NYPD Shooting Incident Data Report

Nikhil Patel

2023-10-10

The NYPD Shooting Incident data report provides a comprehensive list of shooting incidents in NYC dating back to 2006 until the end of the previous calendar year. The data is collected and reviewed quarterly by the Office of Management Analysis and Planning before being published on the NYPD website. Each record contains details about the incident, including location, time, and information about the suspects and victims. This dataset is available to the public and can be used to analyze patterns of shooting and criminal activity in the city. Additional information can be found in the attached data footnotes. This data is recent as of September 2, 2023.

The analysis will focus on demographics of both victims and perpetrators, providing summary statistics on incident counts by season. It will also present incident breakdowns by New York borough. Additionally, a logistic regression model will be employed, utilizing demographics like gender, age, and race to predict future murder rate patterns.

Step 0: Import Library

```
# install.packages("tidyverse")
library(tidyverse)
library(lubridate)
library(ggplot2)
```

Step 1: Load Data

First, the data will be retrieved from the United States government data repository. We will use the `read_csv` to load in the data frame and the `head()` function to view the first 5 rows.

```
## Get current Data for the NYPD Shooting Incident (Historic)
df <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")

## View first 5 rows
head(df)
```

```
## # A tibble: 6 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <chr>      <time>     <chr>    <chr>                <dbl>
## 1    228798151 05/27/2021 21:30      QUEENS   <NA>                   105
## 2    137471050 06/27/2014 17:40      BRONX    <NA>                    40
## 3    147998800 11/21/2015 03:56      QUEENS   <NA>                   108
## 4    146837977 10/09/2015 18:30      BRONX    <NA>                    44
## 5     58921844 02/19/2009 22:58      BRONX    <NA>                    47
```

```
## 6    219559682 10/21/2020 21:36      BROOKLYN <NA>                     81
## # i 15 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>
```

Let's review the number of incidents reported in the dataset and the variables available for analysis. It is also important to see what data types each variable is in case we need to adjust for the purposes of our analysis. To accomplish this task, we will use the `glimpse()` function. The `glimpse()` function in the `dplyr` package is used to provide a concise summary of the structure of a data frame or tibble. It offers a quick way to inspect the data by displaying a few rows and columns, along with information about the data types of each column.

```
glimpse(df)
```

```
## Rows: 27,312
## Columns: 21
## $ INCIDENT_KEY            <dbl> 228798151, 137471050, 147998800, 146837977, 58~
## $ OCCUR_DATE              <chr> "05/27/2021", "06/27/2014", "11/21/2015", "10/~
## $ OCCUR_TIME              <time> 21:30:00, 17:40:00, 03:56:00, 18:30:00, 22:58~
## $ BORO                    <chr> "QUEENS", "BRONX", "QUEENS", "BRONX", "BRONX",~
## $ LOC_OF_OCCUR_DESC       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ PRECINCT                <dbl> 105, 40, 108, 44, 47, 81, 114, 81, 105, 101, 2~
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 2~
## $ LOC_CLASSFCTN_DESC      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ LOCATION_DESC           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "MULTI DWE~
## $ STATISTICAL_MURDER_FLAG <lgl> FALSE, FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, ~
## $ PERP_AGE_GROUP          <chr> NA, NA, NA, NA, "25-44", NA, NA, NA, NA, "25-4~
## $ PERP_SEX                <chr> NA, NA, NA, NA, "M", NA, NA, NA, NA, "M", NA, ~
## $ PERP_RACE               <chr> NA, NA, NA, NA, "BLACK", NA, NA, NA, NA, "BLAC~
## $ VIC_AGE_GROUP           <chr> "18-24", "18-24", "25-44", "<18", "45-64", "25~
## $ VIC_SEX                 <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
## $ VIC_RACE                <chr> "BLACK", "BLACK", "WHITE", "WHITE HISPANIC", "~
## $ X_COORD_CD              <dbl> 1058925.0, 1005028.0, 1007667.9, 1006537.4, 10~
## $ Y_COORD_CD              <dbl> 180924.0, 234516.0, 209836.5, 244511.1, 262189~
## $ Latitude                <dbl> 40.66296, 40.81035, 40.74261, 40.83778, 40.886~
## $ Longitude               <dbl> -73.73084, -73.92494, -73.91549, -73.91946, -7~
## $ Lon_Lat                 <chr> "POINT (-73.73083868899994 40.662964620000025)~
```

