NYPD Shooting Incident Data Report

29 May 2021

```r
# Load Libraries
library(tidyverse)
library(lubridate)
library(ggplot2)
```

Dataset

The NYPD Shooting Incident Data (Historic) dataset provides the list of every shooting incident that occurred in **New York City (NYC)** going back to 2006 through the end of 2020. The data was manually extracted every quarter and reviewed by the **Office of Management Analysis and Planning** before being posted on the **New York Police Department (NYPD)** website. Each record represents a shooting incident in NYC and includes information about the event, such as location, date/time of occurrence, and information related to perpetrator and victim demographics.

Exploration Analysis of the Dataset and Variables

After reading in the dataset from the source, the first step is to get an understanding of the dataset application domain, structure and variables, with the aim of identifying any relevant prior knowledge requirement and set the objectives of the analysis.

Dataset structure

Below a view of the dataset raw structure,

```r
# Load dataset from provided URL
url_in <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
raw_NSIDH <- read.csv(url_in)

# Get an insigth of the dataset structure
str(raw_NSIDH)
```

```
## 'data.frame':    23568 obs. of  19 variables:
##  $ INCIDENT_KEY           : int  201575314 205748546 193118596 204192600 201483468 198255460 194570529 2032117
77 193694863 199582060 ...
##  $ 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 (-73.91857061799993 40
.81869973000005)" "POINT (-73.94547965999999 40.791916091000076)" "POINT (-74.16610830199996 40.63806398200006)"
...
```

Objectives of the Analysis

From the dataset structure, we can observe that there are **19** variables and **23568** observations in the dataset, which include information about:

- location of the incident,
- victim and perpetrator demographics,
- date and time of the incidents.

For this analysis, we will focus our attention in getting an understanding of the nature of shooting activity in NYC, by:

- City and Boroughs,
- Demographic of victims and perpetrators,

Preprocessing

Dimensionality reduction

Features related to spatial data (**X_COORD_CD**, **Y_COORD_CD**, **Latitude**, **Longitude** and **Lon_Lat**), precinct (**PRECINCT**), location description (**LOCATION_DESC**) and jurisdiction code (**JURISDICTION_CODE**) will be removed from the dataset, as they will not add value to the goal of the analysis.

```r
# Remove selected features
pre_NSIDH <- raw_NSIDH %>% select(-c(PRECINCT, LOCATION_DESC, JURISDICTION_CODE, X_COORD_CD:Lon_Lat))
```

Data type conversion

After looking at the structure of the dataset, data fields such as: **OCCUR_DATE**, **OCCUR_TIME**, **PRECINCT**, **STATISTICAL_MURDER_FLAG** should to be converted to date, time, string and boolean data types respectively.

```r
# Data type convertion
pre_NSIDH <- pre_NSIDH %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE), OCCUR_TIME = hms(OCCUR_TIME), STATISTICAL_MURDER_
FLAG = as.logical(STATISTICAL_MURDER_FLAG))
```

Data cleaning

Exploration analysis

Let's take a look at the data pattern and/or ditribution of each variable to identify noises, missing values or any other abnormality,

```r
# Look at the summary of quantitative variables:
pre_NSIDH %>% select(OCCUR_DATE, OCCUR_TIME, STATISTICAL_MURDER_FLAG) %>% summary()

# Look at the qualitatives variables:

# BORO and PRECINCT
pre_NSIDH %>% count(BORO)

# PERP_AGE_GROUP
pre_NSIDH %>% count(PERP_AGE_GROUP)

# PERP_SEX
pre_NSIDH %>% count(PERP_SEX)

# PERP_RACE
pre_NSIDH %>% count(PERP_RACE)

# VIC_AGE_GROUP
pre_NSIDH %>% count(VIC_AGE_GROUP)

# VIC_SEX
pre_NSIDH %>% count(VIC_SEX)

# VIC_RACE
pre_NSIDH %>% count(VIC_RACE)
```

Handling missing values and data noises

From the the exploratory analysis of the variables **some missing values and noises were spotted** in variables: **PERP_AGE_GROUP**, **PERP_RACE** and **PERP_SEX**, to address those issues, we will carry out the following strategy:

