# NYPD Shooting Incident Data Report

## Contents



## 1. Project Outcomes

- The goal of this report is to Import, tidy and analyze the NYPD Shooting Incident dataset obtained and make sure that the project is reproducible and contains some visualization and analysis.
- At least two visualizations and one model must be included.
- Also, to identify any bias possible in the data and in the analysis.

## 2. Loading Datasets

In this section, we shall see about loading the datasets and understanding the schema of the datasets.

### 2.1. Data Source

- The data has been downloaded from https://catalog.data.gov/dataset. The dataset *NYPD Shooting Incident Data (Historic)* has the historical data of the Shooting incident happened in New York. The data has been reported by the New york police department
- I have chosen the CSV data format for its simplicity CSV.
- The comma separated version can be downloaded from https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD

### 2.2. Loading the data

It's better to read directly from the URL so that the code is **reproducible**. Having a local copy of dataset might result in inconsistent data copies by different collaborators.

Code:

```
data_url <- 'https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD'
data <- read.csv(data_url)
```

### 2.3. Displaying sample data

Now let's take a look at a few rows from the dataset.

Code:

```
head(data)
```

Output:

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME          BORO PRECINCT JURISDICTION_CODE
## 1    201575314 08/23/2019   22:10:00        QUEENS      103                 0
## 2    205748546 11/27/2019   15:54:00         BRONX       40                 0
## 3    193118596 02/02/2019   19:40:00     MANHATTAN       23                 0
## 4    204192600 10/24/2019   00:52:00 STATEN ISLAND      121                 0
## 5    201483468 08/22/2019   18:03:00         BRONX       46                 0
## 6    198255460 06/07/2019   17:50:00      BROOKLYN       73                 0
##   LOCATION_DESC STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX      PERP_RACE
## 1                                 false
## 2                                 false            <18        M          BLACK
## 3                                 false          18-24        M WHITE HISPANIC
## 4     PVT HOUSE                    true          25-44        M          BLACK
## 5                                 false          25-44        M BLACK HISPANIC
## 6                                 false          45-64        M WHITE HISPANIC
##   VIC_AGE_GROUP VIC_SEX       VIC_RACE X_COORD_CD Y_COORD_CD Latitude Longitude
## 1         25-44       M          BLACK    1037451     193561 40.69781 -73.80814
## 2         25-44       F          BLACK    1006789     237559 40.81870 -73.91857
## 3         18-24       M BLACK HISPANIC     999347     227795 40.79192 -73.94548
## 4         25-44       F          BLACK     938149     171781 40.63806 -74.16611
## 5         18-24       M          BLACK    1008224     250621 40.85455 -73.91334
## 6         25-44       M          BLACK    1009650     186966 40.67983 -73.90843
##                                           Lon_Lat
## 1 POINT (-73.80814071699996 40.697805308000056)
## 2  POINT (-73.91857061799993 40.81869973000005)
```

```
## 3 POINT (-73.94547965999999 40.791916091000076)
## 4  POINT (-74.16610830199996 40.63806398200006)
## 5  POINT (-73.91333944399999 40.85454734900003)
## 6  POINT (-73.90842523899994 40.67982701600005)
```

### 2.4. Understanding the schema

Inorder to refer tge column names for analysis and visualization, let's try to understand the column name and types

*Code:*