Step 2: Tidy and Transform Data

This analysis is going to include a summary of the demographic variables of victims and perpetrators as well as a visual distribution of the dates of the incidents by seasonality and location of the crime by borough. Let's first isolate our variables of interest in our data set. This is going to include **INCIDENT_KEY**, **OCCUR_DATE**, **BORO**, **PERP_AGE_GROUP**, **PERP_SEX**, **PERP_RACE**, **VIC_AGE_GROUP**, **VIC_SEX**, **VIC_RACE**. Then, we will assess the missingness of our data.

```
tidy_df <- df %>%
  select(INCIDENT_KEY, OCCUR_DATE, BORO, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, V

lapply(tidy_df, function(x) sum(is.na(x)))
```

```
## $INCIDENT_KEY
## [1] 0
##
## $OCCUR_DATE
## [1] 0
##
## $BORO
## [1] 0
##
## $PERP_AGE_GROUP
## [1] 9344
##
## $PERP_SEX
## [1] 9310
##
## $PERP_RACE
## [1] 9310
##
## $VIC_AGE_GROUP
## [1] 0
##
## $VIC_SEX
## [1] 0
##
## $VIC_RACE
## [1] 0
##
## $STATISTICAL_MURDER_FLAG
## [1] 0
```

We can see that approximately 33% of the incidences have missing perpetrator demographic information. Assessing the missingness of data is important in the data analysis process. For instance, missing data can introduce errors and bias into analyses. There can be many reasons why data may be missing from a data set. In this scenario, perhaps the missing data is a function of an ongoing investigation where the perpetrator has not been caught. All missing data points in these columns will be changed to 'Unknown'.

```
tidy_df <- tidy_df %>%
  replace_na(list(PERP_AGE_GROUP = "Unknown", PERP_SEX = "Unknown", PERP_RACE = 'Unknown'))
```

Next, to visualize the seasonality distribution of the data, We will need to create a new column describing the season in which the incident occurred. This column will have 4 values (Winter, Spring, Summer, and Fall). Season will be determined by the following criteria:

- Winter: December, January, February
- Spring: March, April, May
- Summer: June, July, August
- Fall: September, October, November

```
tidy_df_with_seasons <- tidy_df %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE), # Convert 'Date' to Date object (month-day-year)
         Month = month(OCCUR_DATE), # Extract the month
         OCCUR_SEASON = case_when(
```

```
            month(Month) %in% c(12, 1, 2) ~ "Winter",
            month(Month) %in% c(3, 4, 5) ~ "Spring",
            month(Month) %in% c(6, 7, 8) ~ "Summer",
            month(Month) %in% c(9, 10, 11) ~ "Fall",
  )) %>%
  select(-Month) %>% # Remove the intermediate 'Month' column
  select(INCIDENT_KEY, OCCUR_DATE, OCCUR_SEASON, everything()) # Reorder columns

head(tidy_df_with_seasons)
```

```
## # A tibble: 6 x 11
##   INCIDENT_KEY OCCUR_DATE OCCUR_SEASON BORO    PERP_AGE_GROUP PERP_SEX PERP_RACE
##          <dbl> <date>     <chr>        <chr>   <chr>          <chr>    <chr>
## 1    228798151 2021-05-27 Spring       QUEENS  Unknown        Unknown  Unknown
## 2    137471050 2014-06-27 Summer       BRONX   Unknown        Unknown  Unknown
## 3    147998800 2015-11-21 Fall         QUEENS  Unknown        Unknown  Unknown
## 4    146837977 2015-10-09 Fall         BRONX   Unknown        Unknown  Unknown
## 5     58921844 2009-02-19 Winter       BRONX   25-44          M        BLACK
## 6    219559682 2020-10-21 Fall         BROOKL~ Unknown        Unknown  Unknown
## # i 4 more variables: VIC_AGE_GROUP <chr>, VIC_SEX <chr>, VIC_RACE <chr>,
## #   STATISTICAL_MURDER_FLAG <lgl>
```

