NYPD Shooting Incident Project

Chirayu Parikh

5/17/2021

Import Libraries and CSV data, then check column and first few entries of the dataframe.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
rawdata <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   INCIDENT_KEY = col_double(),
##   OCCUR_DATE = col_character(),
##   OCCUR_TIME = col_time(format = ""),
##   BORO = col_character(),
##   PRECINCT = col_double(),
##   JURISDICTION_CODE = col_double(),
##   LOCATION_DESC = col_character(),
##   STATISTICAL_MURDER_FLAG = col_logical(),
##   PERP_AGE_GROUP = col_character(),
##   PERP_SEX = col_character(),
##   PERP_RACE = col_character(),
##   VIC_AGE_GROUP = col_character(),
##   VIC_SEX = col_character(),
##   VIC_RACE = col_character(),
##   X_COORD_CD = col_number(),
##   Y_COORD_CD = col_number(),
##   Latitude = col_double(),
##   Longitude = col_double(),
##   Lon_Lat = col_character()
## )
```

```
summary(rawdata)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:23568       Length:23568      Length:23568
##  1st Qu.: 55317014   Class :character   Class1:hms        Class :character
##  Median : 83365370   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :102218616                      Mode  :numeric
##  3rd Qu.:150772442
##  Max.   :222473262
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.0000    Length:23568       Mode :logical
##  1st Qu.: 44.00   1st Qu.:0.0000    Class :character   FALSE:19080
##  Median : 69.00   Median :0.0000    Mode  :character   TRUE :4488
##  Mean   : 66.21   Mean   :0.3323
##  3rd Qu.: 81.00   3rd Qu.:0.0000
##  Max.   :123.00   Max.   :2.0000
##                   NA's   :2
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##    VIC_SEX            VIC_RACE           X_COORD_CD        Y_COORD_CD
##  Length:23568       Length:23568       Min.   : 914928   Min.   :125757
##  Class :character   Class :character   1st Qu.: 999900   1st Qu.:182565
##  Mode  :character   Mode  :character   Median :1007645   Median :193482
##                                        Mean   :1009363   Mean   :207312
##                                        3rd Qu.:1016807   3rd Qu.:239163
##                                        Max.   :1066815   Max.   :271128
##
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:23568
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Median :40.70   Median :-73.92   Mode  :character
##  Mean   :40.74   Mean   :-73.91
##  3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :40.91   Max.   :-73.70
##
```

```
head(rawdata)
```

```
## # A tibble: 6 x 19
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO          PRECINCT JURISDICTION_CODE
##          <dbl> <chr>      <time>     <chr>            <dbl>              <dbl>
## 1    201575314 08/23/2019 22:10      QUEENS             103                  0
## 2    205748546 11/27/2019 15:54      BRONX               40                  0
## 3    193118596 02/02/2019 19:40      MANHATTAN           23                  0
## 4    204192600 10/24/2019 00:52      STATEN ISLAND      121                  0
## 5    201483468 08/22/2019 18:03      BRONX               46                  0
## 6    198255460 06/07/2019 17:50      BROOKLYN            73                  0
## # ... with 13 more variables: 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>
```

Tidy the data and sort the dataframe by date and time. Since we will not be overlying the datapoints on a map, the location data (latitude, longitude, etc.) has been removed from the dataframe.

```
nypd_shooting <- rawdata %>%
  summarise(Incident_Key=as.numeric(INCIDENT_KEY),
            Date=mdy(OCCUR_DATE),
            Time=OCCUR_TIME,
            Borough=BORO,
            Precinct=as.numeric(PRECINCT),
            Jurisdiction_code=as.numeric(JURISDICTION_CODE),
            Location_Description=LOCATION_DESC,
            Stat_Murder_flag=STATISTICAL_MURDER_FLAG,
            Perp_Age_group=PERP_AGE_GROUP,
            Perp_Sex=PERP_SEX,
            Perp_Race=PERP_RACE,
            Victom_Age_group=VIC_AGE_GROUP,
            Victom_Sex=VIC_SEX,
```

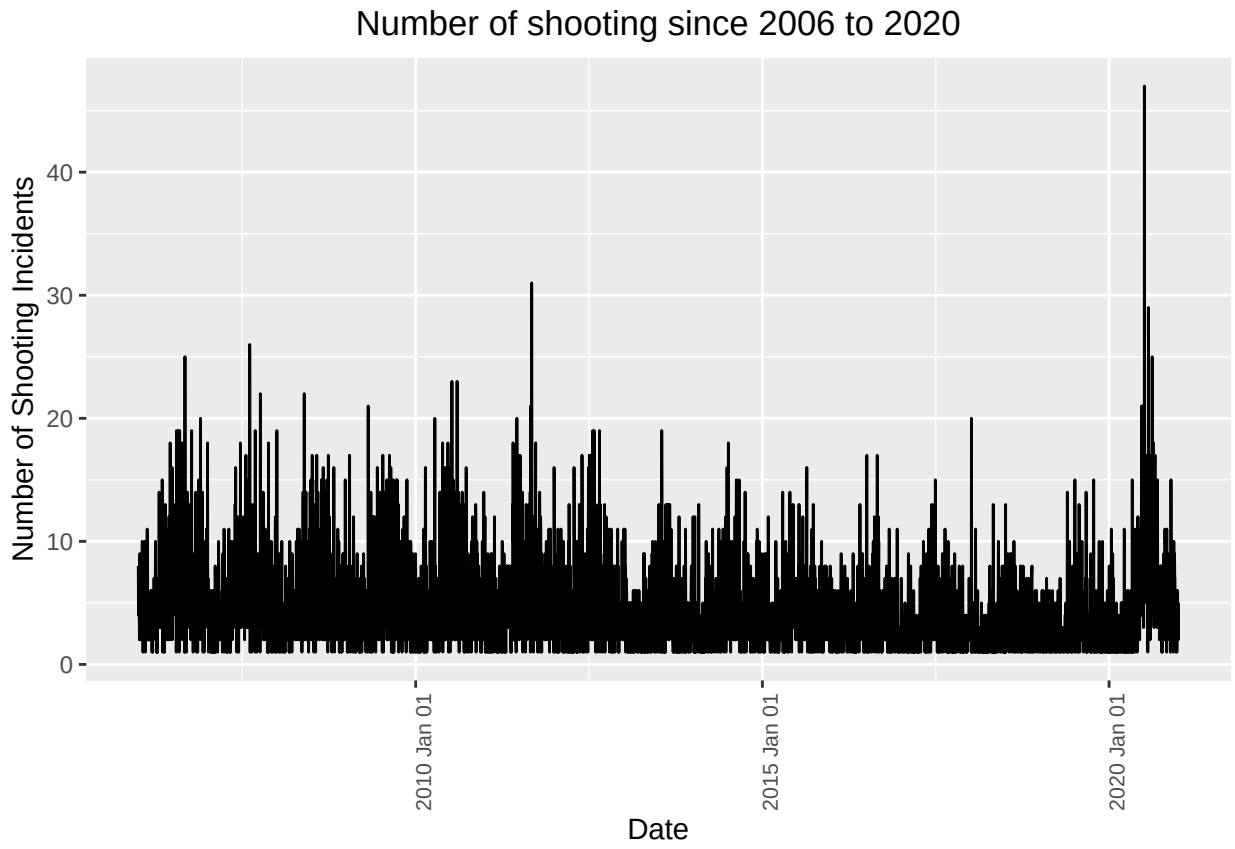
```
            Victom_Race=VIC_RACE)

