Analysis of NYPD Shooting Incidents

2025-03-03

Assignment Description

In this assignment, we are tasked with producing a report analyzing the New York Police Department (NYPD) Shooting Incident data. As per the assignment requirement, we will generate two visualizations from the given data. Additionally, we will perform a predictive analysis on the data to check whether a shooting incident resulting in the victim's death would be counted as a murder.

Data Description

Source: https://data.cityofnewyork.us/api/views/833y-fsy8/

Description: List of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year. This is a breakdown of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year. This data is manually extracted every quarter and reviewed by the Office of Management Analysis and Planning before being posted on the NYPD website. Each record represents a shooting incident in NYC and includes information about the event, the location and time of occurrence. In addition, information related to suspect and victim demographics is also included. This data can be used by the public to explore the nature of shooting/criminal activity.

Load libraries

Let's import the necessary libraries to run this markdown file. Some libraries share the same function name, therefore using the 'conflicted' package to resolve when there is a conflict in the functions from different libraries.

```
# Turned off some flags to hide library load messages
# by using {r results='hide', message=FALSE, warning=FALSE}
# To install the packages uncomment the following two lines
# options(repos = c(CRAN = "https://cloud.r-project.org/"))
# install.packages(c("tidyverse", "lubridate", "ggplot2", "caret", "tinytex"))

library(conflicted)
library(tidyverse)
library(lubridate)
library(ggplot2)
library(caret)
```

Import Data

Let us import the data from the provided url. If the R version is >= 4.0, we do not need to use the parameter and value 'stringsAsFactors=FALSE'

```r
url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
shootings <- read.csv(url, stringsAsFactors = FALSE)
# Let's look at the head of the loaded dataframe
head(shootings, n=2) # View the top 2 rows only
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO LOC_OF_OCCUR_DESC PRECINCT
## 1    231974218 08/09/2021   01:06:00    BRONX                        40
## 2    177934247 04/07/2018   19:48:00 BROOKLYN                        79
##   JURISDICTION_CODE LOC_CLASSFCTN_DESC LOCATION_DESC STATISTICAL_MURDER_FLAG
## 1                 0                                                   false
## 2                 0                                                    true
##   PERP_AGE_GROUP PERP_SEX      PERP_RACE VIC_AGE_GROUP VIC_SEX VIC_RACE
## 1                                            18-24       M    BLACK
## 2          25-44        M WHITE HISPANIC         25-44       M    BLACK
##   X_COORD_CD Y_COORD_CD Latitude Longitude
## 1    1006343   234270.0 40.80967 -73.92019
## 2    1000083   189064.7 40.68561 -73.94291
##                                    Lon_Lat
## 1  POINT (-73.92019278899994 40.80967347200004)
## 2 POINT (-73.94291302299996 40.685609672000055)
```