- Replace **""** value by **"UNKNOWN"** for **PERP_AGE_GROUP** and **PERP_RACE** variables;
- Replace **""** value by **"U"** for **PERP_SEX**;
- Replace values **"1020"**, **"224"** and **"940"** by **"UNKNOWN"** for **PERP_AGE_GROUP** variable.

```r
# Replace "" value by "UNKNOWN" for PERP_AGE_GROUP and PERP_RACE, and by "U" for PERP_SEX
pre_NSIDH <- pre_NSIDH %>% mutate(PERP_AGE_GROUP = ifelse(PERP_AGE_GROUP == "", "UNKNOWN", PERP_AGE_GROUP),
                                  PERP_RACE = ifelse(PERP_RACE == "", "UNKNOWN", PERP_RACE),
                                  PERP_SEX = ifelse(PERP_SEX == "", "U", PERP_SEX))

# Replace values "1020", "224" and "940" by "UNKNOWN" for PERP_AGE_GROUP
pre_NSIDH <- pre_NSIDH %>% mutate(PERP_AGE_GROUP = ifelse(PERP_AGE_GROUP == "1020" | PERP_AGE_GROUP == "224" |
                                    PERP_AGE_GROUP == "940", "UNKNOWN", PERP_AGE_GROUP))
```

Data Transformation

Enriching the dataset

Before proceeding with the visualization of the data pattern, we will read in a new dataset that contains the population of each demographic groups under analysis, with the aim of measuring the impact of the shooting activity per population. The dataset was prepared from the tables of the NYC demographic profile, US Census 2010; posted in the *NYC Department of City Planning* website.

```r
# Importing NYC demographic profile from US Census 2010
# Data was extracted from url: https://www1.nyc.gov/site/planning/planning-level/nyc-population/census-2010.page
# and converted into a csv file and posted in a public Github repository

NYC_dem_profile_2010 <- read_csv("https://raw.githubusercontent.com/rojasael/DTSA-5301/main/NYPD_Shooting_Inciden
t_Report/NYC_Demographic_Profile_Census2010.csv")

# Creating a subset of borough incident and population
NYC_dem_boro <- NYC_dem_profile_2010 %>% filter(Category == "Borough") %>% rename(BORO = Demographic_Group) %>% s
elect(-c(Category))

loc_NSIDH <- pre_NSIDH %>% left_join(NYC_dem_boro) %>% select(c(INCIDENT_KEY, OCCUR_DATE, OCCUR_TIME, BORO, Popul
ation, STATISTICAL_MURDER_FLAG))

# Creating a subset of victim incident and population
NYC_dem_vic <- NYC_dem_profile_2010 %>% filter(Category == "Race") %>% rename(VIC_RACE = Demographic_Group) %>% s
elect(-c(Category))

rvic_NSIDH <- pre_NSIDH %>% left_join(NYC_dem_vic) %>% select(c(INCIDENT_KEY, OCCUR_DATE, OCCUR_TIME, VIC_RACE, P
opulation, STATISTICAL_MURDER_FLAG))
```

Summarization of number of incidents by demographic groups

Now we will manipulate the data to calculate the number of incidents by each demographic group,

```r
# Summarizing the number of incidents by Boro and adding cumulative number of cases
boro_NSIDH <- loc_NSIDH %>% count(OCCUR_DATE, BORO, Population) %>% rename(incidents = n)

boro_NSIDH <- boro_NSIDH %>% group_by(OCCUR_DATE, BORO, Population) %>% summarize(incidents = sum(incidents)) %>%
ungroup()

boro_NSIDH <- boro_NSIDH %>% group_by(BORO) %>% mutate(cum_incidents = cumsum(incidents)) %>% ungroup()

# Summarizing the number of incidents by victim race and adding cumulative number of cases

race_vic_NSIDH <- rvic_NSIDH %>% count(OCCUR_DATE, VIC_RACE, Population) %>% rename(incidents = n)

race_vic_NSIDH <- race_vic_NSIDH %>% group_by(OCCUR_DATE, VIC_RACE, Population) %>% summarize(incidents = sum(inc
idents)) %>% ungroup()

race_vic_NSIDH <- race_vic_NSIDH %>% group_by(VIC_RACE) %>% mutate(cum_incidents = cumsum(incidents)) %>% ungroup
()
```