nypd_sorted <- nypd_shooting[order(nypd_shooting$Date, nypd_shooting$Time),]
head(nypd_sorted)
```

```
## # A tibble: 6 x 14
##   Incident_Key Date       Time   Borough   Precinct Jurisdiction_code
##          <dbl> <date>     <time> <chr>        <dbl>              <dbl>
## 1      9953245 2006-01-01 02:00  BRONX           48                  0
## 2      9953252 2006-01-01 02:22  MANHATTAN       28                  0
## 3      9953250 2006-01-01 02:34  QUEENS         114                  0
## 4      9953250 2006-01-01 02:34  QUEENS         114                  0
## 5      9953247 2006-01-01 03:30  BROOKLYN        67                  0
## 6      9953246 2006-01-01 05:51  BRONX           44                  0
## # ... with 8 more variables: Location_Description <chr>,
## #   Stat_Murder_flag <lgl>, Perp_Age_group <chr>, Perp_Sex <chr>,
## #   Perp_Race <chr>, Victom_Age_group <chr>, Victom_Sex <chr>,
## #   Victom_Race <chr>
```

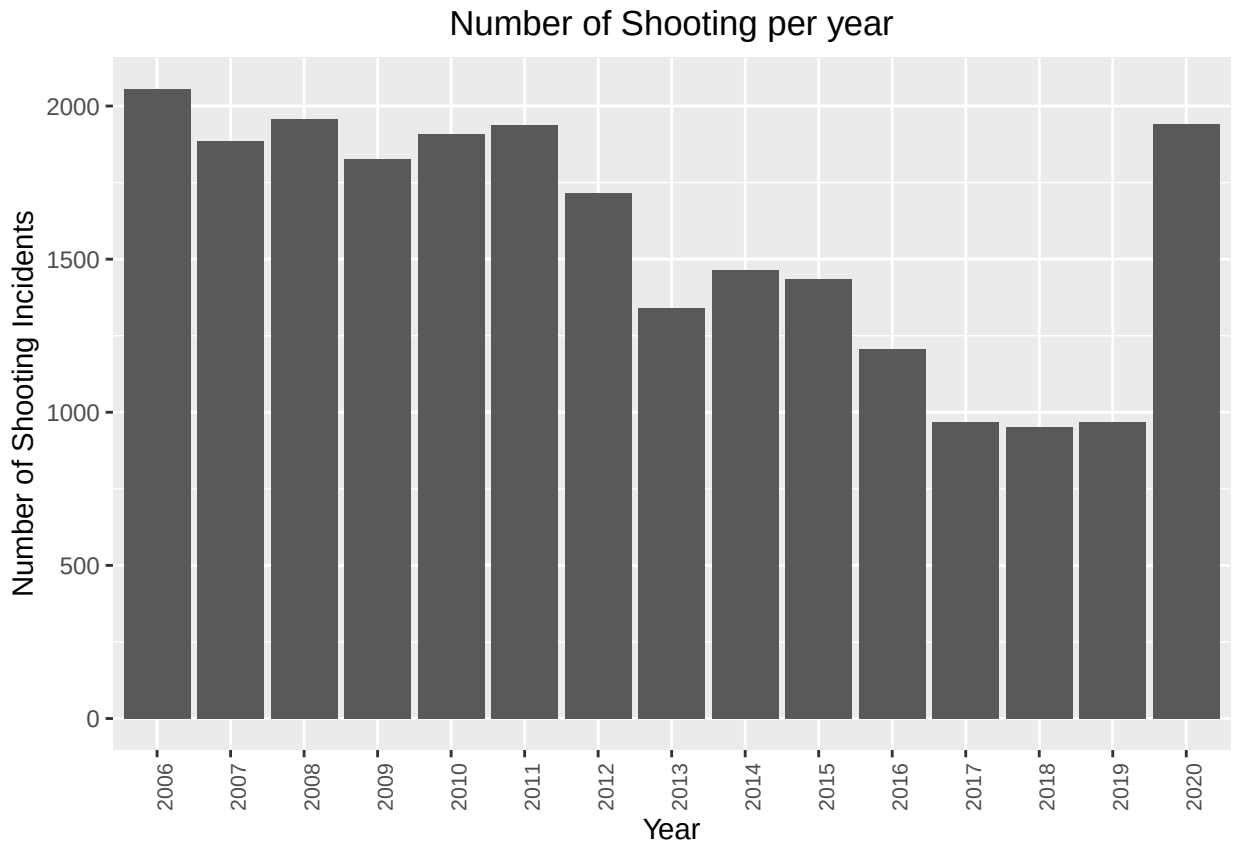
Graph the shooting data by date, we can see the trend in NYC shootings over time. The first graph is very dense, due to large number of datapoints. However it shows slight downward trend in shooting incidents until a large spike in 2020. Further, seperating date into days, months and years and plotting the yearly shooting incidently we find the raise in incidends in 2020 was from July. My theory on the spike in 2020 is that, it was during the time when NYC among other places had experiance riots due to the killing of Geroge Floyd and Black Lifes Matter protest that followed afterwards. That being said, the trend in general is downwards for shooting incidents in NYC.