Data Cleaning

If we have a glimpse of the data, we will see the data type for the column OCCUR_DATE is character. We convert the default OCCUR_DATE type to data type Date. Additionally, we see there are some NULL values. We drop the rows having NULL values.

```r
glimpse(shootings)
```

```
## Rows: 28,562
## Columns: 21
## $ INCIDENT_KEY            <int> 231974218, 177934247, 255028563, 25384540, 726~
## $ OCCUR_DATE              <chr> "08/09/2021", "04/07/2018", "12/02/2022", "11/~
## $ OCCUR_TIME              <chr> "01:06:00", "19:48:00", "22:57:00", "01:50:00"~
## $ BORO                    <chr> "BRONX", "BROOKLYN", "BRONX", "BROOKLYN", "BRO~
## $ LOC_OF_OCCUR_DESC       <chr> "", "", "OUTSIDE", "", "", "", "", "", "", "",~
## $ PRECINCT                <int> 40, 79, 47, 66, 46, 42, 71, 69, 75, 69, 40, 42~
## $ JURISDICTION_CODE       <int> 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 2, 0, 0, 2, 0~
## $ LOC_CLASSFCTN_DESC      <chr> "", "", "STREET", "", "", "", "", "", "", "", ~
## $ LOCATION_DESC           <chr> "", "", "GROCERY/BODEGA", "PVT HOUSE", "MULTI ~
## $ STATISTICAL_MURDER_FLAG <chr> "false", "true", "false", "true", "true", "fal~
## $ PERP_AGE_GROUP          <chr> "", "25-44", "(null)", "UNKNOWN", "25-44", "18~
## $ PERP_SEX                <chr> "", "M", "(null)", "U", "M", "M", "", "", "M",~
## $ PERP_RACE               <chr> "", "WHITE HISPANIC", "(null)", "UNKNOWN", "BL~
## $ VIC_AGE_GROUP           <chr> "18-24", "25-44", "25-44", "18-24", "<18", "18~
## $ VIC_SEX                 <chr> "M", "M", "M", "M", "F", "M", "M", "M", "M", "~
## $ VIC_RACE                <chr> "BLACK", "BLACK", "BLACK", "BLACK", "BLACK", "~
## $ X_COORD_CD              <dbl> 1006343.0, 1000082.9, 1020691.0, 985107.3, 100~
## $ Y_COORD_CD              <dbl> 234270.0, 189064.7, 257125.0, 173349.8, 247502~
## $ Latitude                <dbl> 40.80967, 40.68561, 40.87235, 40.64249, 40.845~
## $ Longitude               <dbl> -73.92019, -73.94291, -73.86823, -73.99691, -7~
## $ Lon_Lat                 <chr> "POINT (-73.92019278899994 40.80967347200004)"~
```

```r
# Statistical summary of each column in the dataframe
summary(shootings)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:28562       Length:28562       Length:28562
##  1st Qu.: 65439914   Class :character   Class :character   Class :character
##  Median : 92711254   Mode  :character   Mode  :character   Mode  :character
##  Mean   :127405824
##  3rd Qu.:203131993
##  Max.   :279758069
##
##  LOC_OF_OCCUR_DESC     PRECINCT     JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:28562       Min.   :  1.0   Min.   :0.0000    Length:28562
##  Class :character   1st Qu.: 44.0   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 67.0   Median :0.0000    Mode  :character
##                     Mean   : 65.5   Mean   :0.3219
##                     3rd Qu.: 81.0   3rd Qu.:0.0000
##                     Max.   :123.0   Max.   :2.0000
##                                     NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:28562       Length:28562            Length:28562
##  Class :character   Class :character        Class :character
##  Mode  :character   Mode  :character        Mode  :character
##
##
##
##
##    PERP_SEX           PERP_RACE         VIC_AGE_GROUP        VIC_SEX
##  Length:28562       Length:28562       Length:28562       Length:28562
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
##  Length:28562       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000068   1st Qu.:182912   1st Qu.:40.67
##  Mode  :character   Median :1007772   Median :194901   Median :40.70
##                     Mean   :1009424   Mean   :208380   Mean   :40.74
##                     3rd Qu.:1016807   3rd Qu.:239814   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :59
##    Longitude         Lon_Lat
##  Min.   :-74.25   Length:28562
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :59
```

```r
# Cast OCCUR_DATE as type Date
shootings$OCCUR_DATE <- as.Date(shootings$OCCUR_DATE, format="%m/%d/%Y")
# Drop rows with NULL values
shootings_clean <- shootings %>% drop_na()
head(shootings_clean, n=2) # View the top two rows only
```