```
str(data)
```

*Output:*

```
## 'data.frame':    23568 obs. of  19 variables:
##  $ INCIDENT_KEY           : int  201575314 205748546 193118596 204192600 201483468 198255460 1945705
##  $ OCCUR_DATE             : chr  "08/23/2019" "11/27/2019" "02/02/2019" "10/24/2019" ...
##  $ OCCUR_TIME             : chr  "22:10:00" "15:54:00" "19:40:00" "00:52:00" ...
##  $ BORO                   : chr  "QUEENS" "BRONX" "MANHATTAN" "STATEN ISLAND" ...
##  $ PRECINCT               : int  103 40 23 121 46 73 81 67 114 69 ...
##  $ JURISDICTION_CODE      : int  0 0 0 0 0 0 0 0 2 0 ...
##  $ LOCATION_DESC          : chr  "" "" "" "PVT HOUSE" ...
##  $ STATISTICAL_MURDER_FLAG: chr  "false" "false" "false" "true" ...
##  $ PERP_AGE_GROUP         : chr  "" "<18" "18-24" "25-44" ...
##  $ PERP_SEX               : chr  "" "M" "M" "M" ...
##  $ PERP_RACE              : chr  "" "BLACK" "WHITE HISPANIC" "BLACK" ...
##  $ VIC_AGE_GROUP          : chr  "25-44" "25-44" "18-24" "25-44" ...
##  $ VIC_SEX                : chr  "M" "F" "M" "F" ...
##  $ VIC_RACE               : chr  "BLACK" "BLACK" "BLACK HISPANIC" "BLACK" ...
##  $ X_COORD_CD             : chr  "1037451" "1006789" "999347" "938149" ...
##  $ Y_COORD_CD             : chr  "193561" "237559" "227795" "171781" ...
##  $ Latitude               : num  40.7 40.8 40.8 40.6 40.9 ...
##  $ Longitude              : num  -73.8 -73.9 -73.9 -74.2 -73.9 ...
##  $ Lon_Lat                : chr  "POINT (-73.80814071699996 40.697805308000056)" "POINT (-73.918570
```

## 3. Data Cleaning & Pre-processing

In this section, I've tried to cleanse the data as much as possible so that analysis and modeling will be smooth.

### 3.1. Identifying and replacing nulls

Lets convert all the blanks with "NA" meaning Not-Applicable.

*Code:*

```
#Looking at few blanks
print("Before preprocessing : ")
print(head(data[data==""]))
```

```r
#clean blanks with 'NA's
cleaned_data <- data
cleaned_data[data==""] <- NA

print("After preprocessing : ")
print(head(cleaned_data[data==""]))
```

*Output:*

```
## [1] "Before preprocessing : "

## [1] NA NA "" "" "" ""

## [1] "After preprocessing : "

## [1] NA NA NA NA NA NA
```

### 3.2. Looking at the NAs

Let's see how many NAs are present in each columns.

*Code:*

```r
colSums(is.na(cleaned_data))
```

*Output:*

```
##            INCIDENT_KEY              OCCUR_DATE              OCCUR_TIME
##                       0                       0                       0
##                    BORO                PRECINCT       JURISDICTION_CODE
##                       0                       0                       2
##           LOCATION_DESC STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP
##                   13581                       0                    8459
##                PERP_SEX               PERP_RACE           VIC_AGE_GROUP
##                    8425                    8425                       0
##                 VIC_SEX                VIC_RACE              X_COORD_CD
##                       0                       0                       0
##              Y_COORD_CD                Latitude               Longitude
##                       0                       0                       0
##                 Lon_Lat
##                       0
```

It appears that the columns LOCATION_DESC, PERP_RACE, PERP_SEX and PERP_AGE_GROUP have lots of null values.

## 4. Exploratory Data Analysis (EDA)

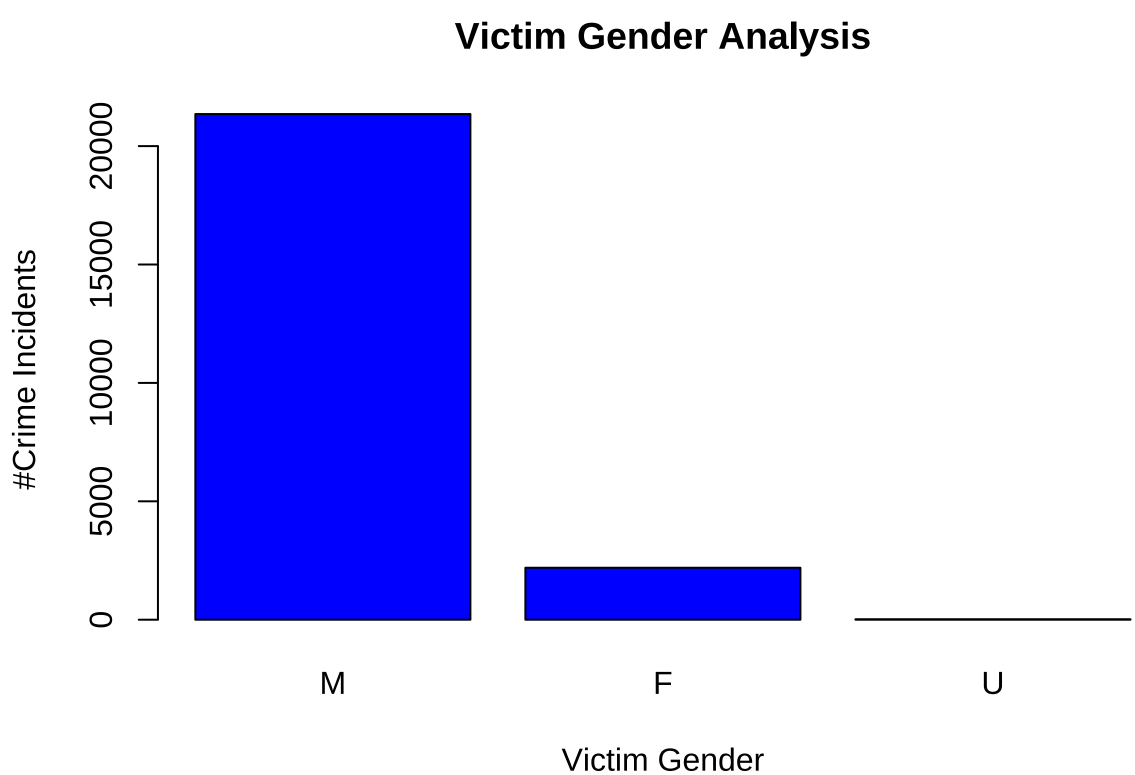
Let's explore by analysing a few dimensions from the dataset.

### 4.1. Which gender of victims are more susceptible?

- Gender is a primary but significant dimension that could help us understand the targets of the shooting crime incidents.
- For the gender analysis, we shall use the VIC_SEX dimension.

*Code*:

```
gender <- cleaned_data %>%
                count(VIC_SEX) %>%
                rename(VIC_SEX_CNT = n)

#Sorting by Victim gender count and finding % of gender
gender %>%
arrange(desc(VIC_SEX_CNT)) %>%
mutate(percentage = percent(VIC_SEX_CNT / sum(VIC_SEX_CNT))) -> gender


#Plotting the numbers
bar_plot <- barplot(gender$VIC_SEX_CNT,
                    names.arg=gender$VIC_SEX,
                    xlab="Victim Gender",
                    ylab="#Crime Incidents",
                    col="blue",
                    main="Victim Gender Analysis")
```



*Output*:

```
##   VIC_SEX VIC_SEX_CNT percentage
## 1       M       21353      90.6%
## 2       F        2195       9.3%
## 3       U          20       0.1%
```

```
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
```

*Analysis*:

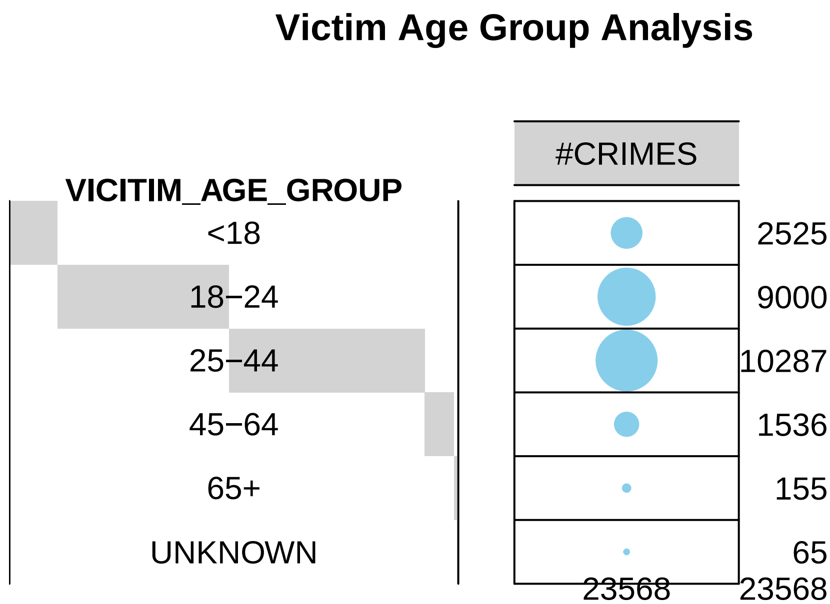
- From the numbers, *90.6%* of victims are identified as males.
- Males victims are highly susceptible to shooting crime according to the NYPD data.

### 4.2. Which age-group of victims are more susceptible?

- Age-group is another interesting dimension where the victims could potentially be targeted by their age
- For age group analysis, we shall use the VIC_AGE_GROUP dimension.
- I've used balloon plots for age-group analysis.

*Code*:

```
age_group <- cleaned_data %>%
                count(VIC_AGE_GROUP) %>%
                rename(VIC_AGE_GROUP_CNT = n)

#convert the data as a table

age_group_matrix <- as.table(as.matrix(age_group$VIC_AGE_GROUP_CNT))
row.names(age_group_matrix) <- age_group$VIC_AGE_GROUP
colnames(age_group_matrix) <- "#CRIMES"

#Plotting the numbers
age_plot <- balloonplot(t(age_group_matrix),
                        main ="Victim Age Group Analysis",
                        xlab="",
                        ylab="VICITIM_AGE_GROUP",
                        label = FALSE,
                        show.margins = TRUE)
```




*Output*:

```
##          #CRIMES
## <18         2525
## 18-24       9000
## 25-44      10287
## 45-64       1536
## 65+          155
## UNKNOWN       65

## NULL
```

*Analysis*:

- From the above balloon plot, we can see that the victims of age-group *25-44* are more prone to the NYPD shooting crime using the size of the balloons.
- Followed by that, the victims age group of *18-24* are also susceptible to the crime.
- Around 72.6% of the victims fall under the age-groups 18-24 and 25-44.

## 5. Identifying Biases

Let's find out if there any biases are present in the dataset.

### 5.1. Significance of Bias

- Bias can cause the results of a scientific study to be disproportionately weighted in favor of one result or group of subjects.
- This can cause misunderstandings of natural processes that may make conclusions drawn from the data unreliable.

### 5.2. Racial Bias

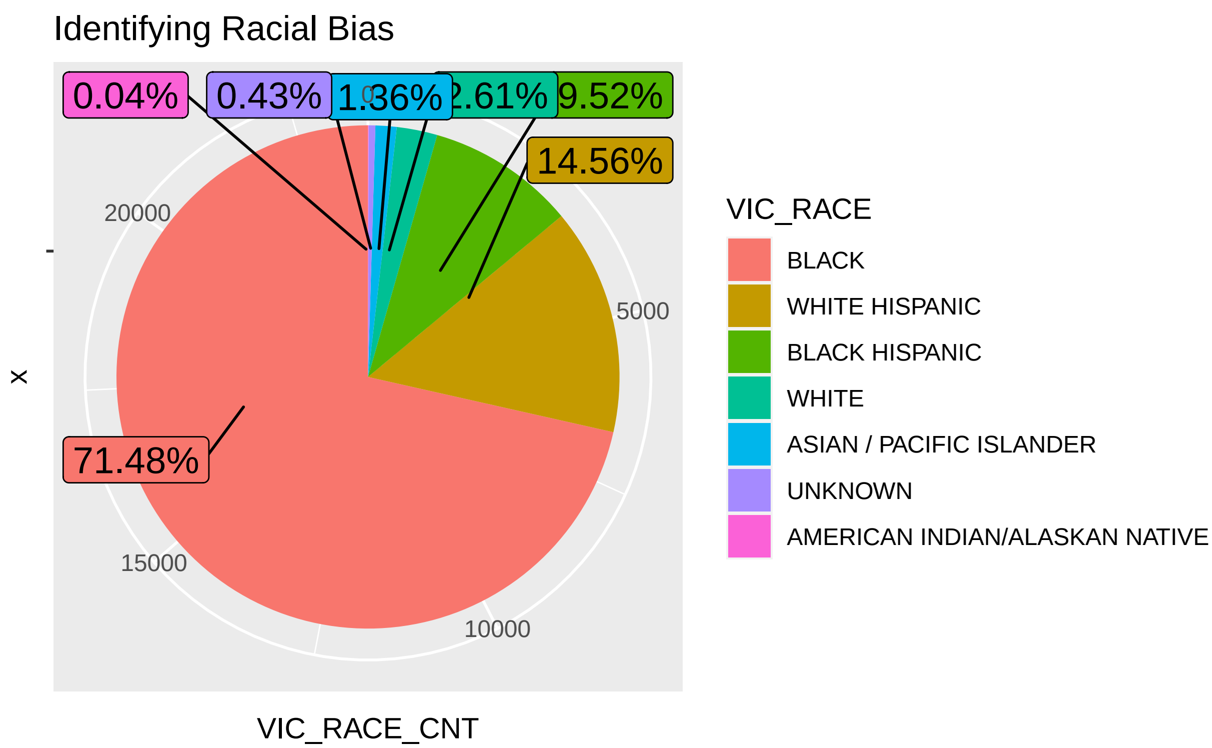
- When it comes to racial dimension, it is better to check for Racial bias so that the model won't be inclined to a particular race or community.
- We will use the column VIC_RACE and compute the % distribution across the races.

*Code*:

```
library(scales)

#Getting the count of each race
races <- cleaned_data %>%
                count(VIC_RACE) %>%
                rename(VIC_RACE_CNT = n)

#Sorting by Victim race count and finding % of races
races %>%
arrange(desc(VIC_RACE_CNT)) %>%
mutate(percentage = percent(VIC_RACE_CNT / sum(VIC_RACE_CNT))) -> races
print(races)

#Plotting the numbers

library(ggplot2)
library(ggrepel)
library(forcats)

pie <- ggplot(races, aes(x = "", y = VIC_RACE_CNT, fill = fct_inorder(VIC_RACE))) +
       geom_bar(width = 1, stat = "identity") +
       coord_polar("y", start = 0) +
       geom_label_repel(aes(label = percentage), size=5, show.legend = F, nudge_x = 1) +
       guides(fill = guide_legend(title = "VIC_RACE")) +
       ggtitle("Identifying Racial Bias")
```

*Output:*

```
##                         VIC_RACE VIC_RACE_CNT percentage
## 1                          BLACK        16846     71.48%
## 2                 WHITE HISPANIC         3432     14.56%
## 3                 BLACK HISPANIC         2244      9.52%
## 4                          WHITE          615      2.61%
## 5       ASIAN / PACIFIC ISLANDER          320      1.36%
## 6                        UNKNOWN          102      0.43%
## 7 AMERICAN INDIAN/ALASKAN NATIVE            9      0.04%
```




*Analysis*:

It is clearly seen that the race "BLACK" (71.5%) is over-represented in the dataset. Also, put together, Black and Black-Hispanic races contribute to >80% of the overall races.

## 6. Data Visualization

In this section, I have tried to visualize the data from different views.

### 6.1. Geo-plot using Longitude and Longitude

- Given that the latitude and longitude of the crime incident, we can visualize a geo-plot and try to infer whether any clusters are significantly seen.
- I'm interested in viewing the clusters/zones where more incidents have occurred. So I'm using "density2d" for the contour effect.

*Code:*