```
nypd_sorted %>%
  group_by(Date) %>%
  ggplot(aes(x = Date)) +
  geom_line(stat="count") +
  scale_x_date(date_labels = "%Y %b %d") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Date",
       y = "Number of Shooting Incidents",
       title = "Number of shooting since 2006 to 2020")
```

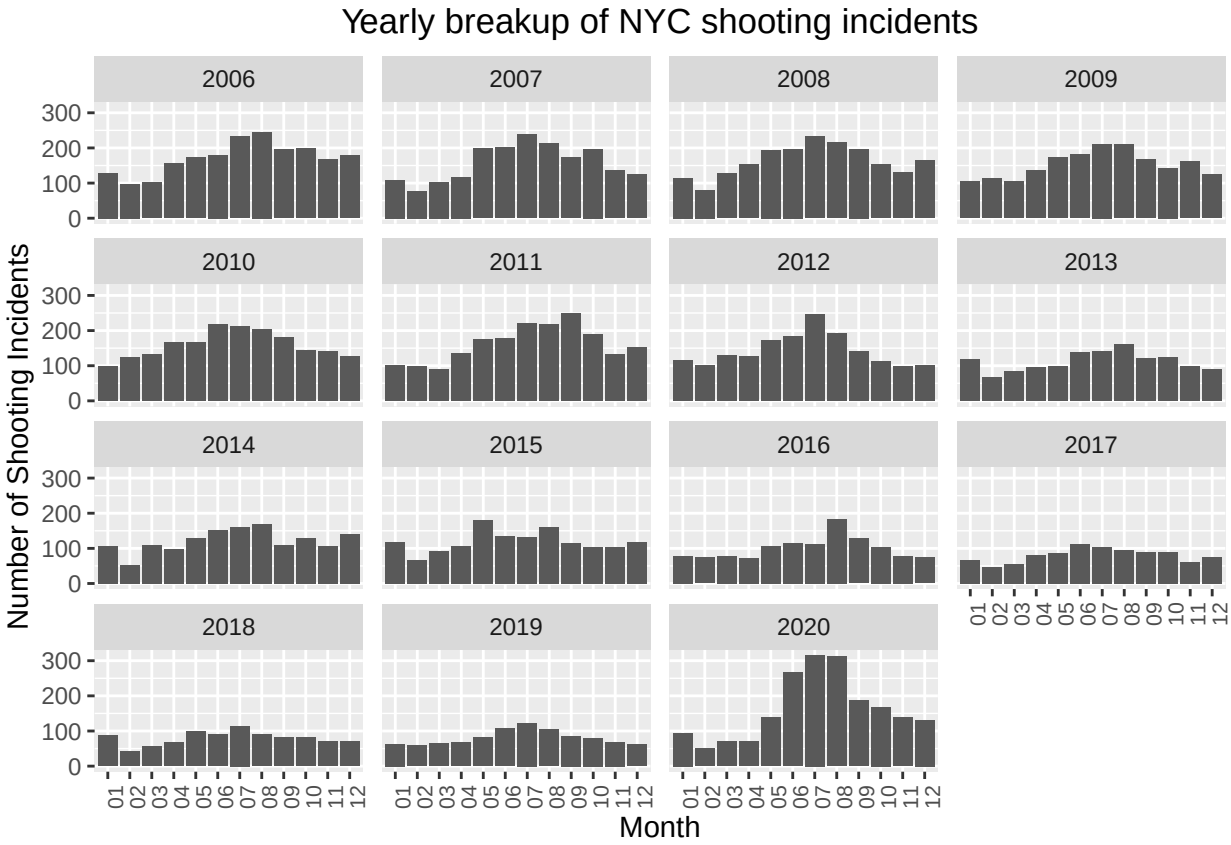



```
nypd_year <- rawdata %>%
  mutate(Month = str_sub(OCCUR_DATE,1,2)) %>% # Seperating the Month
  mutate(Day = str_sub(OCCUR_DATE,4,5)) %>% # Seperating the Day
  mutate(Year = str_sub(OCCUR_DATE,7)) # Seperating the year