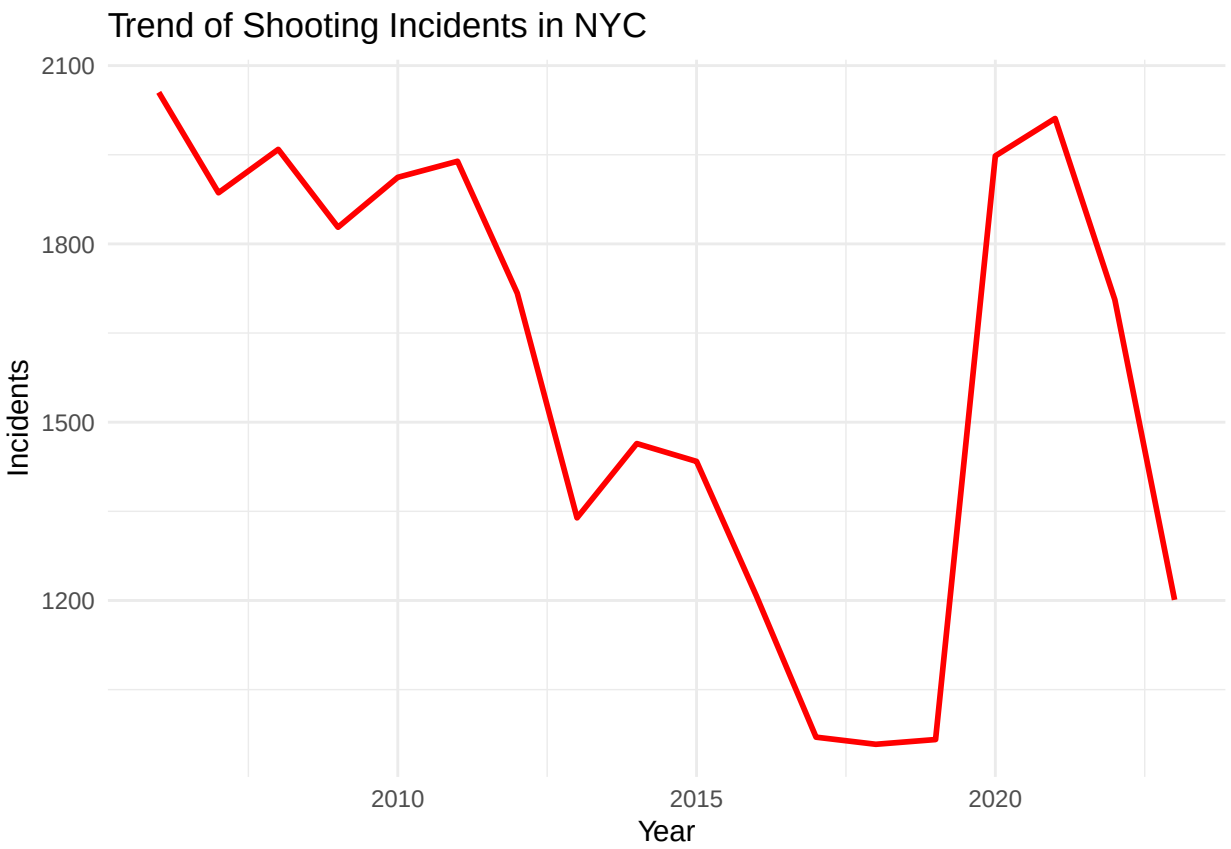
```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO LOC_OF_OCCUR_DESC PRECINCT
## 1    231974218 2021-08-09   01:06:00    BRONX                        40
## 2    177934247 2018-04-07   19:48:00 BROOKLYN                        79
##   JURISDICTION_CODE LOC_CLASSFCTN_DESC LOCATION_DESC STATISTICAL_MURDER_FLAG
## 1                 0                                                   false
## 2                 0                                                    true
##   PERP_AGE_GROUP PERP_SEX      PERP_RACE VIC_AGE_GROUP VIC_SEX VIC_RACE
## 1                                            18-24       M    BLACK
## 2          25-44        M WHITE HISPANIC         25-44       M    BLACK
##   X_COORD_CD Y_COORD_CD Latitude Longitude
## 1    1006343   234270.0 40.80967 -73.92019
## 2    1000083   189064.7 40.68561 -73.94291
##                                      Lon_Lat
## 1  POINT (-73.92019278899994 40.80967347200004)
## 2 POINT (-73.94291302299996 40.685609672000055)
```

Exploratory Data Analysis

1. Shooting Trends Over the Years

In this section, we are going to generate a plot highlighting the trend of shooting incidents in New York City over the years. To do so we first group the incidents by year and count how many incidents are there for each year. Then we generate a line graph by plotting the year on the X-axis and associated incident counts on the Y-axis.

```r
shootings_clean %>%
  mutate(Year = year(OCCUR_DATE)) %>%
  group_by(Year) %>%
  summarise(Incidents = n()) %>%
  ggplot(aes(x = Year, y = Incidents)) +
  geom_line(color = "red", linewidth = 1) +
  ggtitle("Trend of Shooting Incidents in NYC") +
  theme_minimal()
```

