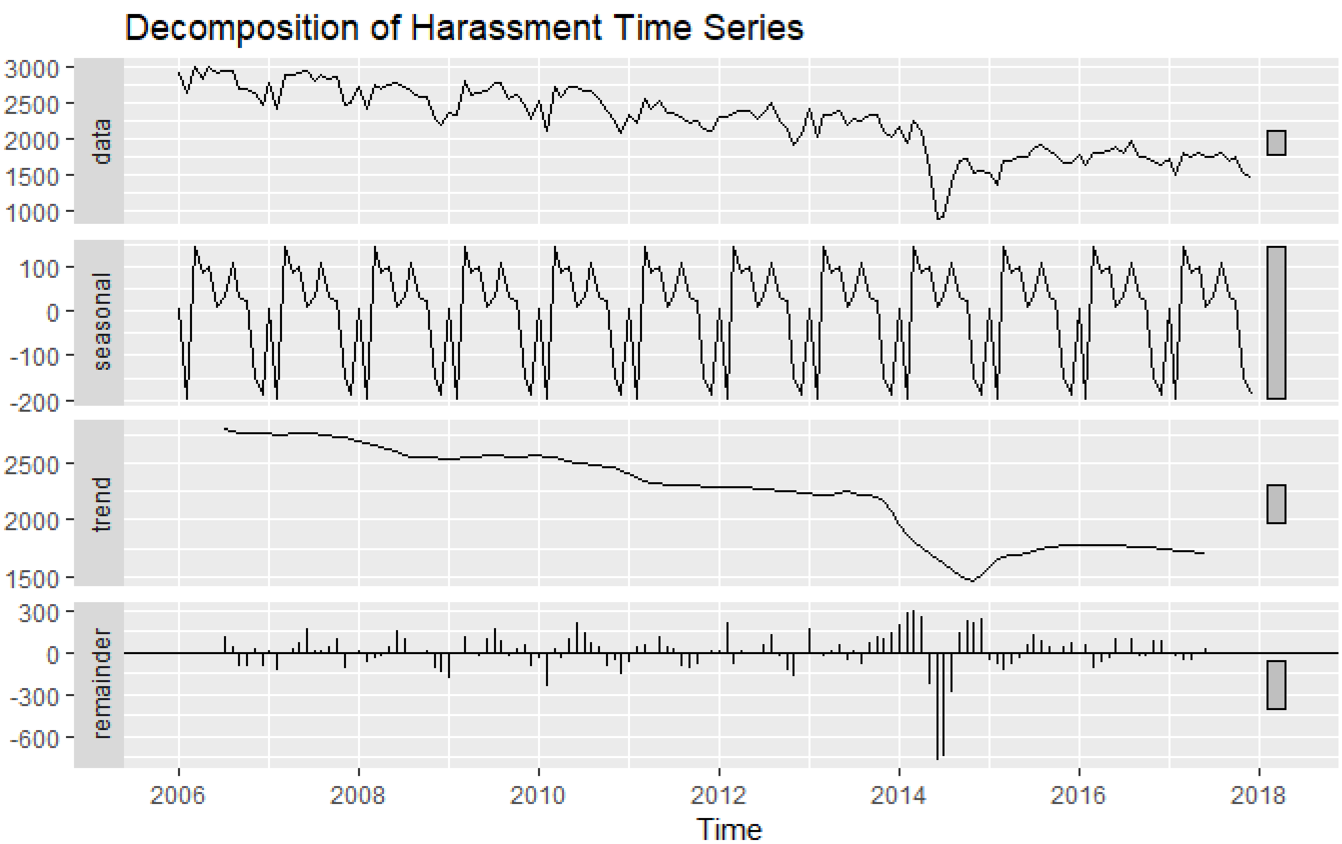
Time Series

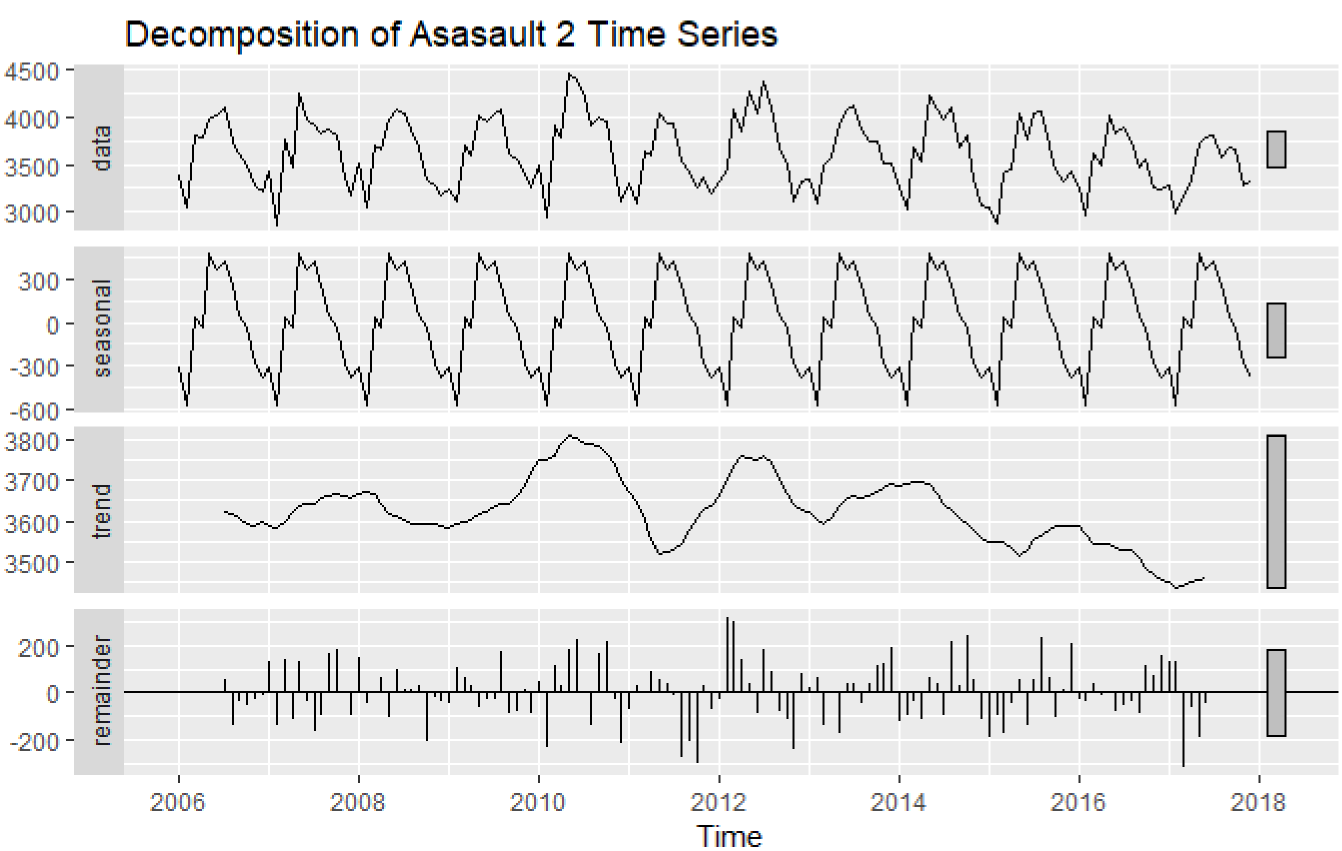
The following two chunks are to determine whether there are seasonal trends in the frequencies of different types of incidents.