nypd_year %>%
  group_by(Year) %>%
  ggplot(aes(x = Year)) +
  geom_bar(stat="count") +
  #theme(legend.position = "bottom", axis.text.x = element_text(angle = 90))
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Year",
       y = "Number of Shooting Incidents",
       title = "Number of Shooting per year")
```

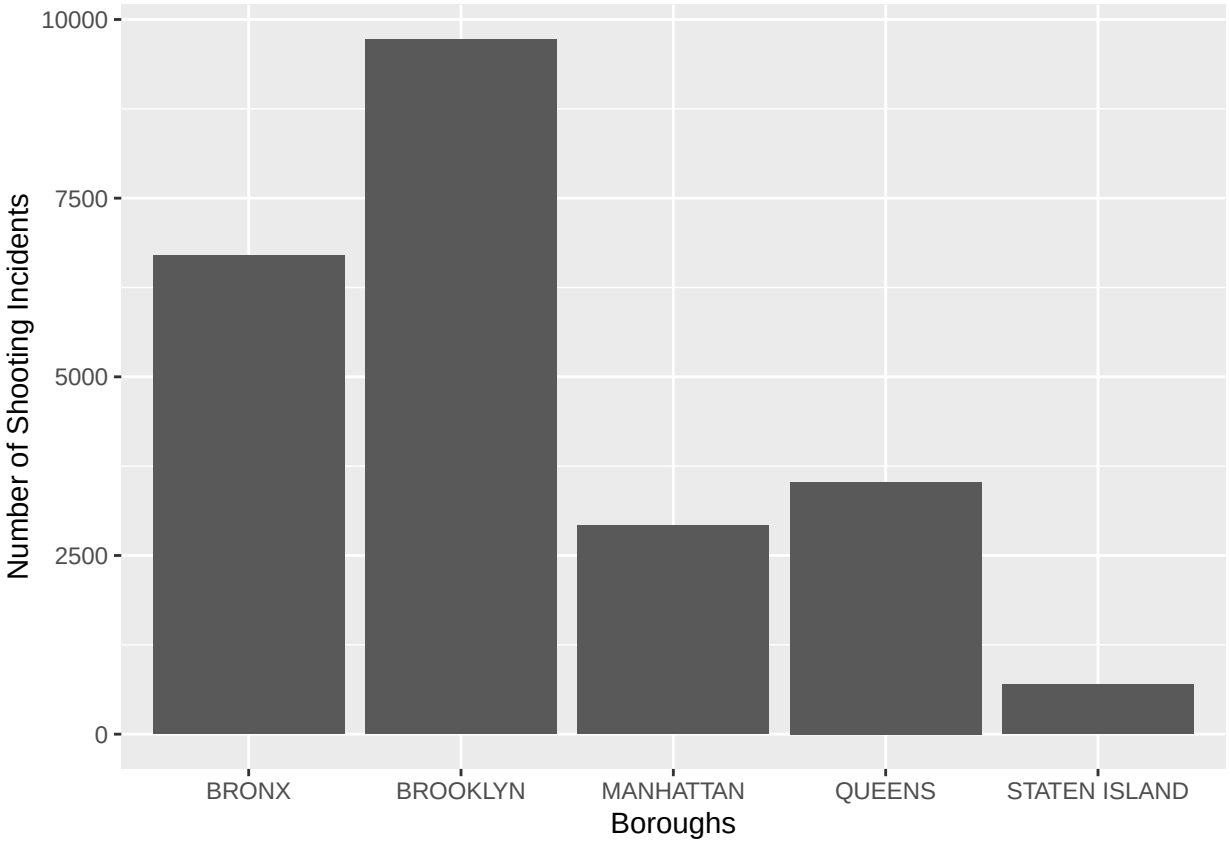