```
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

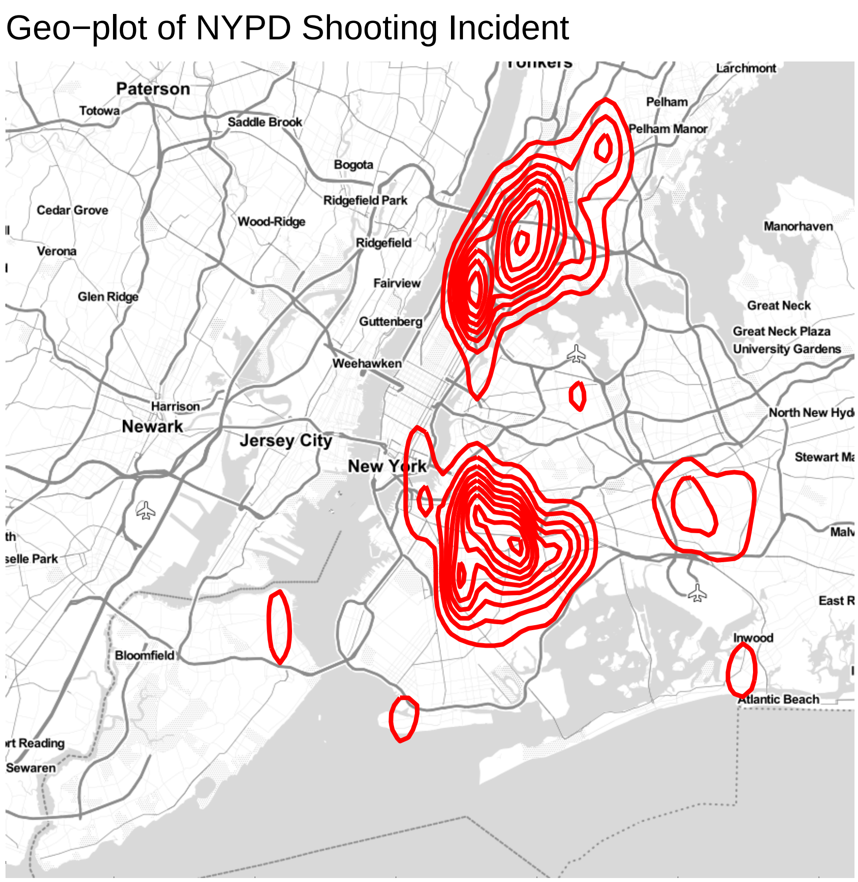
```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
geo_plot <- qmplot(Longitude,
                   Latitude,
                   data=cleaned_data,
                   colour = I('red'),
                   size = I(.8),
                   geom = "density2d",
                   main = "Geo-plot of NYPD Shooting Incident")
```

```
## Using zoom = 11...
```

```
## Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```

*Output:*




*Analysis:*

There are two primary crime zones - *Brooklyn* and *Bronx* from the above plot.

### 6.2. Time Series plot of the crime incidents

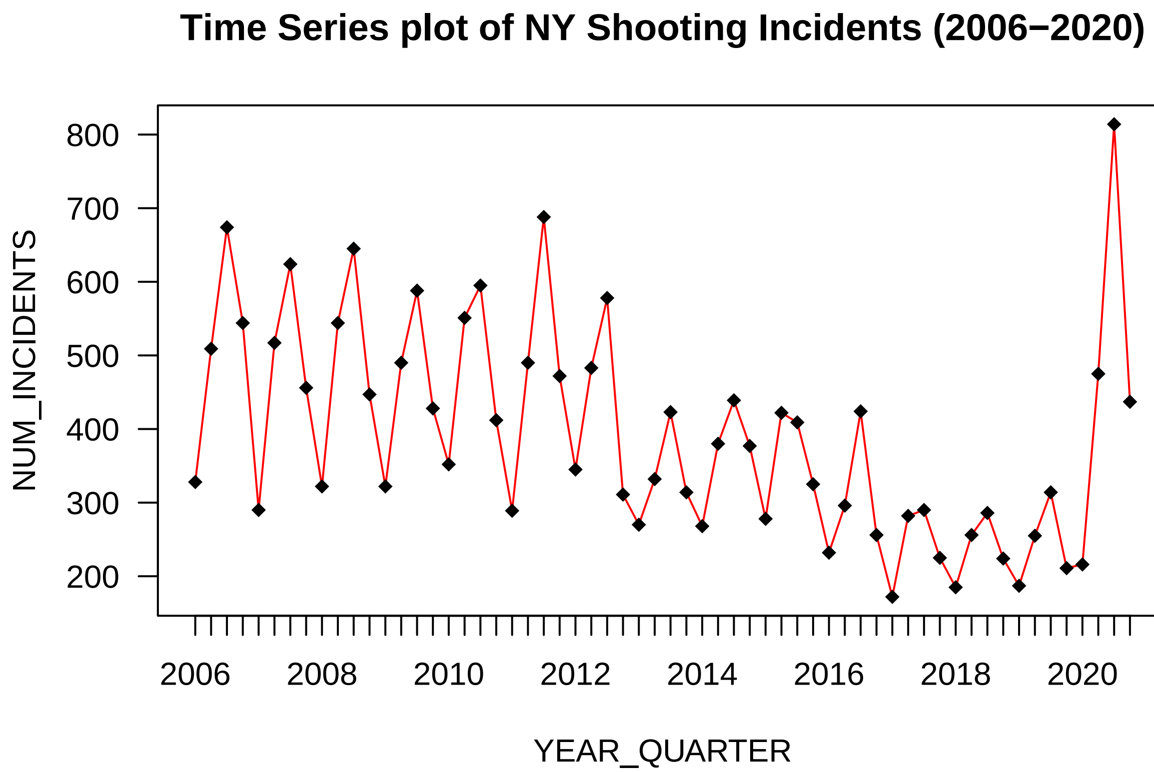
- Since we have the time dimension (OCCUR_DATE), let's attempt to visualize the time series plot of the crime incidents.
- I've considered quarterly plots since the date or month granularity is too much fragmented.

*Code:*

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
#Derive YEAR_QUARTER Column from OCCUR_DATE
cleaned_data$YEAR_QUARTER <- as.yearqtr(cleaned_data$OCCUR_DATE, format = "%m/%d/%Y")

#Aggregate the data by YEAR_QUARTER
time_series_data <- cleaned_data %>%
                       count(YEAR_QUARTER) %>%
                       rename(NUM_INCIDENTS = n) %>%
                       arrange(YEAR_QUARTER)
time_series_stats <- time_series_data %>%
                        arrange(desc(NUM_INCIDENTS)) %>%
                        mutate(percentage = percent(NUM_INCIDENTS / sum(NUM_INCIDENTS)))

#Plot the time series
time_series_plot <- plot(time_series_data$YEAR_QUARTER,
                         time_series_data$NUM_INCIDENTS,
                         main="Time Series plot of NY Shooting Incidents (2006-2020)",
                         xlab="YEAR_QUARTER",
                         ylab="NUM_INCIDENTS",
                         col="black",pch=18,las=1,
                         lines(time_series_data$YEAR_QUARTER,time_series_data$NUM_INCIDENTS, col="red"))
```