Understanding the data type of each variable is critical to transforming data into the correct format for analysis. Below is a list of variables in which the data type must be changed.

- **INCIDENT_KEY**: double → string
- **OCCUR_SEASON**: string → factor
- **BORO**: string → factor
- **PERP_AGE_GROUP**: string → factor
- **PERP_SEX**: string → factor
- **PERP_RACE**: string → factor
- **VIC_AGE_GROUP**: string → factor
- **VIC_SEX**: string → factor
- **VIC_RACE**: string → factor

```
## Reclassify variables
tidy_df_with_seasons <- tidy_df_with_seasons %>%
  mutate(across(
    c(PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VIC_RACE),
    ~ case_when(
      . == "UNKNOWN" ~ "Unknown",
      . == "U" ~ "Unknown",
      . == "UNKNOWN" ~ "Unknown",
      . == "UNKNOWN" ~ "Unknown",
      . == "U" ~ "Unknown",
      . == "UNKNOWN" ~ "Unknown",
      TRUE ~ .
    )
  )) %>%
  mutate(
    INCIDENT_KEY = as.character(INCIDENT_KEY),
```

```
    OCCUR_SEASON = as.factor(OCCUR_SEASON),
    BORO = as.factor(BORO),
    PERP_AGE_GROUP = as.factor(PERP_AGE_GROUP),
    PERP_SEX = as.factor(PERP_SEX),
    PERP_RACE = as.factor(PERP_RACE),
    VIC_AGE_GROUP = as.factor(VIC_AGE_GROUP),
    VIC_SEX = as.factor(VIC_SEX),
    VIC_RACE = as.factor(VIC_RACE)
  ) %>%
  subset(VIC_AGE_GROUP != "1022" & PERP_AGE_GROUP != "1020" & PERP_AGE_GROUP != "224" & PERP_AGE_GROUP

## Summary statistics
summary(tidy_df_with_seasons)
```

```
##  INCIDENT_KEY         OCCUR_DATE         OCCUR_SEASON             BORO
##  Length:27308       Min.   :2006-01-01   Fall  :6795   BRONX        : 7935
##  Class :character   1st Qu.:2009-07-18   Spring:6239   BROOKLYN     :10932
##  Mode  :character   Median :2013-04-29   Summer:9222   MANHATTAN    : 3571
##                     Mean   :2014-01-06   Winter:5052   QUEENS       : 4094
##                     3rd Qu.:2018-10-15                 STATEN ISLAND:  776
##                     Max.   :2022-12-31
##
##  PERP_AGE_GROUP     PERP_SEX              PERP_RACE     VIC_AGE_GROUP
##  Unknown:12492   (null) :  640   BLACK          :11430   <18    : 2839
##  18-24  : 6221   F      :  424   Unknown        :11146   1022   :    0
##  25-44  : 5687   M      :15435   WHITE HISPANIC: 2339   18-24  :10085
##  <18    : 1591   Unknown:10809   BLACK HISPANIC: 1314   25-44  :12279
##  (null) :  640                   (null)         :  640   45-64  : 1863
##  45-64  :  617                   WHITE          :  283   65+    :  181
##  (Other):   60                   (Other)        :  156   Unknown:   61
##     VIC_SEX                                VIC_RACE     STATISTICAL_MURDER_FLAG
##  F      : 2615   AMERICAN INDIAN/ALASKAN NATIVE:   10   Mode :logical
##  M      :24682   ASIAN / PACIFIC ISLANDER      :  404   FALSE:22042
##  Unknown:   11   BLACK                         :19437   TRUE :5266
##                  BLACK HISPANIC                : 2646
##                  Unknown                       :   66
##                  WHITE                         :  698
##                  WHITE HISPANIC                : 4047
```