```
nypd_year %>%
  group_by(Year) %>%
  ggplot(aes(x = Month)) +
  geom_bar(stat="count") +
  facet_wrap( ~ Year) +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Month",
       y = "Number of Shooting Incidents",
       title = "Yearly breakup of NYC shooting incidents")
```



Boroughs wise graph of shooting incidents in NYC. New York City is composed of five boroughs, they are, Bronx, Brooklyn, Manhattan, Queens, and Staten Island. We see that Brooklyn has the highest shooting incidents follow by Brox. As expected Staten Island has the lowest incident of the five boroughs.

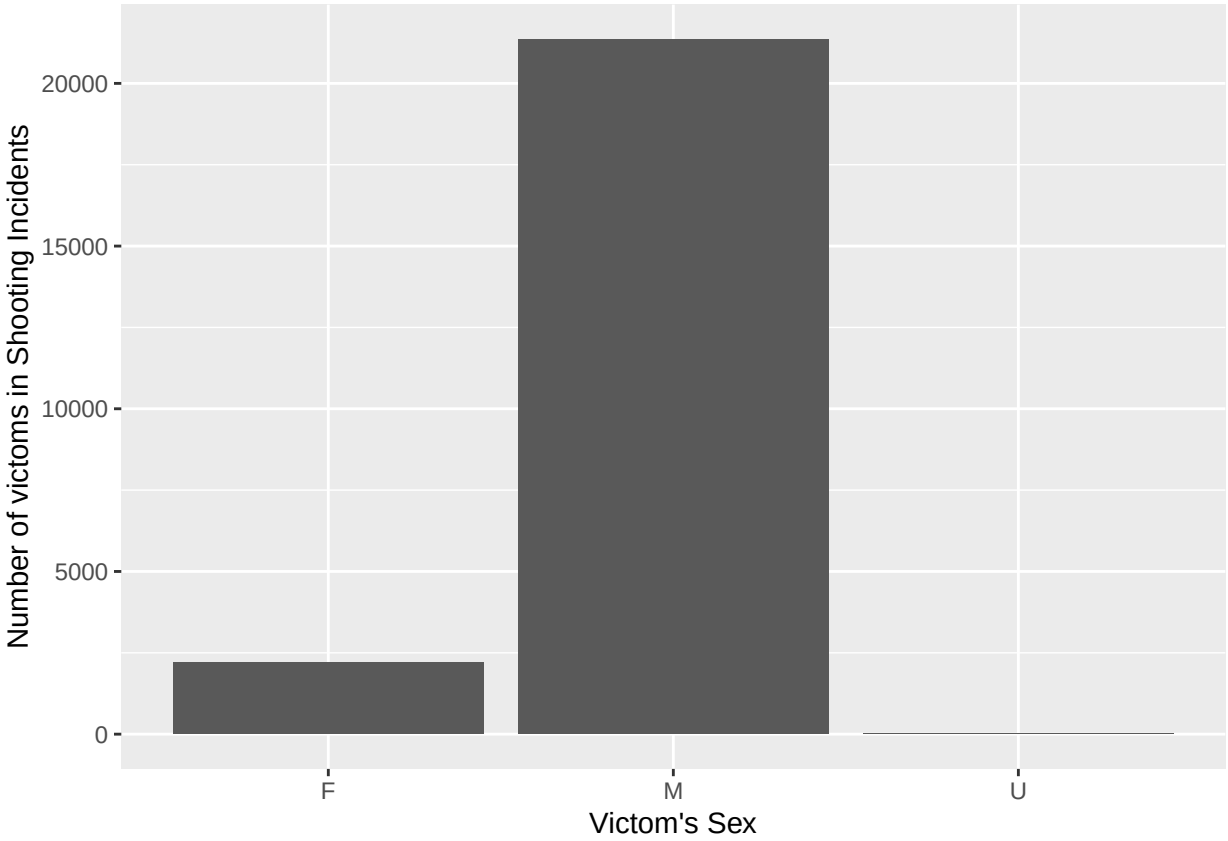
```
qplot(nypd_sorted$Borough, xlab = "Boroughs", ylab = "Number of Shooting Incidents")
```



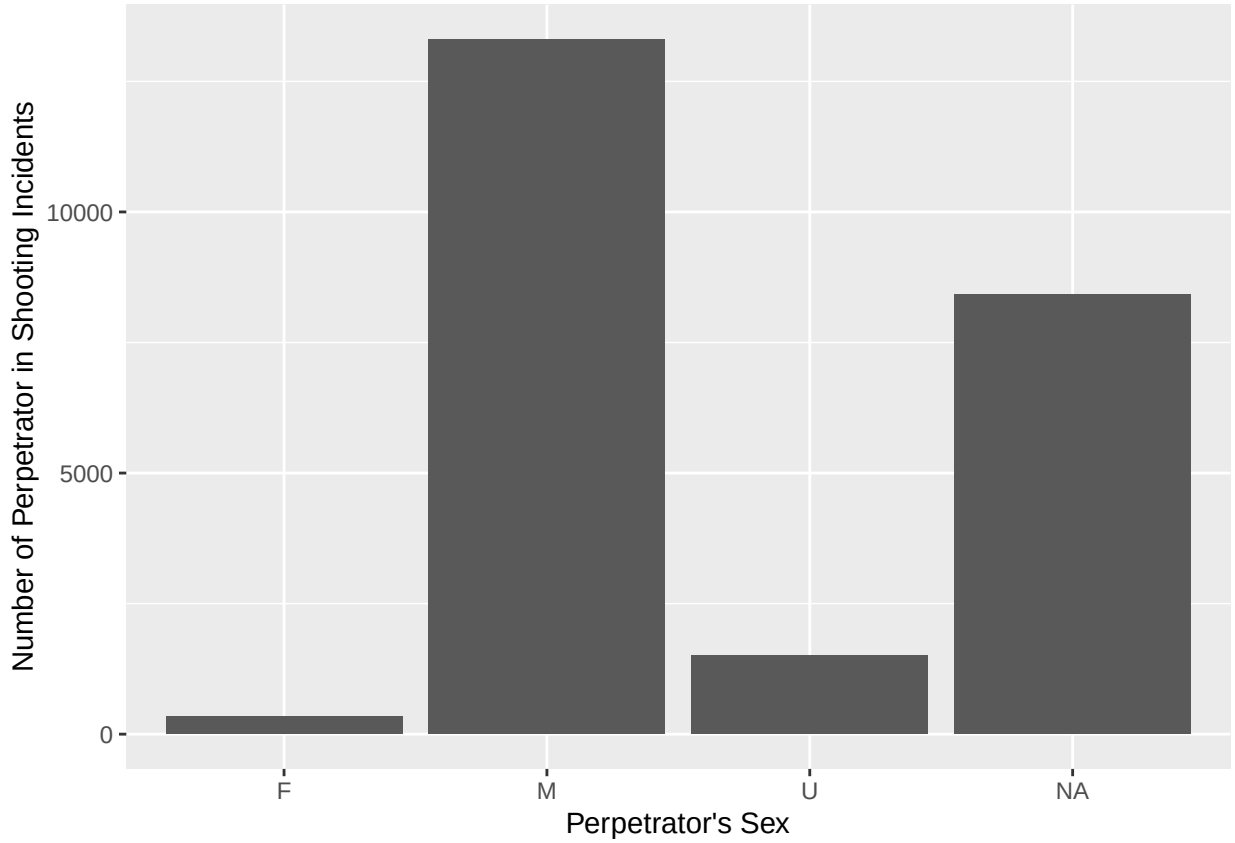
We now plot the perpetrator and victoms based on sex, racial profile and age group. We see that majority of the Perpetrators and victoms are male, about 90 percent of them. We also see that perpetrators and victoms of the shooting incidents are predominant black. Finally, we see the that large number of incidents fall in age group from 18 to 44 years for both perpetrators and victoms.

One intersting observaton for perpetrators dataset is that we see some misisng data points, for instance 'NA' in the field. This maybe because information about the perpetrator is still not known. For example, if there was a shooting event and by the time it was reported the perpetrator(s) could have fled the scene.

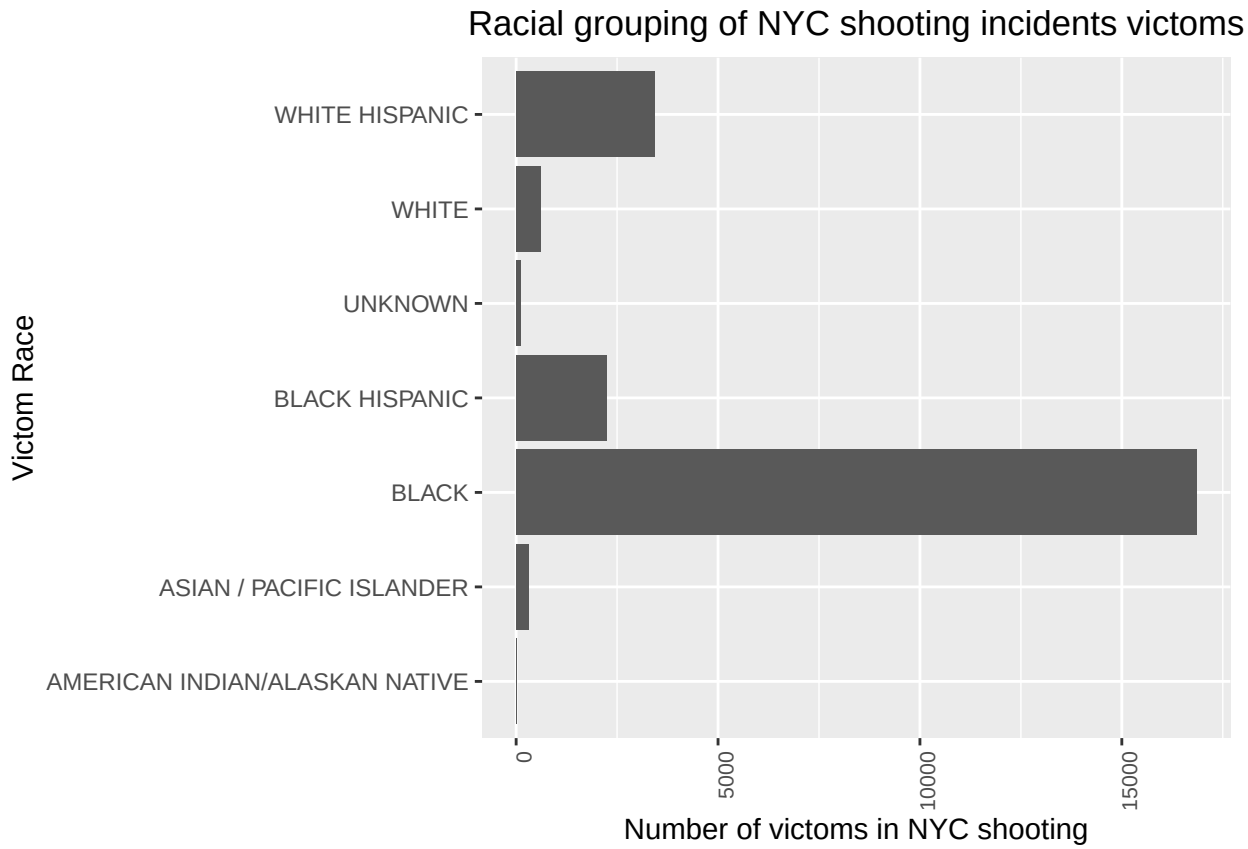
```
qplot(nypd_sorted$Victom_Sex,
      xlab = "Victom's Sex",
      ylab = "Number of victoms in Shooting Incidents")
```



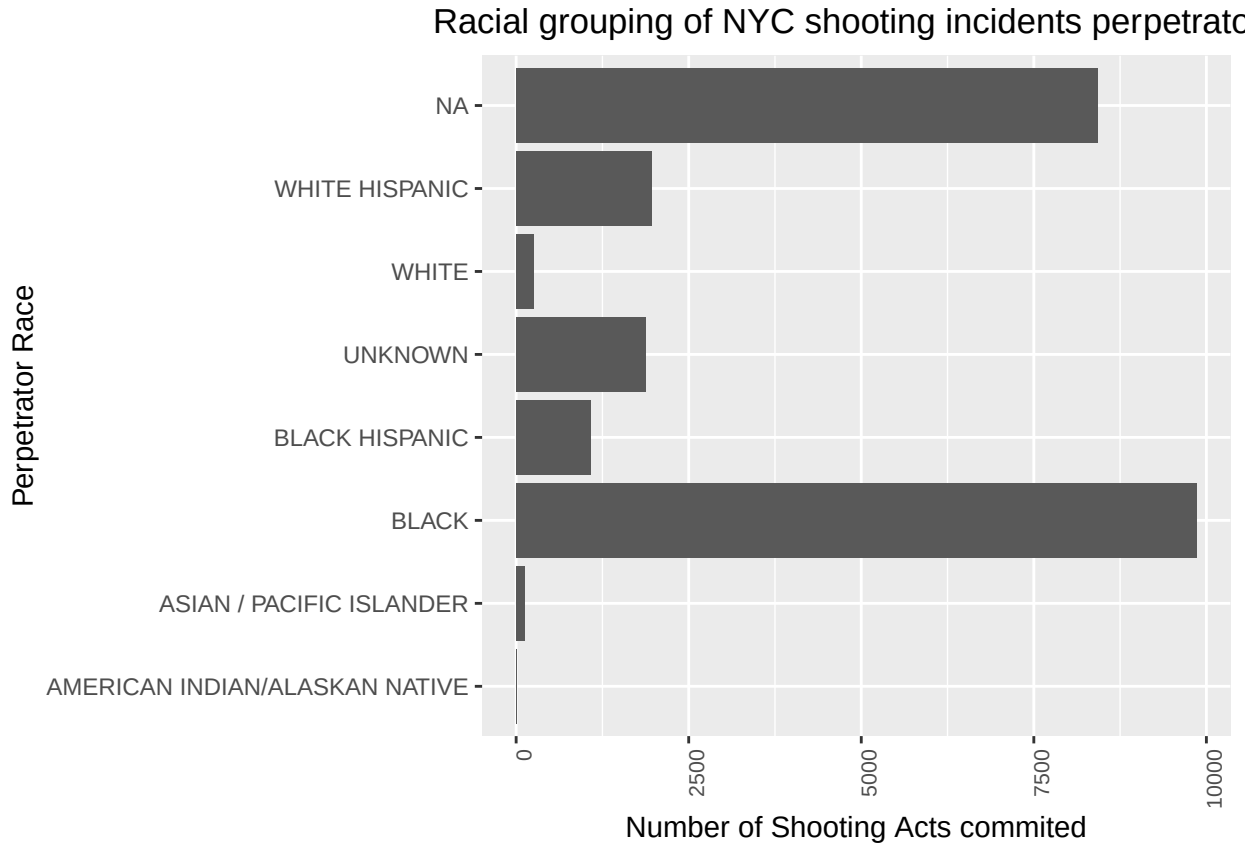