*Output:*

```
##   YEAR_QUARTER NUM_INCIDENTS percentage
## 1      2020 Q3           814    3.4538%
## 2      2011 Q3           688    2.9192%
## 3      2006 Q3           674    2.8598%
## 4      2008 Q3           645    2.7368%
## 5      2007 Q3           624    2.6477%
## 6      2010 Q3           595    2.5246%
```

```
## NULL
```

*Analysis:*

- It appears that maximum number of crime incidents happened during *Q3*.
- The years *2020* and *2011* seems to have more number of crimes.
- Around *3.5%* of entire crime happened in 2020-Q3

## 7. Data Modeling

In this section we will frame data science problems and try to answer them using regression models.

### 7.1. Defining the Problem

*Problem:*

- Let's try to forecast the number of crimes that might happen in the near future.
- Forecast is done at global level

### 7.2. Choosing a model

- From 6.2, it is clearly evident that the data has seasonality where Q3 reprensents the peak of crime incidents.
- We shall do a *Simple Exponential Smoothing (SES)* to predict the forecast for next 12 months (2021)

### 7.3. Time Series Forecasting using SES

- Lets validate how good is SES forecast using the historical time series.
- I'm using the dataframe time_series_data from section 6.2 for modeling and analysis.
- The time_series_data has 60 data points from 2006 Q1 to 2020 Q4.
- Let's use 75% of the data for training (45 data points)
- And the remaining 25% of the data for testing (15 data points)