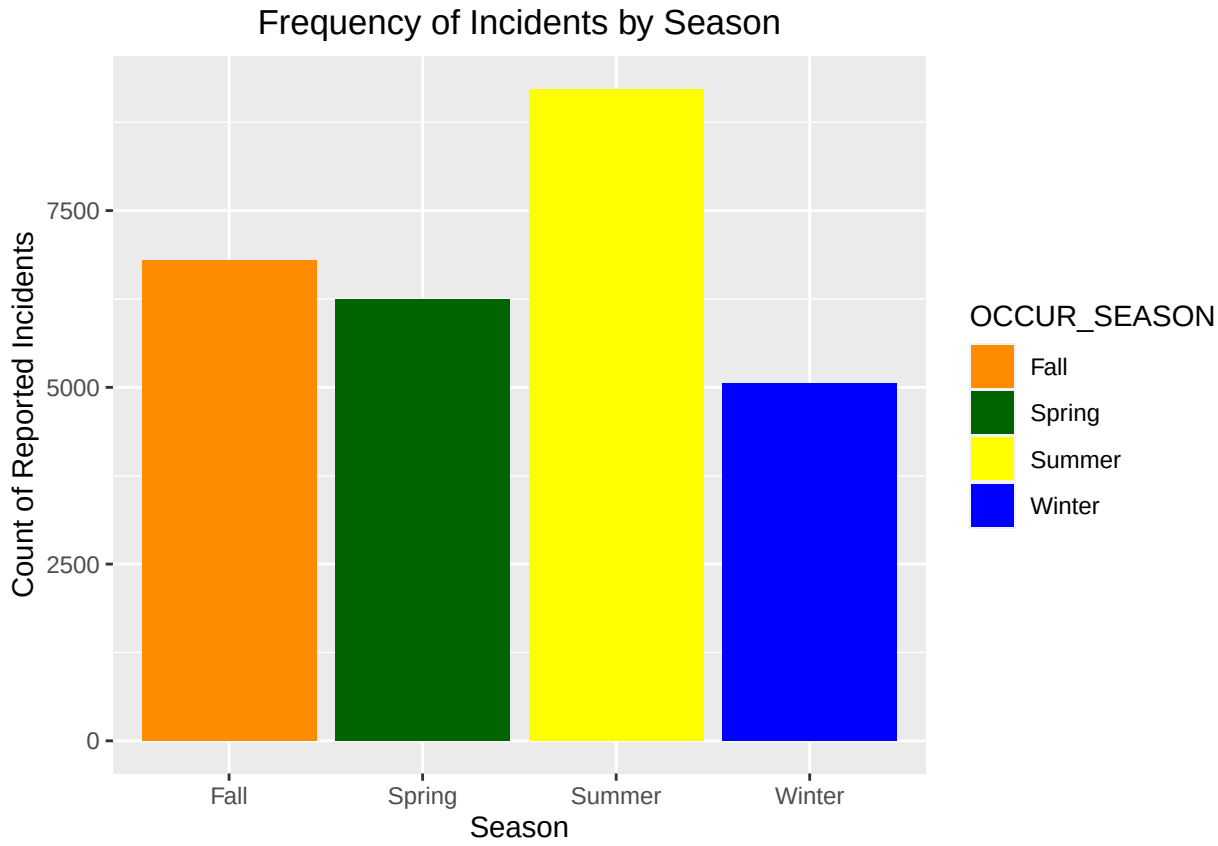
Step 3: Add Visualization and Analysis

1) Research Question: What is the distribution of incidents across each season?