First, I create a new dataframe that contains the unique dates in the topComplaints dataframe, as well as the count of each incident for that date. In order to broadly look at the data, I created several dummy variables that recorded in which quarter the incident occurred. For instance, if an incident occurred in January, the newly created Q1 variable would contain a 1, and the other three variables (Q2, Q3, and Q3) would contain a 0.

Using this dataframe, I created time series data for each incident, which I then plotted.

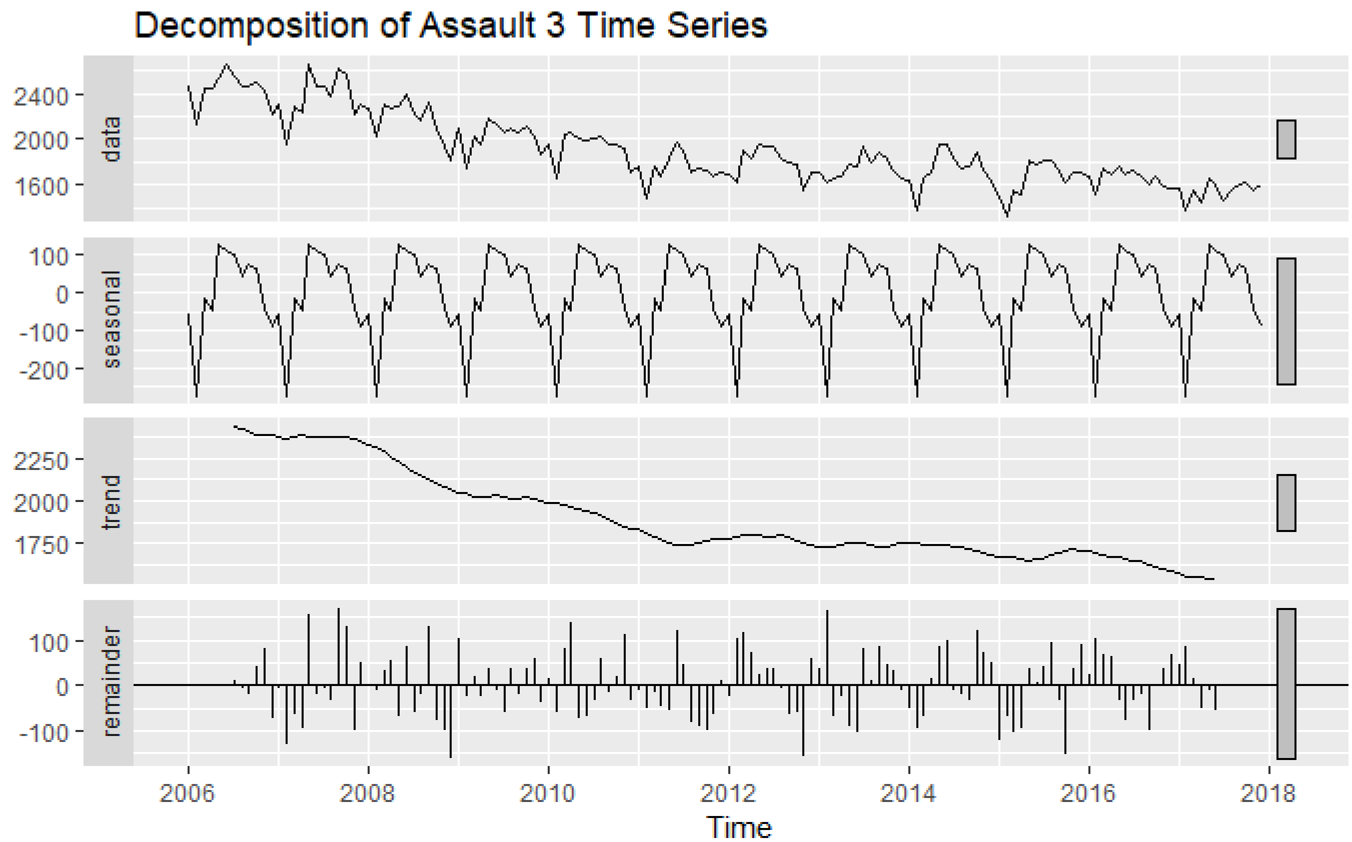
Based on the plots, it is clear that there is a seasonal trend for each type of incident. It is also easy to determine that there is a gradual downward trend in harassment and assault 2, but an upward trend in the number of graffiti incidents. The NYPD has grown in size over the last few years which may account for lower number of serious offenses.


Decomposition of Harassment Time Series
data
seasonal
trend
remainder
3000
2500
2000
1500
1000
100
0
-100
-200
2500
2000
1500
300
0
-300
-600
2006
2008
2010
2012
2014
2016
2018
Time


Decomposition of Asasault 2 Time Series
data
seasonal
trend
remainder
4500
4000
3500
3000
300
0
-300
-600
3800
3700
3600
3500
200
0
-200
2006
2008
2010
2012
2014
2016
2018
Time