```
qplot(nypd_sorted$Perp_Sex,
      xlab = "Perpetrator's Sex",
      ylab = "Number of Perpetrator in Shooting Incidents")
```



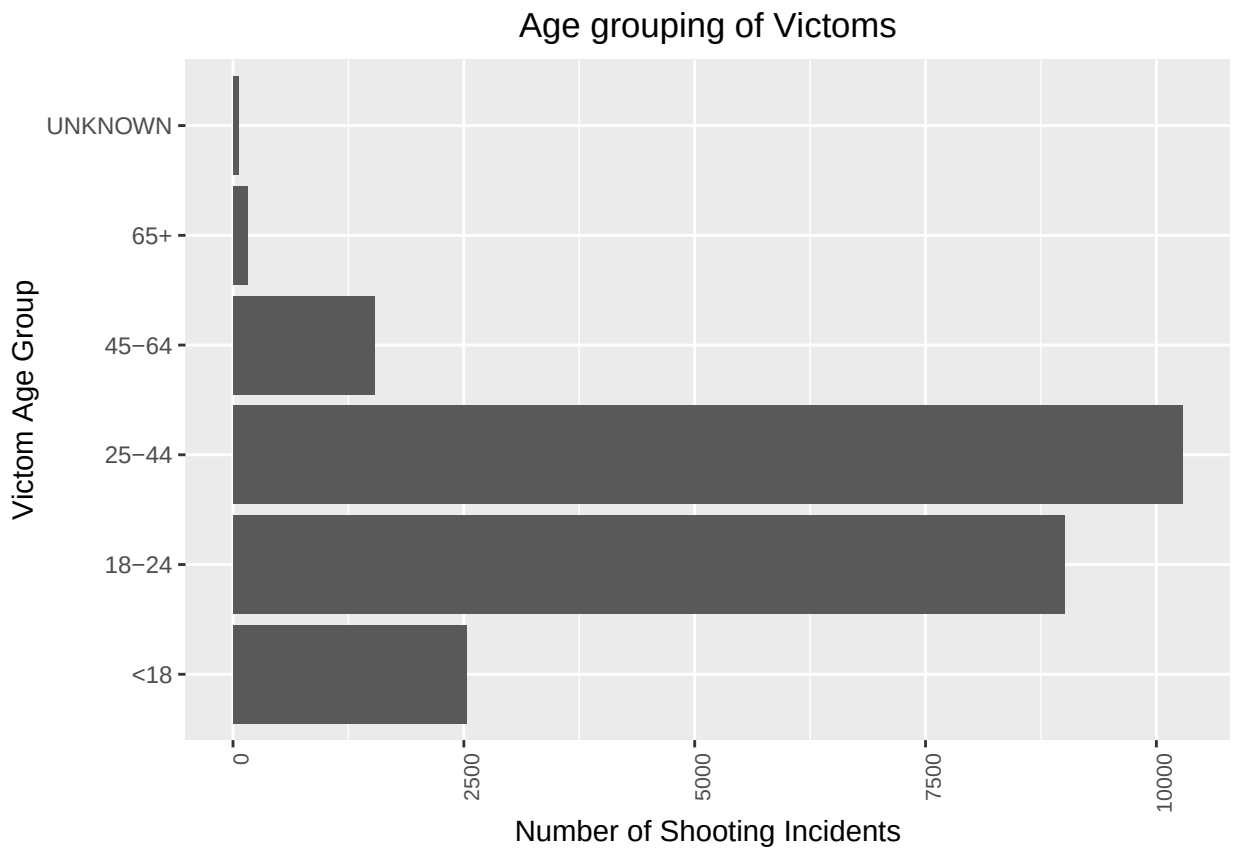