Normalization of number of incidents by population

```r
# Add cumulative number of cases per 100K people
boro_NSIDH <- boro_NSIDH %>% mutate(new_inc_phtp = ((incidents * 100000)/Population), cum_inc_phtp = ((cum_incide
nts * 100000)/Population))

# Add cumulative number of cases per 100K people
race_vic_NSIDH <- race_vic_NSIDH %>% mutate(new_inc_phtp = ((incidents * 100000)/Population), cum_inc_phtp = ((cu
m_incidents * 100000)/Population))
```

Data Visualization and Analysis

In this section, we are going to explore, visualize and analyze different data patterns that will help us to get the understanding required to formulate our conclusion,

NYC Shooting Activity, 2006-2020

First, let's look at the trend of number of shooting incidents in NYC from 2006 to 2010,

```r
# Number of incidents in NYC per year
ni_NYC_NSIDH <- boro_NSIDH %>% group_by(year(OCCUR_DATE)) %>% summarize(new_inc_phtp = sum(new_inc_phtp)) %>% ung
roup() %>% rename(Year = `year(OCCUR_DATE)`)

# Linear regression for incidents from 2006 to 2019
NYC_noi_0619 <- ni_NYC_NSIDH %>% filter(Year >= 2006 & Year <= 2019)

LRM_NYC_noi <- lm(new_inc_phtp ~ Year, data = NYC_noi_0619)

# Rate of incident per 100K people from 2006 to 2019
ia_rate <- round(LRM_NYC_noi$coefficients["Year"],1)

# Add prediction model into NSIDH_nyc_year
NYC_noi_pred <- ni_NYC_NSIDH %>% mutate(pred = predict(LRM_NYC_noi, newdata = ni_NYC_NSIDH))

# Visualization
NYC_noi_pred %>%  ggplot(aes(x=Year)) + geom_line(aes(y = new_inc_phtp), colour = "blue") + geom_point(aes(y = ne
w_inc_phtp), colour = "blue") + geom_line(aes(y = pred), color = "red") + theme(legend.position = "bottom") +  la
bs(x = "year", y = "incidents per 100K people") + scale_x_continuous(breaks = ni_NYC_NSIDH$Year)
```

Figure 1.1 NYC Annual Shooting Incidents per 100K People

We used a liner regression model to calculate the interannual decline rate of incidents per 100K people from 2006 to 2019. Below the summary of the linear regression model,

```r
# Linear Regression Model
summary(LRM_NYC_noi)
```

```
##
## Call:
## lm(formula = new_inc_phtp ~ Year, data = NYC_noi_0619)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7775 -5.6089 -0.7132  3.9082 13.6625
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10366.5018   954.9973   10.86 1.47e-07 ***
## Year           -5.1066     0.4745  -10.76 1.61e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.157 on 12 degrees of freedom
## Multiple R-squared:  0.9061, Adjusted R-squared:  0.8983
## F-statistic: 115.8 on 1 and 12 DF,  p-value: 1.612e-07
```

NYC Shooting Activity by Borough, 2006-2020

Second, let's look at the trends of the NYC boroughs,

```r
# Number of incidents by boro per year
ni_boro_NSIDH <- boro_NSIDH %>% group_by(BORO, year(OCCUR_DATE)) %>% summarize(new_inc_phtp = sum(new_inc_phtp))
%>% ungroup() %>% rename(Year = `year(OCCUR_DATE)`)

ni_boro_NSIDH %>%  ggplot(aes(x = Year, y = new_inc_phtp, color = BORO)) + geom_line(aes(y = new_inc_phtp, color
= BORO)) + geom_point(aes(y = new_inc_phtp, color = BORO)) + theme(legend.position = "bottom") +  labs(x = "year"
, y = "incidents per 100K people") + scale_x_continuous(breaks = ni_NYC_NSIDH$Year)
```