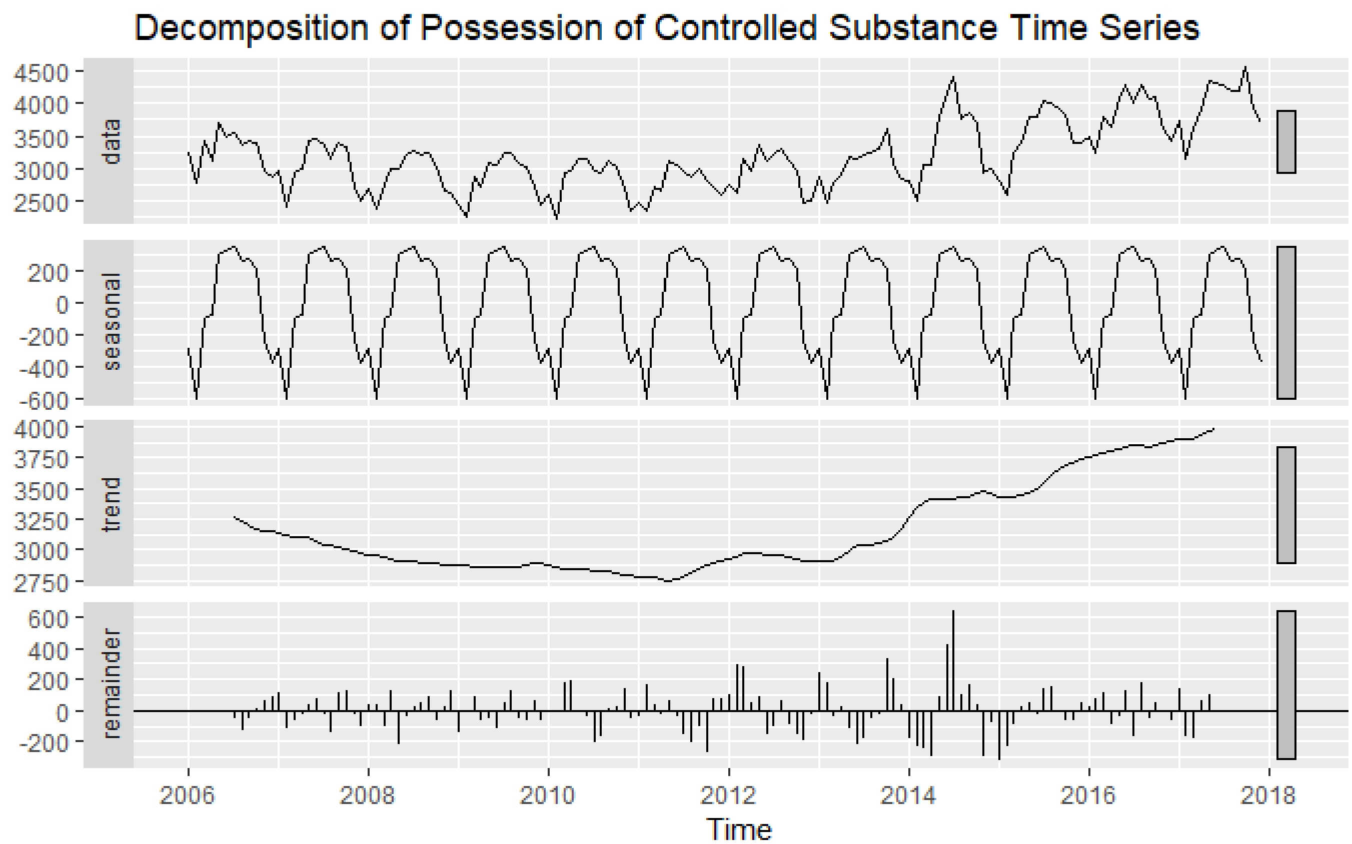
Hide

```
# Plots
autoplot(harassment_decomp, main="Decomposition of Harassment Time Series")
autoplot(assault2_decomp, main="Decomposition of Asasault 2 Time Series")

autoplot(assault3_decomp, main="Decomposition of Assault 3 Time Series")
```



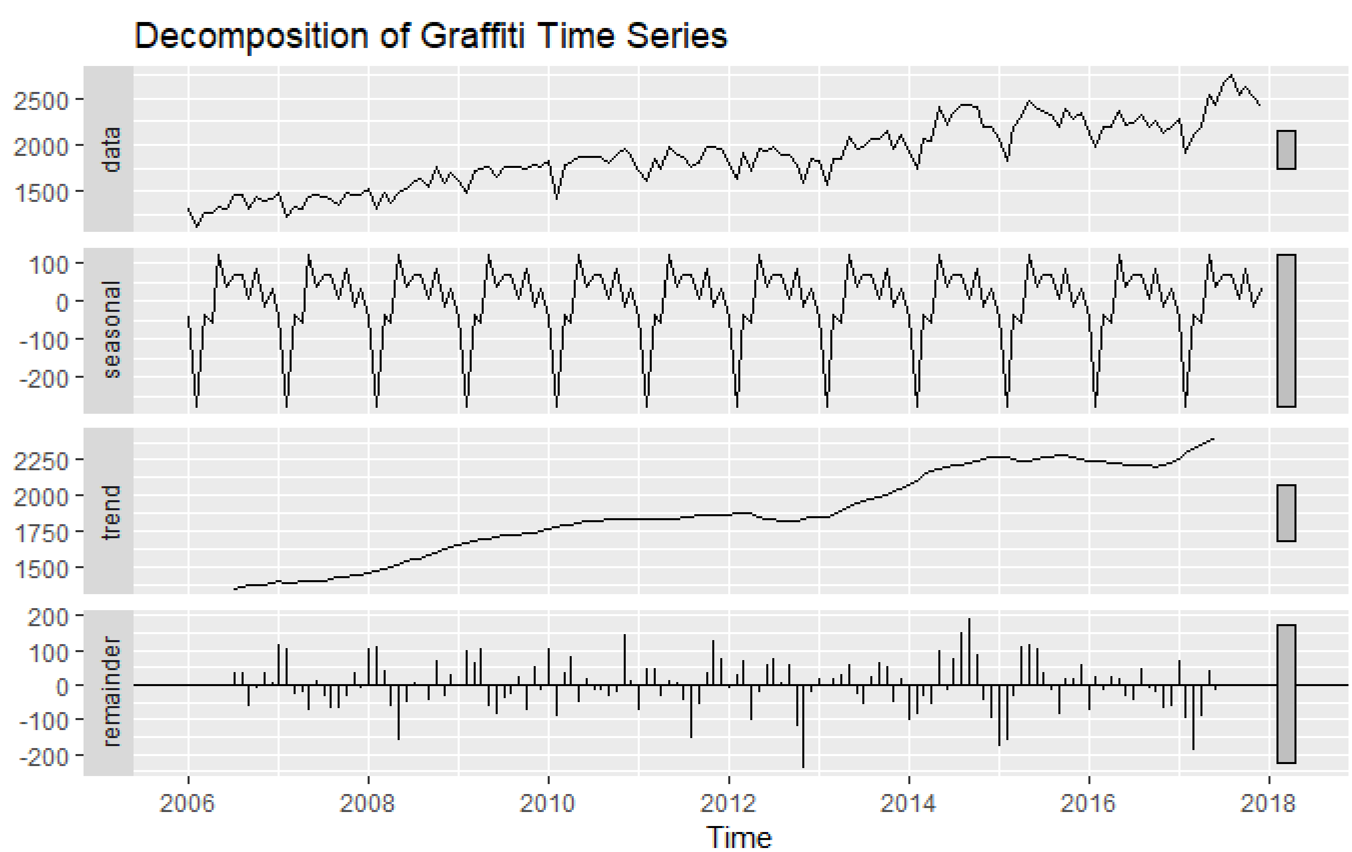
Hide

```
autoplot(substances_decomp, main="Decomposition of Possession of Controlled Substance Time Ser
ies")
```