```
nypd_sorted %>%
  group_by(Victom_Race) %>%
  ggplot(aes(x = Victom_Race)) +
  geom_bar(stat="count") +
  coord_flip() +
  #theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Victom Race",
       y = "Number of victoms in NYC shooting",
       title = "Racial grouping of NYC shooting incidents victoms")
```



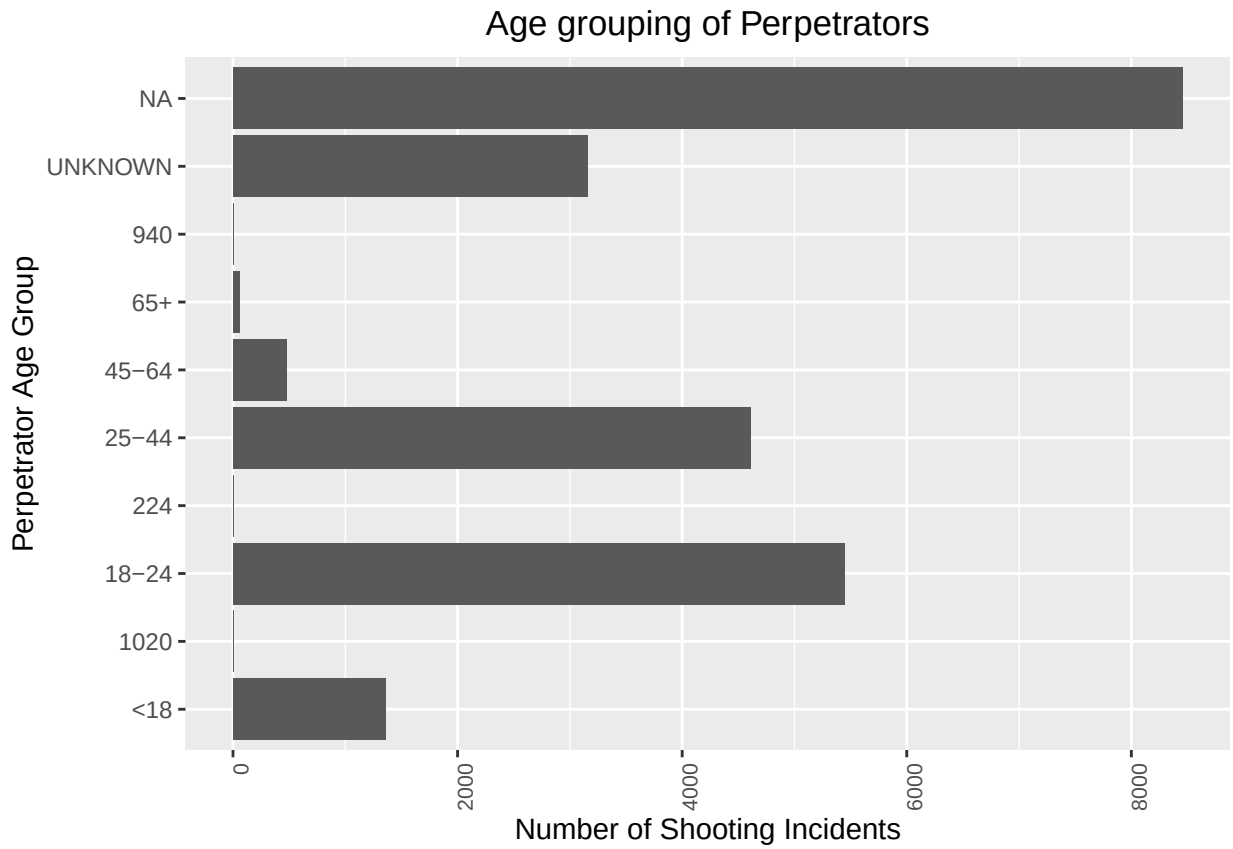