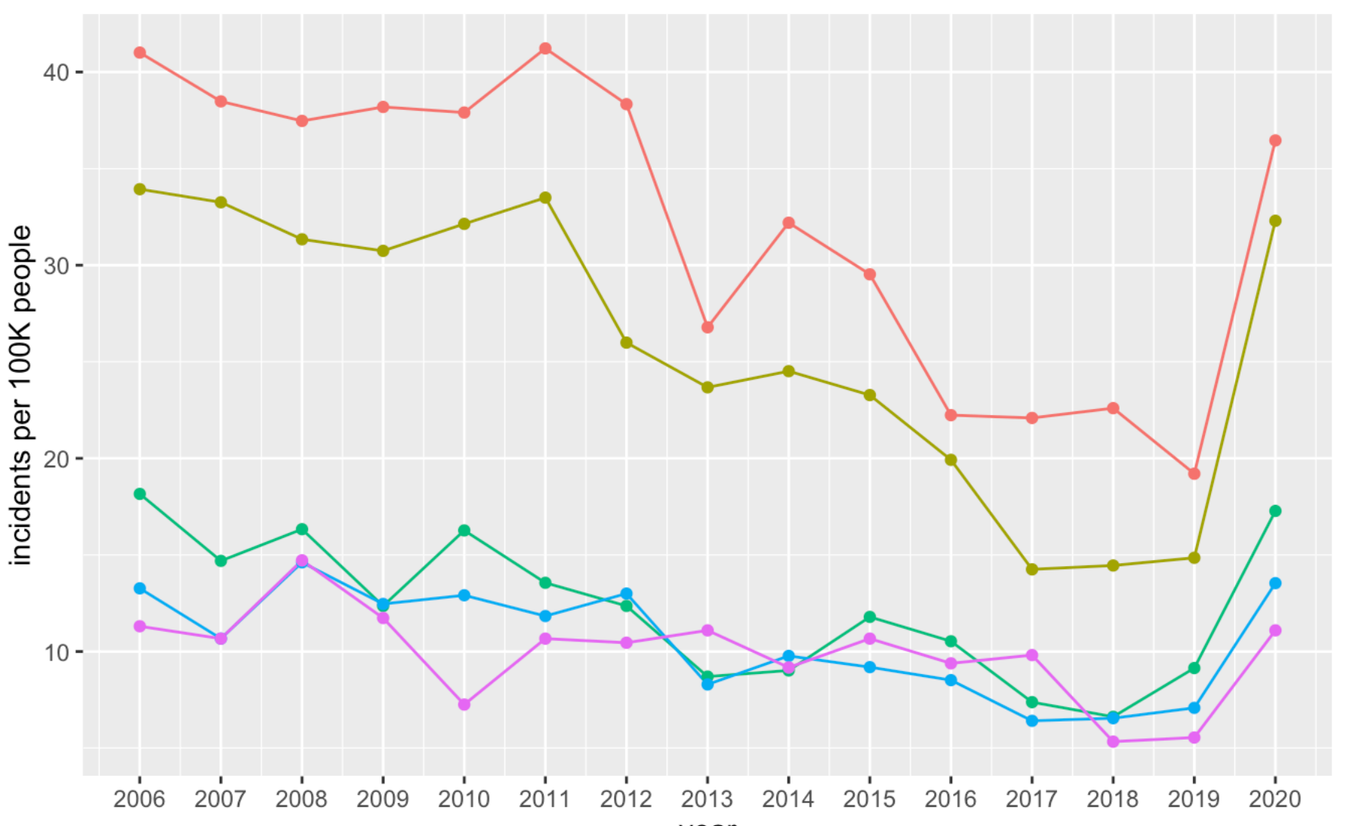

Figure 1.2 Annual Incidents per 100K People by Borough

```r
# Cumulative number of incidents by boro per year
boro_NSIDH %>% ggplot(aes(x = OCCUR_DATE, y = cum_inc_phtp, color = BORO)) + geom_line(aes(y = cum_inc_phtp, colo
r = BORO)) + theme(legend.position = "bottom") +  labs(x = "year", y = "incidents per 100K people") + scale_x_dat
e(date_breaks = "1 year", date_labels = "%Y")
```