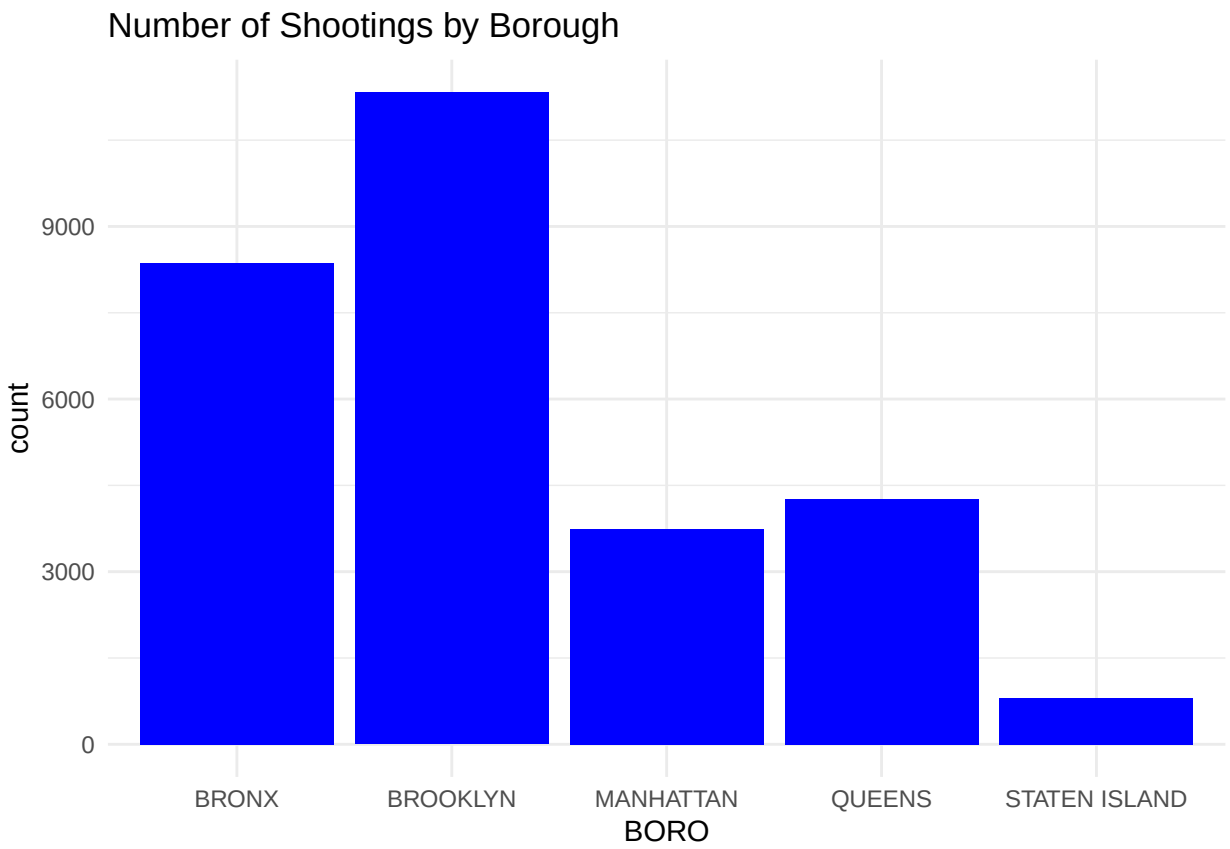


By visualizing the graph, we see the number of shooting incidents decreased gradually from the year 2006 to 2019. But there was a sharp increase in 2020 and 2021 which also decreased gradually after.

2. Borough-wise Distribution of Shootings

In this section, we are going to generate a plot displaying the borough-wise shooting incidents that happened over the years. From the data we see there are five boroughs in NYC. In this case, we are going to use a bar graph to represent the number of incidents for each NYC borough.

```
ggplot(shootings_clean, aes(x = BORO)) +
  geom_bar(fill = "blue") +
  ggtitle("Number of Shootings by Borough") +
  theme_minimal()
```



From the plot above we can see the most number of shooting incidents happened in BROOKLYN. On the other hand, the least number of shootings happened in STATEN ISLAND.