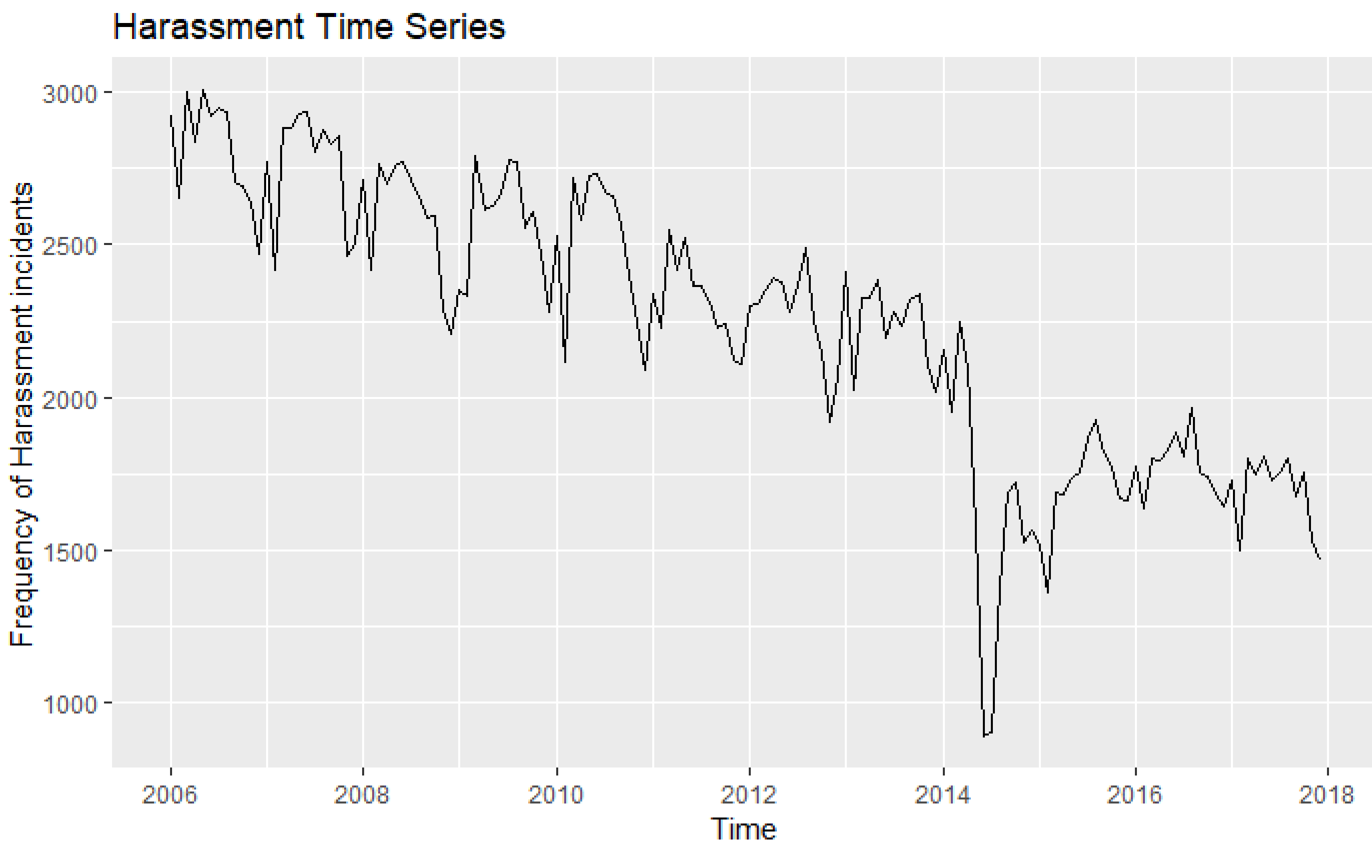
Hide

```
autoplot(grafitti_decomp, main="Decomposition of Graffiti Time Series")
```



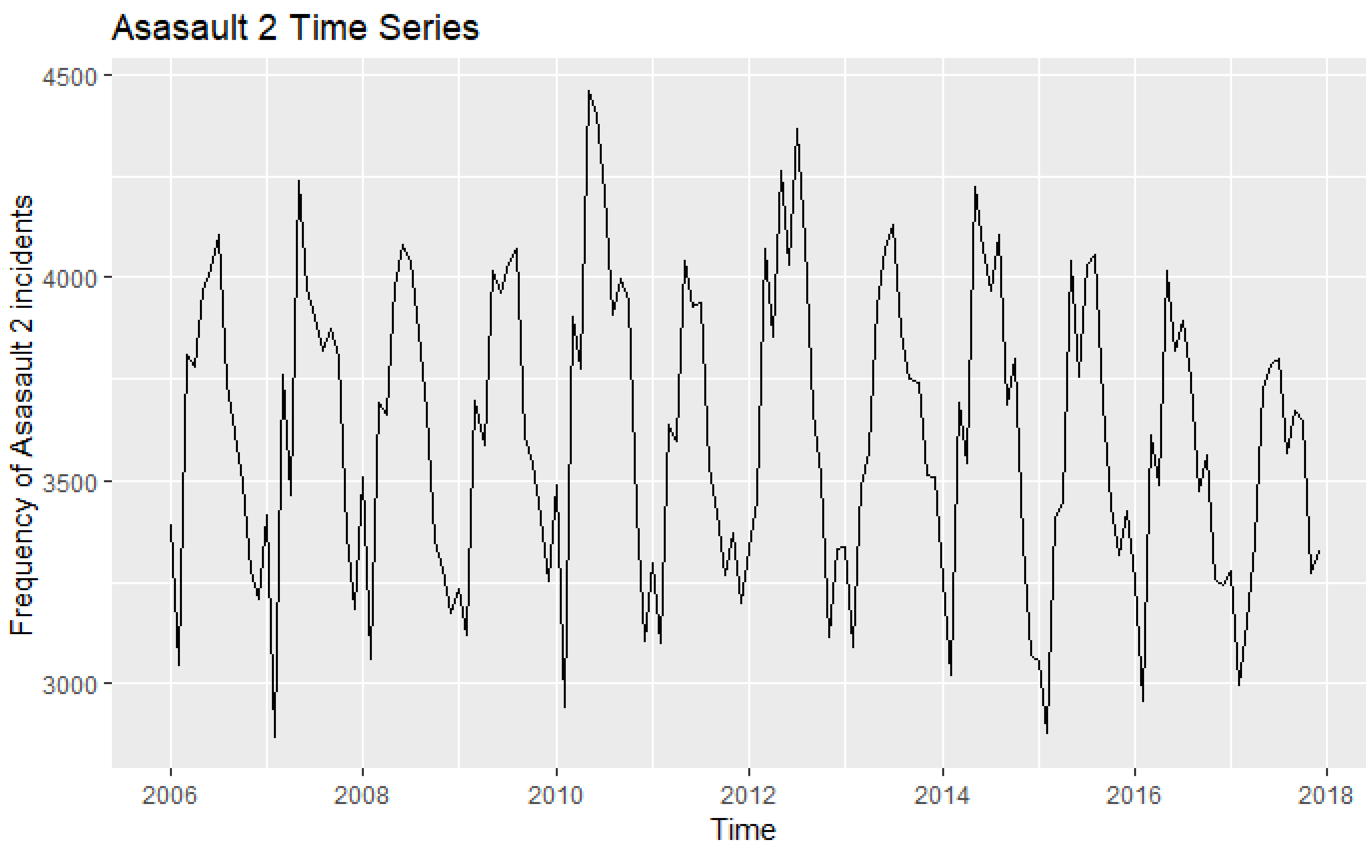
Hide

```
autoplot(harassment_ts, ylab="Frequency of Harassment incidents",main="Harassment Time Series"
)
```