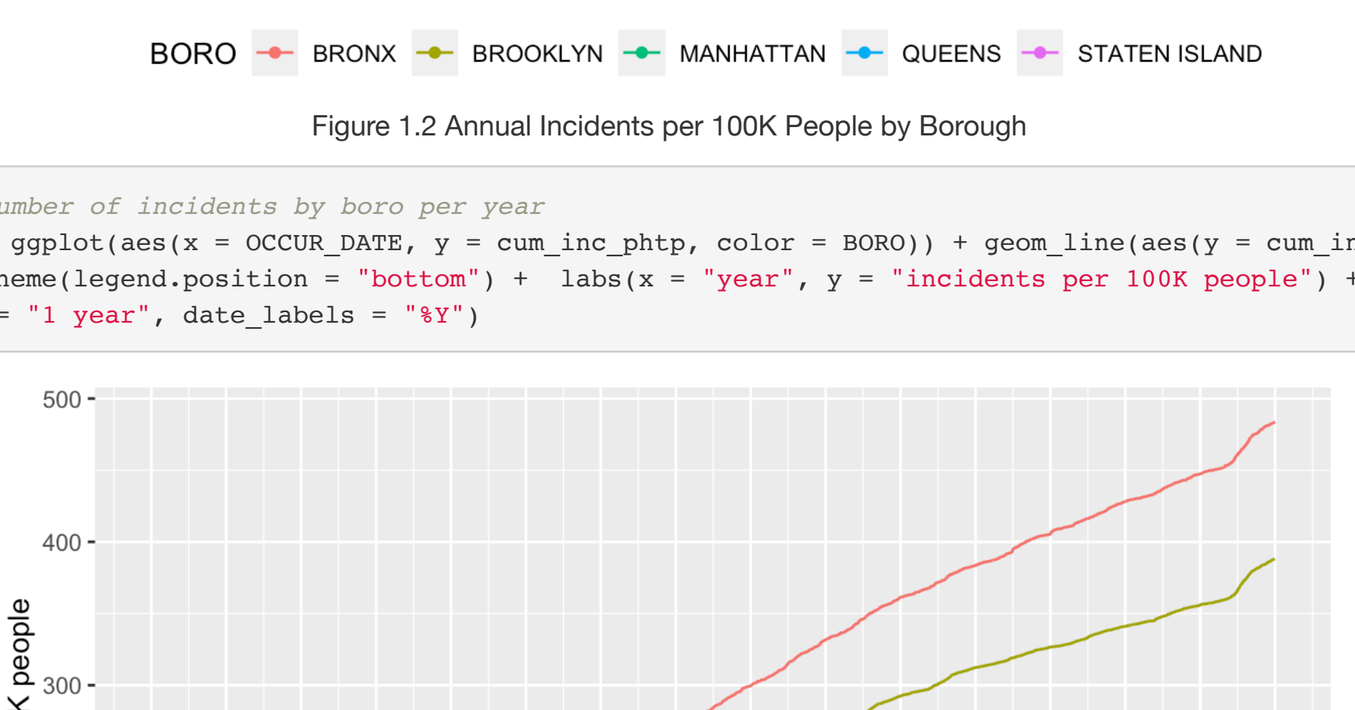

Figure 1.4 Figure 1.3 Cumulative Incidents per 100K People by Borough

NYC Shooting Activity by Victim Race Group, 2006-2020

Third, let's visualize the number of incidents by victim race group,

```r
#Number of incidents by victim race per year
ni_race_vic_NSIDH <- race_vic_NSIDH %>% group_by(VIC_RACE, year(OCCUR_DATE)) %>% summarize(new_inc_phtp = sum(new
_inc_phtp)) %>% ungroup() %>% rename(Year = `year(OCCUR_DATE)`) %>% filter(VIC_RACE != "UNKNOWN")

ni_race_vic_NSIDH %>%  ggplot(aes(x = Year, y = new_inc_phtp, color = VIC_RACE)) + geom_line(aes(y = new_inc_phtp
, color = VIC_RACE)) + geom_point(aes(y = new_inc_phtp, color = VIC_RACE)) + theme(legend.position = "bottom") +
labs(x = "year", y = "incidents per 100K people") + scale_x_continuous(breaks = ni_NYC_NSIDH$Year) + scale_y_log1
0()
```