```
nypd_sorted %>%
  group_by(Perp_Race) %>%
  ggplot(aes(x = Perp_Race)) +
  geom_bar(stat="count") +
  coord_flip() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Perpetrator Race",
       y = "Number of Shooting Acts commited",
       title = "Racial grouping of NYC shooting incidents perpetrators")
```




```
nypd_sorted %>%
  group_by(Victom_Age_group) %>%
  ggplot(aes(x = Victom_Age_group)) +
  geom_bar(stat="count") +
  coord_flip() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Victom Age Group",
       y = "Number of Shooting Incidents",
       title = "Age grouping of Victoms")
```




```
nypd_sorted %>%
  group_by(Perp_Age_group) %>%
  ggplot(aes(x = Perp_Age_group)) +
  geom_bar(stat="count") +
  coord_flip() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x=element_text(angle=90, hjust=1, size=8),
        panel.spacing.x=unit(0.5, "lines")) +
  labs(x = "Perpetrator Age Group",
       y = "Number of Shooting Incidents",
       title = "Age grouping of Perpetrators")
```



Calculate male to female percentage for Perpetrator and Victoms

```
# Calculating percentage of Male vs Female for both Perpetrator and Victoms
all_female_victoms <- dplyr::filter(nypd_sorted, Victom_Sex %in% "F")
all_male_victoms <- dplyr::filter(nypd_sorted, Victom_Sex %in% "M")

percent_female_vic <- (nrow(all_female_victoms) / nrow(nypd_sorted)) * 100
percent_male_vic <- (nrow(all_male_victoms) / nrow(nypd_sorted)) * 100

all_female_perp <- dplyr::filter(nypd_sorted, Perp_Sex %in% "F")
all_male_perp <- dplyr::filter(nypd_sorted, Perp_Sex %in% "M")

percent_female_perp <- (nrow(all_female_perp) / nrow(nypd_sorted)) * 100
percent_male_perp <- (nrow(all_male_perp) / nrow(nypd_sorted)) * 100

percent_female_perp
```

```
## [1] 1.417176
```

```
percent_male_perp
```

```
## [1] 56.45367
```

```
percent_female_vic
```

```
## [1] 9.313476
```

```
percent_male_vic
```

```
## [1] 90.60166
```

Using Naive Model which will do a simple prediction of victom's race with the most occurrences in the data set. The model however, is not very accurate with accuracy of 71 percentage.

```
# Factorize victoms race
nypd_sorted$Victom_Race <- factor(nypd_sorted$Victom_Race)

y <- nypd_sorted$Victom_Race
test_index <- createDataPartition(y, times = 1, p = 0.2, list = FALSE)
train_set <- nypd_sorted %>% slice(-test_index)
test_set <- nypd_sorted %>% slice(test_index)

naive_guess <- train_set %>%
  group_by(Victom_Race) %>%
  summarize(count = n()) %>%
  filter(count == max(count)) %>%
  pull(Victom_Race)
y_naive <- test_set %>%
  mutate(y_hat = naive_guess) %>%
  pull(y_hat)
naive_acc <- confusionMatrix(y_naive, reference = test_set$Victom_Race)$overall["Accuracy"]

naive_acc
```

```
##  Accuracy
## 0.7145886
```

In conclusion, there is an big spike in the shooting incident's in year 2020. However, in genral we see downward trend in the number of incident's. The dataset does not include demographics and other economic data. Therefore, this data would need to be used with other dataset's to do further analysis. This would be one of the bias in this analysis. Also since I am very familer with crime statistics and crime related data. This kind of analysis would require multi domain expertise and combing different ecomonic and demographic dataset's. There is also some nuances about different cities, which can also add bias.