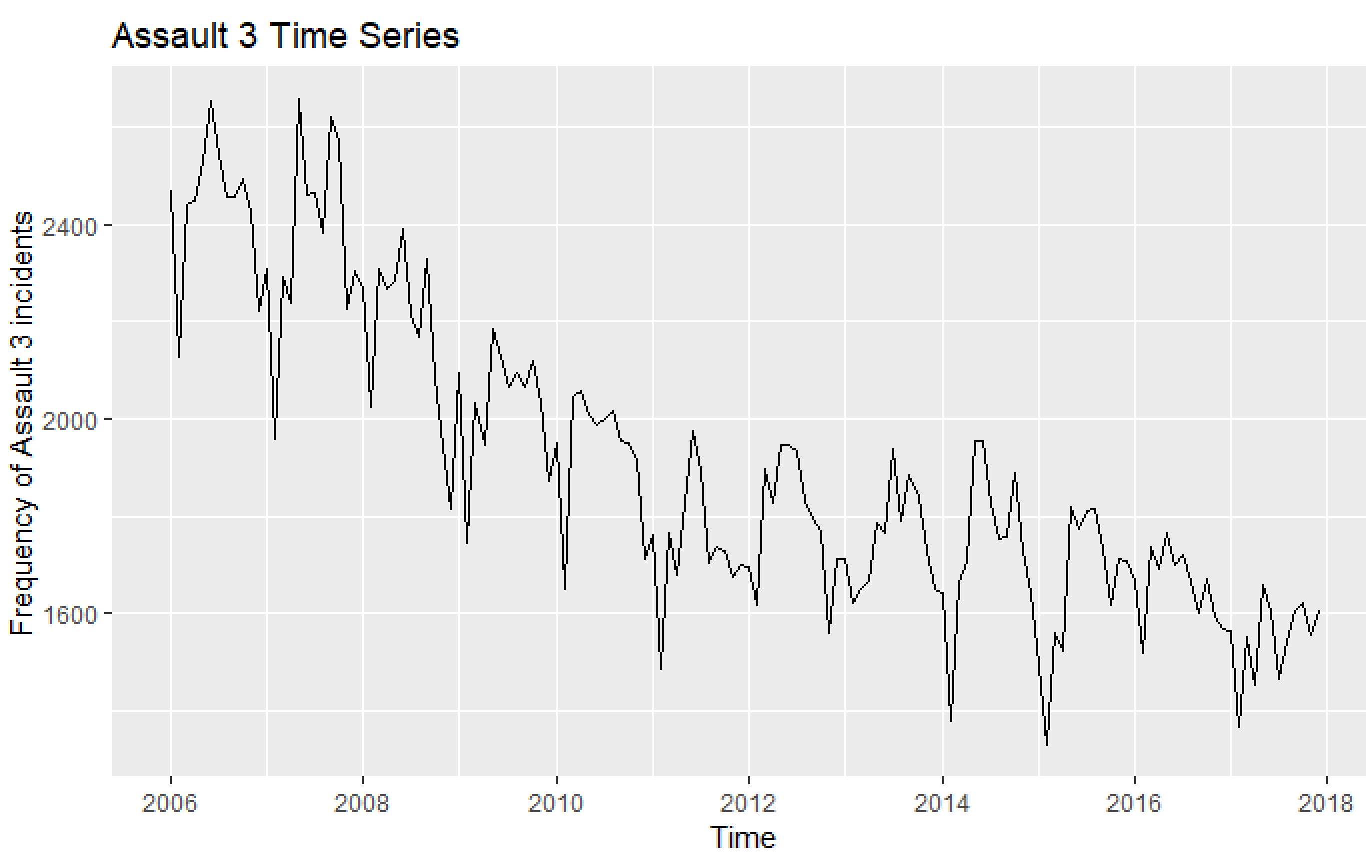
Hide

```
autoplot(assault2_ts, ylab="Frequency of Asasault 2 incidents",main="Asasault 2 Time Series")
```



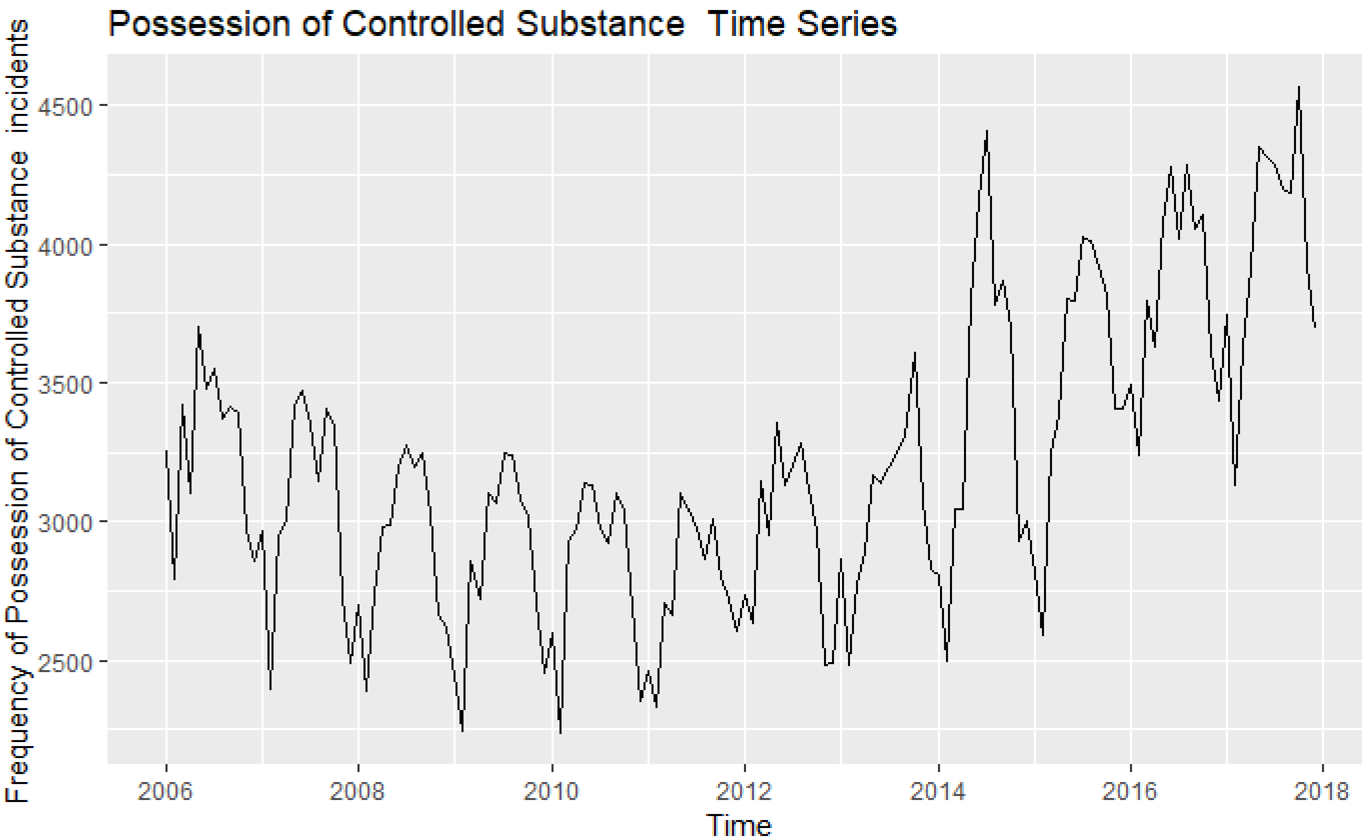
Hide

```
autoplot(assault3_ts, ylab="Frequency of Assault 3 incidents",main="Assault 3 Time Series")
```