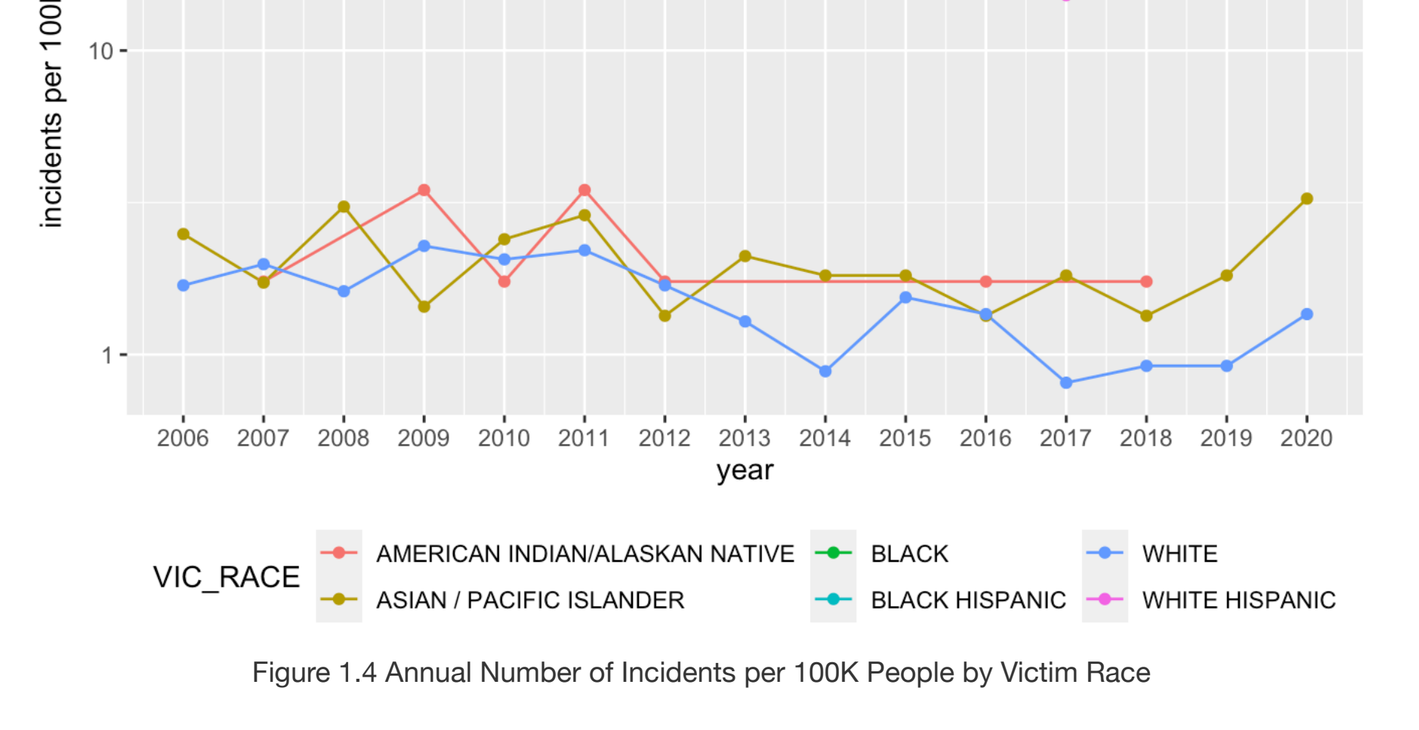

Figure 1.4 Annual Number of Incidents per 100K People by Victim Race

NYC Shooting Activity by Victim Age Groups and Gender, 2006-2020

Four, let's look at the distribution of incidents by victim age group and gender,

```r
pre_NSIDH %>% ggplot(aes(VIC_AGE_GROUP, ..count..)) + geom_bar(aes(fill = VIC_SEX), position = "dodge") + scale_y
_log10()
```