We can see that most of the reported incidents occur during the summer months while the least number of incidents are occurring in the winter months.

```
ggplot(tidy_df_with_seasons, aes(x = OCCUR_SEASON, fill = OCCUR_SEASON)) +
  geom_bar() +
  scale_fill_manual(values = c(
  "Spring" = "dark green",
  "Summer" = "yellow",
  "Fall" = "dark orange",
```

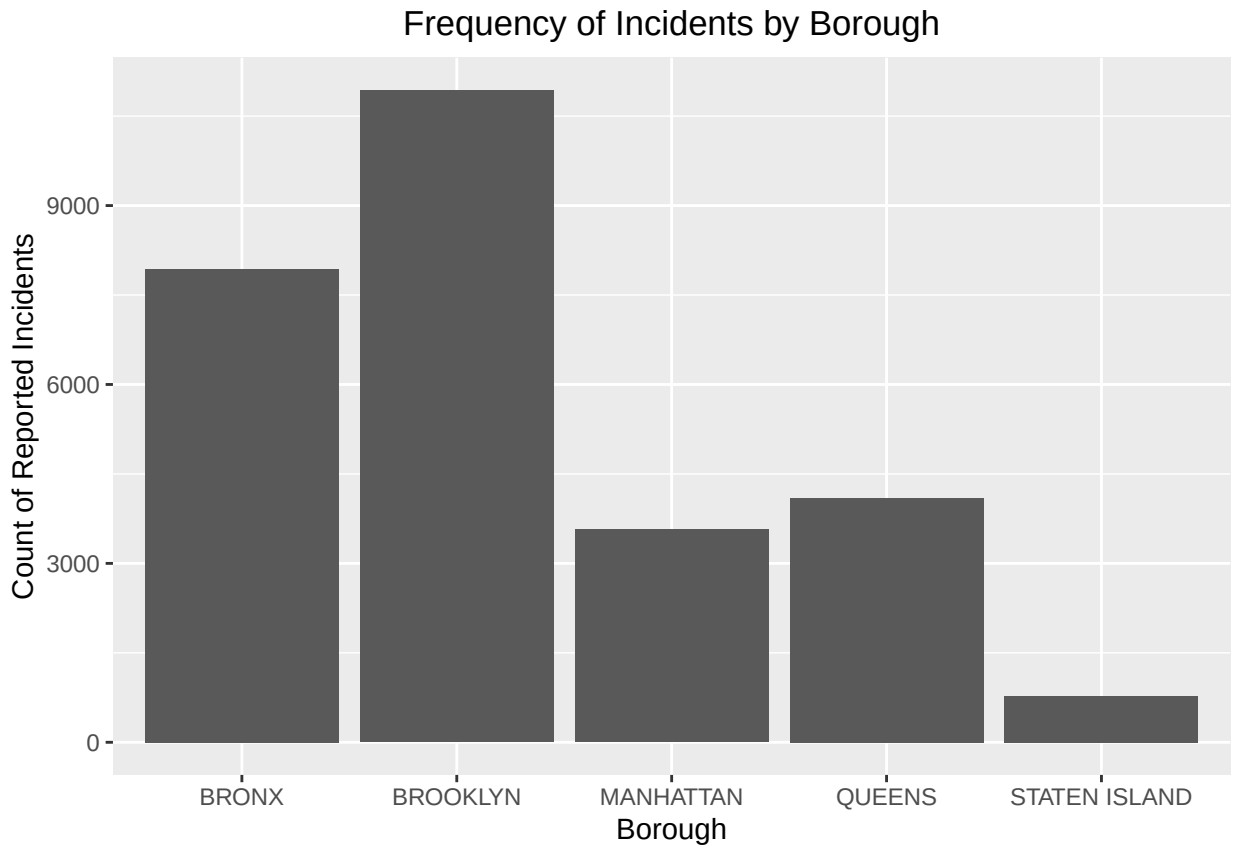
```
  "Winter" = "blue")) +
  labs(title = "Frequency of Incidents by Season", x = "Season", y = "Count of Reported Incidents") +
  theme(plot.title = element_text(hjust = 0.5))
```



2) What is the distribution of incidents across boroughs of New York City?