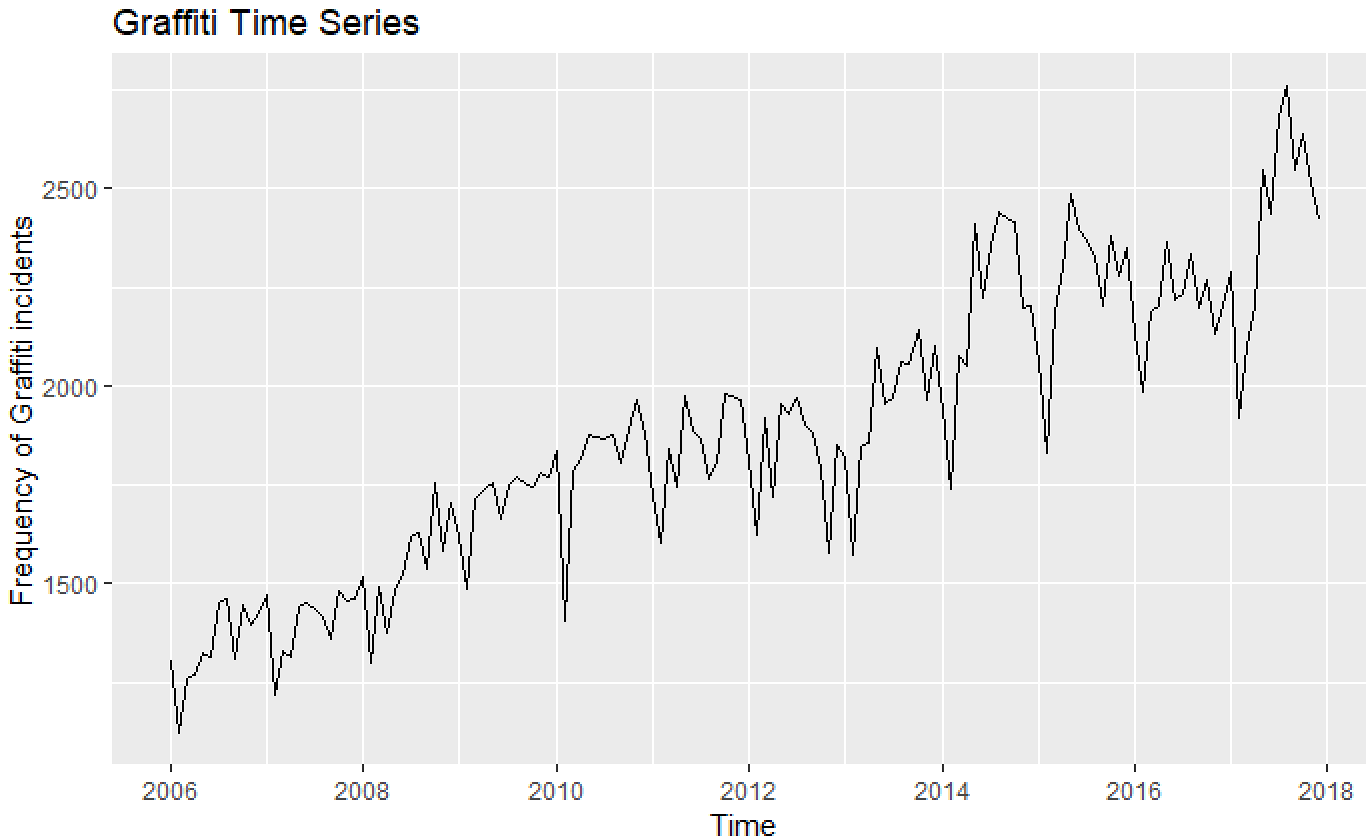
Hide

```
autoplot(substances_ts, ylab="Frequency of Possession of Controlled Substance  incidents",main
="Possession of Controlled Substance  Time Series")
```



Hide

```
autoplot(grafitti_ts, ylab="Frequency of Graffiti incidents",main="Graffiti Time Series")
```



Hide

```
NA
```

Predictive Time Series

The time series models previously created can be used as predictive tools. Using Holt-Winters, I could forecast trends for the different types of incidents. In addition, I created a linear model, using quarters, which predicts the frequency of harassment incidents, given the number of months and the quarter. This model uses 70% of the observations from the complaintsByMonth dataframe to train and then was tested on the remaining 30%, and has an accuracy of 95% when the actual frequency is within one standard deviation of the predicted frquency.

Hide

```
# HoltWinters model of the time series data
harassmentHW <- holt(harassment_ts, level = .95)
assault2HW<-holt(assault2_ts,level=.95)
grafittiHW <- holt(grafitti_ts,level=.95)
summary(assault2HW)
```

```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = assault2_ts, level = 0.95)

  Smoothing parameters:
    alpha = 0.817
    beta  = 1e-04

  Initial states:
    l = 3655.7527
    b = 0.9985

  sigma:  308.9059

     AIC     AICc      BIC
2372.711 2373.146 2387.560

Error measures:
                    ME     RMSE      MAE       MPE     MAPE     MASE        ACF1
Training set -3.969813 304.5854 236.6919 -0.542587 6.596856 1.244854 -0.05597808

Forecasts:
```

	Point Forecast <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
Jan 2018	3331.604	2726.160	3937.049
Feb 2018	3332.545	2550.693	4114.398
Mar 2018	3333.487	2408.239	4258.734
Apr 2018	3334.428	2285.175	4383.682
May 2018	3335.369	2175.265	4495.474
Jun 2018	3336.311	2075.036	4597.585
Jul 2018	3337.252	1982.319	4692.185
Aug 2018	3338.193	1895.650	4780.737
Sep 2018	3339.135	1813.985	4864.284
Oct 2018	3340.076	1736.551	4943.600

1-10 of 10 rows

Hide

```r
summary(harassmentHW)
```

```

Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = harassment_ts, level = 0.95)

  Smoothing parameters:
    alpha = 0.7103
    beta  = 1e-04

  Initial states:
    l = 2920.527
    b = -8.5935

  sigma:  185.5984

     AIC     AICc      BIC
2225.989 2226.424 2240.838

Error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set -1.767103 183.0025 133.6949 -0.7819118 6.703389 0.6846574 0.003983976

Forecasts:
```