Figure 1.5 Number of Incidents by Victim Age Group and Gender

Correlation between Perpetrator and Victim Race Groups, 2006-2020

Last, let's look at the correlation between perpetrator and victim race groups by statistical murder,

```r
# Correlation matrix between victim race and perpetrator race
pre_NSIDH %>% ggplot(aes(PERP_RACE, VIC_RACE,  color=STATISTICAL_MURDER_FLAG)) + geom_count() + theme(legend.posi
tion = "bottom", axis.text.x = element_text(angle = 15)) +scale_size_area(max_size = 10)
```

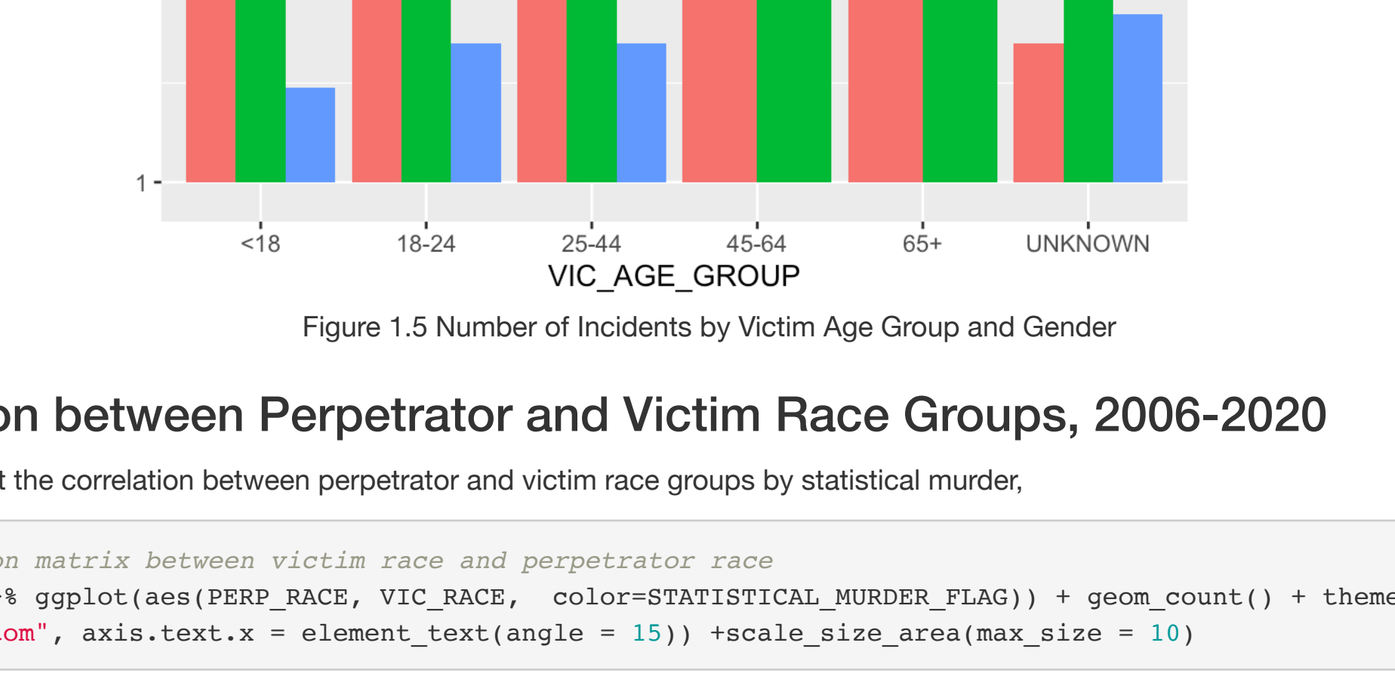

Figure 1.6 Correlation between Perpetrator and Victim Race Groups

```r
# Percentage of perpetrator data classified as "Unknown"

# Number of incidents in NYC per year
perp_NYC_NSIDH <- pre_NSIDH %>% count(PERP_AGE_GROUP, PERP_SEX, PERP_RACE) %>% mutate(Unknown = (PERP_RACE == "UN
KNOWN")) %>% group_by(Unknown) %>% summarise(n = sum(n))

prop_unknown <- round(100*sum(perp_NYC_NSIDH %>% filter(Unknown == TRUE)) / sum(perp_NYC_NSIDH$n),1)
```