*Code:*

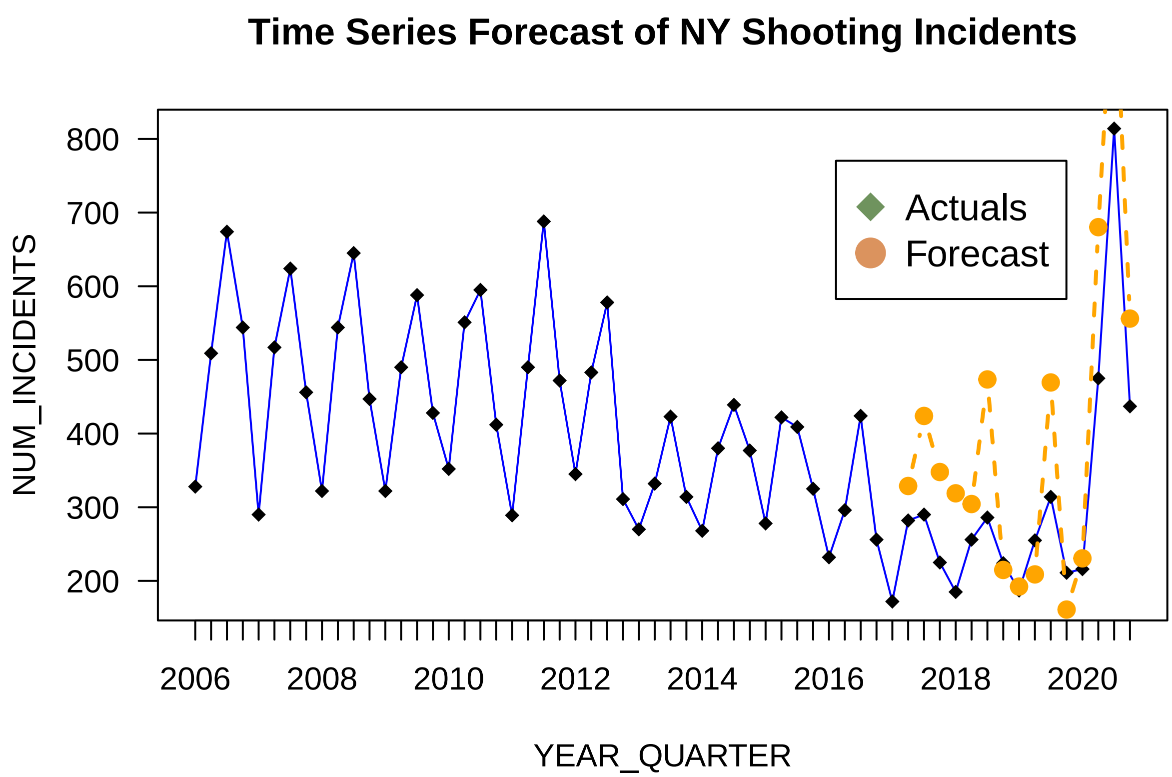
```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(dygraphs)
#Lets define the training & testing data
train <- time_series_data[1:45,]$NUM_INCIDENTS
test <- time_series_data[46:60,]

#Now, let's do a SES forecast for the next 15 quarters
forecast_results <- forecast(train,h=15)

#Assign actuals  from test set and compute averages
forecast_results <- as.data.frame(forecast_results)
forecast_results$actuals <- test$NUM_INCIDENTS
forecast_results$YEAR_QUARTER <- test$YEAR_QUARTER
forecast_results$avg_forecast <- (forecast_results$"Lo 80"+forecast_results$"Hi 80")/2


#PLot the forecast results
forecast_plot <- plot(time_series_data$YEAR_QUARTER,
                         time_series_data$NUM_INCIDENTS,
                         main="Time Series Forecast of NY Shooting Incidents",
                         xlab="YEAR_QUARTER",
                         ylab="NUM_INCIDENTS",
                         col="black",pch=18,las=1,
```

```
                         lines(time_series_data$YEAR_QUARTER,time_series_data$NUM_INCIDENTS, col="blue")

forecast_line <- lines(forecast_results$YEAR_QUARTER,
                       time_series_data$avg_forecast,
                       col="orange",
                       lwd=2,
                       pch=19 ,
                       type="b",
                       lty=2)

legend_obj <-legend("topright",
                    legend = c("Actuals", "Forecast"),
                    col = c(rgb(0.2,0.4,0.1,0.7),
                    rgb(0.8,0.4,0.1,0.7)),
                    pch = c(18,19),
                    pt.cex = 2,
                    cex = 1.2,
                    text.col = "black",
                    horiz = F ,
                    inset = c(0.1, 0.1))
```

*Output:*



### 7.4. Forecast Analysis

Let's compute the forecast accuracy and error.

*Code:*

```
library(Metrics)
```

```
##
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':
##
##     accuracy
```

```
x<-forecast_results$actuals
y<-forecast_results$avg_forecast

#Determining the Mean Absolute Error (MAE)
print("Mean Absolute Error (MAE) : ")
print(mae(x,y))
print("Root Mean Square Error (RMSE) : ")
print(rmse(x,y))

#Determining the accuracy score
print("Forecast Accuracy : ")
print(accuracy(x,y))
```

*Output:*

```
## [1] "Mean Absolute Error (MAE) : "

## [1] 78.68497

## [1] "Root Mean Square Error (RMSE) : "

## [1] 89.58393

## [1] "Forecast Accuracy : "

## [1] 81.32827
```

*Results:*

- We achieved a forecast model with *81.3%* confidence for predicting the crime incidents using the historical data
- The forecast accuracy can be improved by using other statistical models like Holt Winners, ARIMA, etc.,

## 8. Summary

- The report summarizes the various aspects of looking into NYPD Shooting crime datasets
- A basic SES model has been developed for forecasting the number of crime incidents.
- This report mainly focuses on the victims (ie., VIC_AGE_GROUP, VIC_SEX, VIC_RACE). There are other dimensions which might have potential insights towards understanding the crime.
- Modeling is limited to statistical approach and there is a scope for advanced ML models which might give interesting results.