	Point Forecast <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
Jan 2018	1492.606	1128.8398	1856.372
Feb 2018	1483.987	1037.7812	1930.193
Mar 2018	1475.368	959.7206	1991.015
Apr 2018	1466.749	889.9444	2043.554
May 2018	1458.130	826.0432	2090.217
Jun 2018	1449.511	766.5888	2132.433
Jul 2018	1440.892	710.6520	2171.132
Aug 2018	1432.273	657.5879	2206.958
Sep 2018	1423.654	606.9273	2240.381
Oct 2018	1415.035	558.3165	2271.754

1-10 of 10 rows

Hide

```
summary(grafittiHW)
```

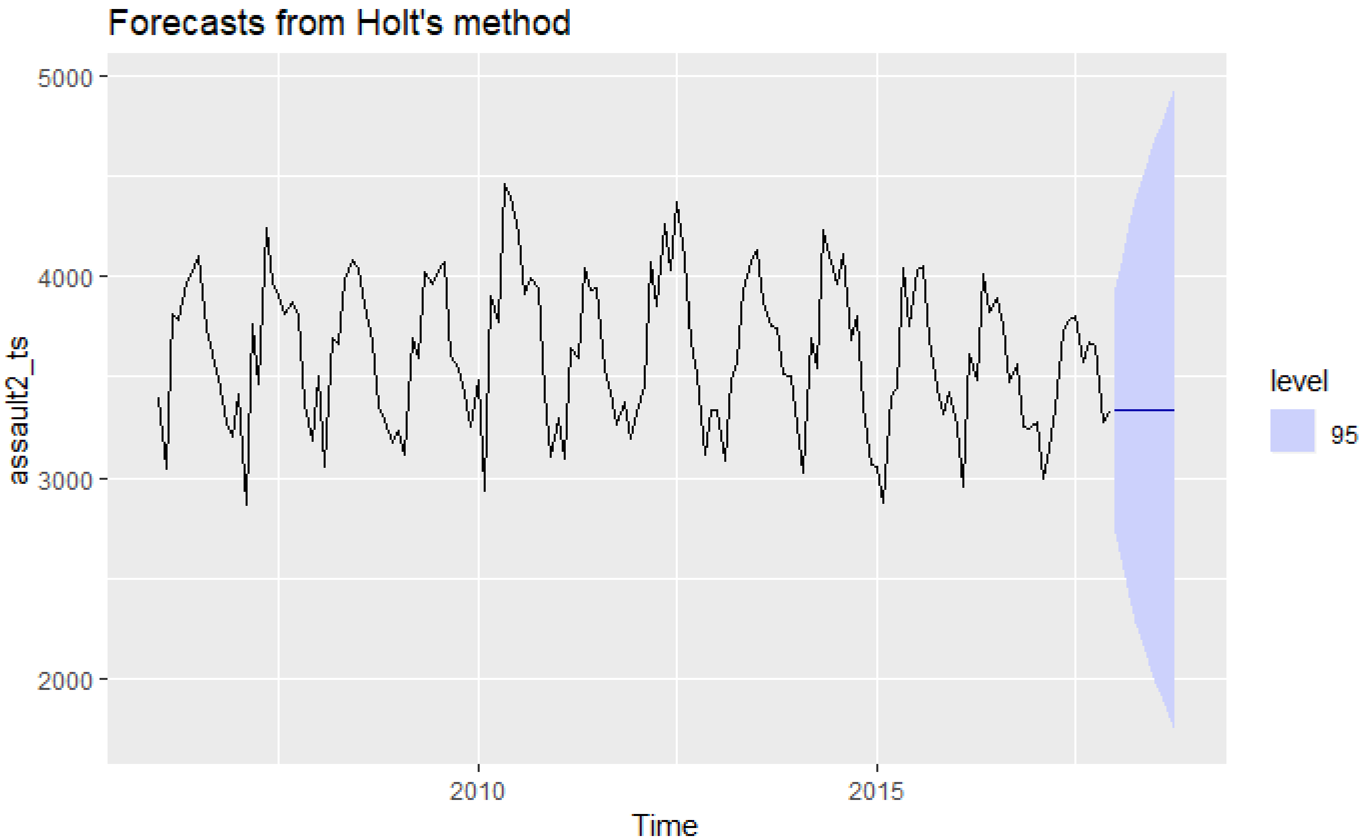
```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = grafitti_ts, level = 0.95)

  Smoothing parameters:
    alpha = 0.5085
    beta  = 1e-04

  Initial states:
    l = 1244.6585
    b = 8.7398

  sigma:  137.5054

     AIC     AICc      BIC
2139.612 2140.046 2154.461

Error measures:
                      ME     RMSE      MAE        MPE     MAPE      MASE       ACF1
Training set -0.03702489 135.5822 102.8011 -0.3935563 5.583505 0.7324706 0.03438069

Forecasts:
```

	Point Forecast <dbl>	**Lo 95** <dbl>	**Hi 95** <dbl>
Jan 2018	2509.211	2239.705	2778.717
Feb 2018	2517.950	2215.589	2820.311
Mar 2018	2526.690	2194.698	2858.681
Apr 2018	2535.429	2176.232	2894.626
May 2018	2544.168	2159.677	2928.659
Jun 2018	2552.907	2144.677	2961.138
Jul 2018	2561.647	2130.975	2992.318
Aug 2018	2570.386	2118.378	3022.394
Sep 2018	2579.125	2106.735	3051.515
Oct 2018	2587.864	2095.929	3079.800