Brooklyn has the most reported incidents followed by the Bronx, Queens, Manhattan, and Staten Island.

```
ggplot(tidy_df_with_seasons, aes(x = BORO)) +
  geom_bar() +
  labs(title = "Frequency of Incidents by Borough", x = "Borough", y = "Count of Reported Incidents") +
  theme(plot.title = element_text(hjust = 0.5))
```



3) Are there any demographic variables for victims that are predictors of an incident being a murder?

For this analysis, we will utilize a logistic regression model. A logistic regression model is good for predicting the likelihood of a binary outcome. It's like a tool that helps you answer yes-or-no questions or make decisions based on specific factors or characteristics.

The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable. From the model, we see that `VIC_AGE_GROUP18-24`, `VIC_AGE_GROUP25-44`, `VIC_AGE_GROUP45-64`, `VIC_AGE_GROUP65+`, and `VIC_AGE_GROUPUnknown` were statistically significant. This means, for example, a victim in the age group of 65+, when compared to an individual under 18 years of age, changes the log odds of being murdered by 1.02.

```
glm.fit <- glm(STATISTICAL_MURDER_FLAG ~ VIC_AGE_GROUP + VIC_SEX + VIC_RACE, family = binomial, data = t

summary(glm.fit)
```

```
##
## Call:
## glm(formula = STATISTICAL_MURDER_FLAG ~ VIC_AGE_GROUP + VIC_SEX +
##     VIC_RACE, family = binomial, data = tidy_df_with_seasons)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -12.86438  102.16017  -0.126  0.89979
## VIC_AGE_GROUP18-24             0.28569    0.06197   4.610 4.02e-06 ***
```

```
## VIC_AGE_GROUP25-44                 0.61280    0.06005  10.204  < 2e-16 ***
## VIC_AGE_GROUP45-64                 0.75940    0.07781   9.760  < 2e-16 ***
## VIC_AGE_GROUP65+                   1.01924    0.17146   5.944 2.77e-09 ***
## VIC_AGE_GROUPUnknown               0.87540    0.31661   2.765  0.00569 **
## VIC_SEXM                          -0.04756    0.05206  -0.914  0.36091
## VIC_SEXUnknown                    -0.58932    1.08280  -0.544  0.58626
## VIC_RACEASIAN / PACIFIC ISLANDER  11.28111  102.16022   0.110  0.91207
## VIC_RACEBLACK                     11.00318  102.16015   0.108  0.91423
## VIC_RACEBLACK HISPANIC            10.82202  102.16017   0.106  0.91564
## VIC_RACEUnknown                   10.25877  102.16101   0.100  0.92001
## VIC_RACEWHITE                     11.34232  102.16019   0.111  0.91160
## VIC_RACEWHITE HISPANIC            11.12495  102.16016   0.109  0.91328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26779  on 27307  degrees of freedom
## Residual deviance: 26502  on 27294  degrees of freedom
## AIC: 26530
##
## Number of Fisher Scoring iterations: 11
```

Step 4: Identify Bias

When assessing bias, it is important to first look at how the data was collected. Is data being collected from all regions of New York equally or are some areas more or less represented than others? Furthermore, when looking at Perpetrator data, we can see that there are some Unknown values. What is the cause of this and from where are these crimes being committed? When analyzing any data set, it is important to assess the methodology of data collection and determine any shortcomings in the process. Furthermore, it is important to make any insights using data-driven conclusions and eliminating any personal bias. Only then can fair and factual evidence come to light through data analytics.