3. Predictive Modeling

Next we will perform a predictive analysis on the given NYC shooting dataset. For this analysis, let's assume we want to predict whether a shooting incident which resulted in a victim's death would be counted as murder. Therefore, the output variable for the predictive model in this case is the *STATISTICAL_MURDER_FLAG*. As far as the input variables are concerned, we can use a few independent variables or columns of the given dataset. But let's assume we want to predict whether a shooting incident is a murder by using the information about the age group (*VIC_AGE_GROUP*) of the victim and in which borough (*BOROUGH*) the shooting incident occurred.

```
# We saw our STATISTICAL_MURDER_FLAG values are "given as strings"true" and "false"
# and the data type is character or string.
# Let's convert "true" and "false" to numeric 1 and 0 respectively
shootings_clean$STATISTICAL_MURDER_FLAG <-
  ifelse(shootings_clean$STATISTICAL_MURDER_FLAG == "true", 1, 0)

# Check the distribution of the target variable
table(shootings_clean$STATISTICAL_MURDER_FLAG)
```

```
##
##     0     1
## 22979  5522
```

```r
# Ensure there are at least two unique values before partitioning
if (length(unique(shootings_clean$STATISTICAL_MURDER_FLAG)) > 1) {
  # We set a seed value to make sure the train and test set are reproducible
  set.seed(123)

  # We split the dataset into two parts: train and test with a ratio of 0.7 to 0.3.
  train_index <- createDataPartition(shootings_clean$STATISTICAL_MURDER_FLAG, p = 0.7, list = FALSE)
  train_data <- shootings_clean[train_index, ]
  test_data <- shootings_clean[-train_index, ]

  # We train a Logistic Regression Model on the train split
  model <- glm(STATISTICAL_MURDER_FLAG ~ BORO + VIC_AGE_GROUP,
               data = train_data,
               family = binomial)

  # Print a summary of the trained model
  summary(model)

  # We make Predictions on the test split
  predictions <- predict(model, newdata = test_data, type = "response")
  # If probability of a prediction is greater than 0.5, we consider the incident
  # as a murder, otherwise we consider it not a murder.
  predicted_classes <- ifelse(predictions > 0.5, 1, 0)

  # Evaluate Model Accuracy
  confusionMatrix(as.factor(predicted_classes), as.factor(test_data$STATISTICAL_MURDER_FLAG))
} else {
  print("Not enough variation in the target variable for partitioning.")
}
```

```
## Warning in confusionMatrix.default(as.factor(predicted_classes),
## as.factor(test_data$STATISTICAL_MURDER_FLAG)): Levels are not in the same order
## for reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6906 1644
##          1    0    0
##
##                Accuracy : 0.8077
##                  95% CI : (0.7992, 0.816)
##     No Information Rate : 0.8077
##     P-Value [Acc > NIR] : 0.5066
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.8077
```

```
##          Neg Pred Value :    NaN
##              Prevalence : 0.8077
##          Detection Rate : 0.8077
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

Model assesment

By looking at the confusion matrix output, we see that our Logistic Regression model trained on the train split had an accuracy of 0.8077 on the test split with a confidence interval of [0.7992, 0.816]. It suggests 80.77% of the cases were correctly identified as 'murder' from the total number of predictions the model made for 'murder'.

Bias Assessment

- **Data Collection Bias**: Some shootings may be underreported or misclassified.
- **Demographic Bias**: Differences in age, race, or gender classification could affect accuracy.
- **Model Bias**: Logistic regression assumes linear relationships, which may not fully capture complex patterns.

Conclusion

This analysis provides valuable insights into shooting incidents in NYC, revealing trends over time and geographical distributions. The logistic regression model attempts to predict fatal outcomes but is limited by the available data and inherent biases. Future work could enhance accuracy by incorporating additional features such as socioeconomic data, weather conditions, or historical crime rates. Additionally, exploring more sophisticated machine learning models may provide better predictive performance. Addressing biases in data collection and model assumptions is crucial for improving the reliability of such analyses.