1-10 of 10 rows

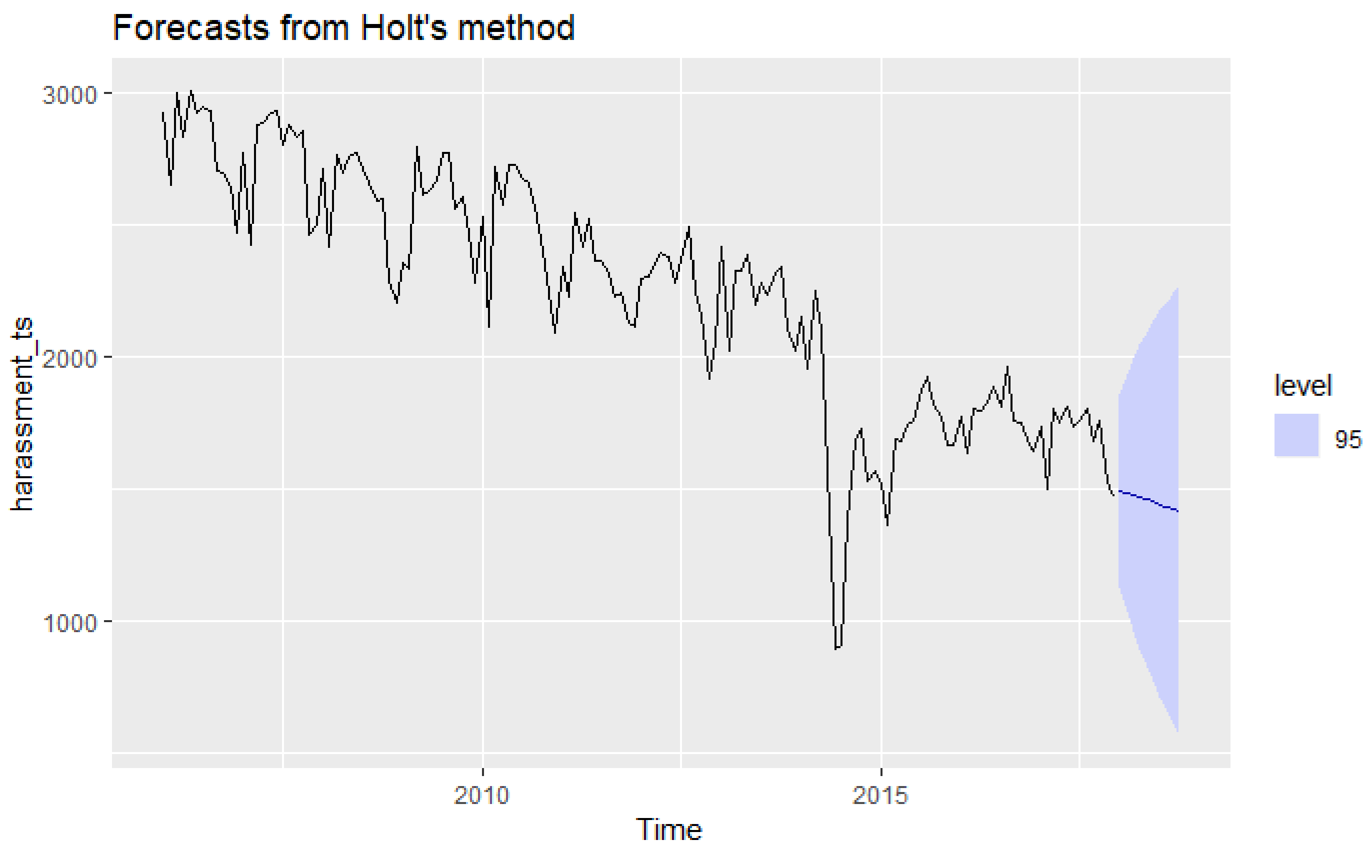
Hide

```
autoplot(assault2HW)
```



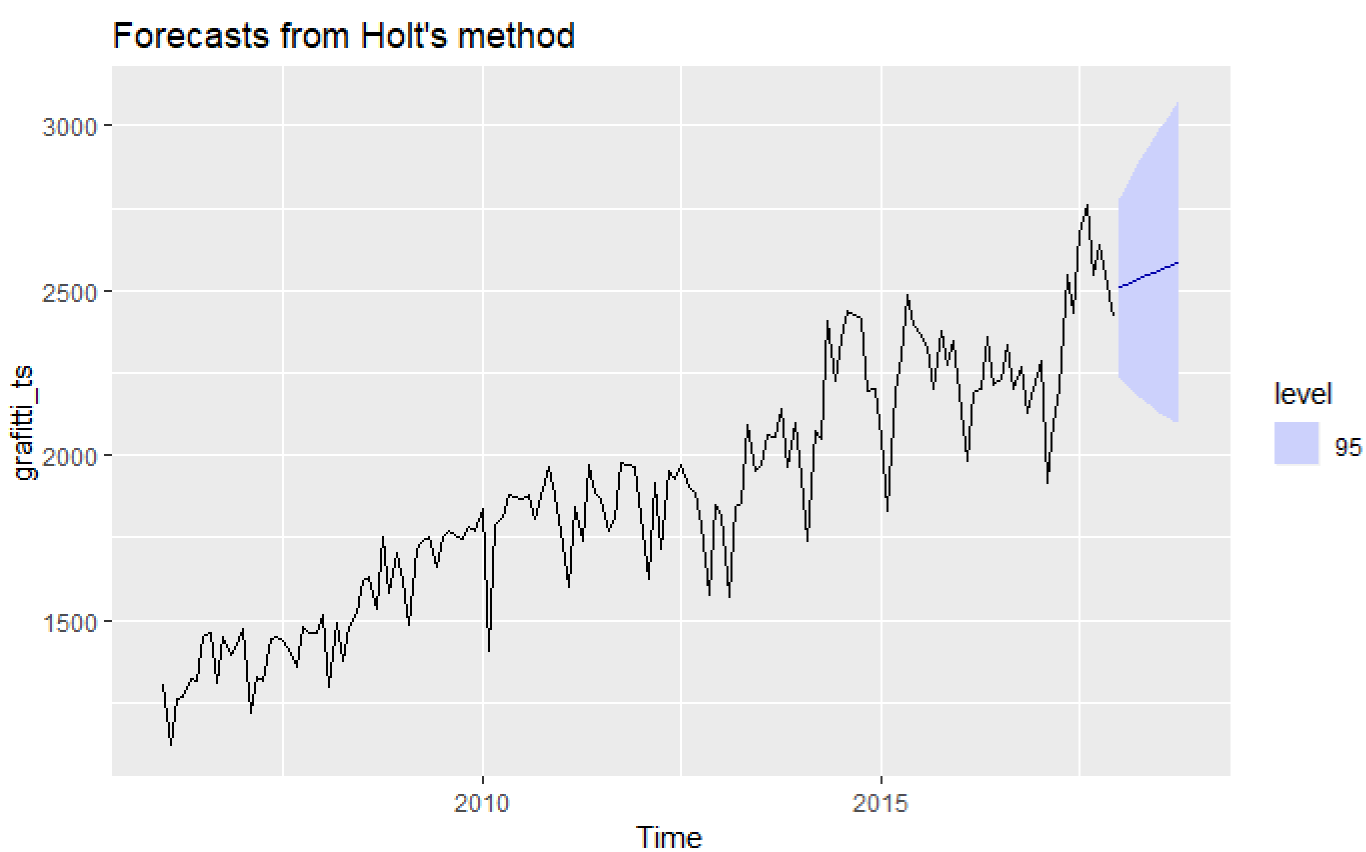
Hide

```
autoplot(harassmentHW)
```



Hide

```
autoplot(grafittiHW)
```



Hide

```
trainHarassment <-
complaintsByMonth[sample(1:nrow(complaintsByMonth), floor(.7 * nrow(complaintsByMonth))), ]
testHarassment <-
anti_join(complaintsByMonth, trainHarassment, by = 'MONTH')
# Training the linear model based on month and quarter (to account for seasonal trends)
harassment.lm <-
lm(
AGGRAVATED_HARASSMENT ~ MONTH + Q1 +
Q2 + Q3,data = trainHarassment
)
testHarassment$predictedHarassment <-
predict(harassment.lm, testHarassment, type = "response")
numRight <- nrow(filter(testHarassment, abs(AGGRAVATED_HARASSMENT-predictedHarassment)<sd(predic
tedHarassment)))
numRows <- nrow(testHarassment)
accuracy <- numRight / numRows
sprintf("Accuracy rate = %f", 100 * accuracy)
```

```
[1] "Accuracy rate = 95.454545"
```