Conclusion

- NYC experienced a steady reduction in the number of shooting incidents across all its boroughs from 2006 to 2019, with an estimated interannual decline rate of **-5.1 incidents per 100K people**. However, the number of incidents rebounded sharply in 2020, deleting almost all the progress achieved during the previous 8 years.
- When looking at the performance of the boroughs in 2020, whatever the cause of the increase, it affected all boroughs in almost the same proportion.
- **Bronx** is the NYC borough with the highest number of incidents per 100K people in the last 14 years, followed closely by **Brooklyn**; while **Manhattan**, **Queens** and **Staten Island** have experience a similar rate of shooting activity since 2006; being **Staten Island** the borough with the least prevalence in the shooting activity.
- **Black** and **Black Hispanic** are the racial groups with highest number of victims per 100K people, followed by the **White Hispanic** group; while **White**, **Asian / Pacific Islander** and **American Indian / Alaskan Native** are the groups with the lowest number of victims.
- Since 2006, most of the victims are male, age between 18 and 44 years old. While female, in much less proportion, follows a similar pattern in the prevalence of the age groups victim of the shooting activty.
- Since 2006, in most of the cases the **perpetrator** and the **victim** belong to the **same racial group or ethnicity group**.
- In **43.7%** of the reported incidents, the perpetrator racial profile is unknown, where most of its victims were from **Black** racial group or **Hispanic** ethnicity group.

Possible Sources of Bias

In my opinion, the main source of bias in the dataset is in the classification of the victim and perpetrator by race, as the classification depends on the personal perception of the person who reported the incident, either by the police officer(s) at the crime scene or information collected from witnesses of the incident. However, due to the disproportionately depict in the media of certain racial groups as criminals, and others as victims, people perception is highly suggestive to bias while reporting a crime.

Another potential source of bias is the benchmarking of demographic groups without taking into consideration each group population.

Personal Bias and Mitigations

My main challenge was how to write the report in a language that is, as much as possible, free of bias, especially when referring to racial groups. To mitigate any possible bias in my writing I consulted and applied the **APA Style** guideline for bias-free language regarding to Racial and Ethnic Identity.

Another source of personal bias was on how to deal with missing values and data noises, which affected mostly perpetrator data. I tried to minimize the impact of the personal bias by not eliminating the records but classifying them as "unknown".

```r
sessionInfo()
```

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.7.10 forcats_0.5.1    stringr_1.4.0    dplyr_1.0.6
##  [5] purrr_0.3.4      readr_1.4.0      tidyr_1.1.3      tibble_3.1.0
##  [9] ggplot2_3.3.3    tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.21         haven_2.4.1       colorspace_2.0-1
##  [5] vctrs_0.3.6       generics_0.1.0    htmltools_0.5.1.1 yaml_2.2.1
##  [9] utf8_1.1.4        rlang_0.4.10      pillar_1.6.0      glue_1.4.1
## [13] withr_2.4.1       DBI_1.1.1         dbplyr_2.1.1      modelr_0.1.8
## [17] readxl_1.3.1      lifecycle_1.0.0   munsell_0.5.0     gtable_0.3.0
## [21] cellranger_1.1.0  rvest_1.0.0       evaluate_0.14     labeling_0.4.2
## [25] knitr_1.31        curl_4.3          fansi_0.4.1       highr_0.8
## [29] broom_0.7.6       Rcpp_1.0.6        scales_1.1.1      backports_1.1.8
## [33] jsonlite_1.7.2    farver_2.1.0      fs_1.5.0          hms_1.0.0
## [37] digest_0.6.25     stringi_1.4.6     grid_4.0.4        cli_2.5.0
## [41] tools_4.0.4       magrittr_2.0.1    crayon_1.4.1      pkgconfig_2.0.3
## [45] ellipsis_0.3.1    xml2_1.3.2        reprex_2.0.0      assertthat_0.2.1
## [49] rmarkdown_2.8     httr_1.4.2        rstudioapi_0.13   R6_2.4.1
## [53] compiler_4.0.4
```