NYPDShootingData

2024-03-25

Introduction

This report shows a list of every shooting incident that occurred in the city of New York from 2006 to 2022. Each record represents a shooting incident in NYC and includes information about the event, the location, and the time of occurrence.

Libraries.

```
library(lubridate)
```

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

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(dplyr)
library(ggpubr)
```

Importing Dataset and reading data from csv file

First we need to import the required dataset from the following source: https://catalog.data.gov/dataset. After importing the dataset, proceed with the following steps:

1.) Search for the dataset titled 'NYPD Shooting Incident Data. 2.) Right-click on the CSV button next to the title and copy the link address into your respective RMD document.

The following code completes this process and reads in the data in CSV format.

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

Tidy and Transform

Let us explore the column names present in the data set, and we can start cleaning up the data so that we can transform and use those data in our further analysis.

```
colnames(nypd_data)
```

```
##  [1] "INCIDENT_KEY"            "OCCUR_DATE"
##  [3] "OCCUR_TIME"              "BORO"
##  [5] "LOC_OF_OCCUR_DESC"       "PRECINCT"
##  [7] "JURISDICTION_CODE"       "LOC_CLASSFCTN_DESC"
##  [9] "LOCATION_DESC"           "STATISTICAL_MURDER_FLAG"
## [11] "PERP_AGE_GROUP"          "PERP_SEX"
## [13] "PERP_RACE"               "VIC_AGE_GROUP"
## [15] "VIC_SEX"                 "VIC_RACE"
## [17] "X_COORD_CD"              "Y_COORD_CD"
## [19] "Latitude"                "Longitude"
## [21] "Lon_Lat"
```

Removing Unnecessary Columns

From the above columns, we could remove some of the unnecessary columns that are not useful for our analysis.

```
nypd_data <- nypd_data %>% select(-c(
                      JURISDICTION_CODE,
                      Lon_Lat,
                      X_COORD_CD,
                      Y_COORD_CD,
                      Longitude,
                      Latitude
                      ))
```

Converting OCCUR_DATE to date data type and time to time object

Now, we can convert the OCCUR_DATE, OCCUR_TIME variable to a date so that the month, year, and day can be extracted for further analysis.

```
nypd_data <- nypd_data %>%
            mutate(OCCUR_DATE = as.Date(OCCUR_DATE, "%m/%d/%Y"))
nypd_data$OCCUR_TIME <- hms(nypd_data$OCCUR_TIME)
nypd_data <- nypd_data %>%
            mutate(YEAR = format(nypd_data$OCCUR_DATE,"%Y"))
```

Convert categorical data to factors

We can further clean up the data by converting categorical data into factors

```
nypd_data <- nypd_data %>%
  mutate(BORO = as.factor(BORO)) %>%
  mutate(VIC_AGE_GROUP = as.factor(VIC_AGE_GROUP)) %>%
  mutate(VIC_SEX = as.factor(VIC_SEX)) %>%
  mutate(PERP_SEX = as.factor(PERP_SEX)) %>%
  mutate(PERP_RACE = as.factor(PERP_RACE)) %>%
  mutate(PERP_AGE_GROUP = as.factor(PERP_AGE_GROUP)) %>%
  mutate(VIC_RACE = as.factor(VIC_RACE))
```

Handling the missing values by specifying 'Unknown' in PERP_AGE_GROUP,PERP_SEX, columns

In this dataset, we could see some missing values and data noise. We can handle those NA values by representing them as "unknown" in the columns PERP_AGE_GROUP, PERP_SEX, and PERP_RACE.

```
nypd_data$STATISTICAL_MURDER_FLAG <- as.logical(nypd_data$STATISTICAL_MURDER_FLAG)
```

```
nypd_data <- nypd_data %>%
  mutate(PERP_AGE_GROUP = replace_na(PERP_AGE_GROUP, "Unknown"),
         PERP_SEX = replace_na(PERP_SEX, "Unknown"),
         PERP_RACE = replace_na(PERP_RACE, "Unknown"))
```

Now let us summarize the dataset that we have cleaned and transformed.

```
#summary of data
summary(nypd_data)
```

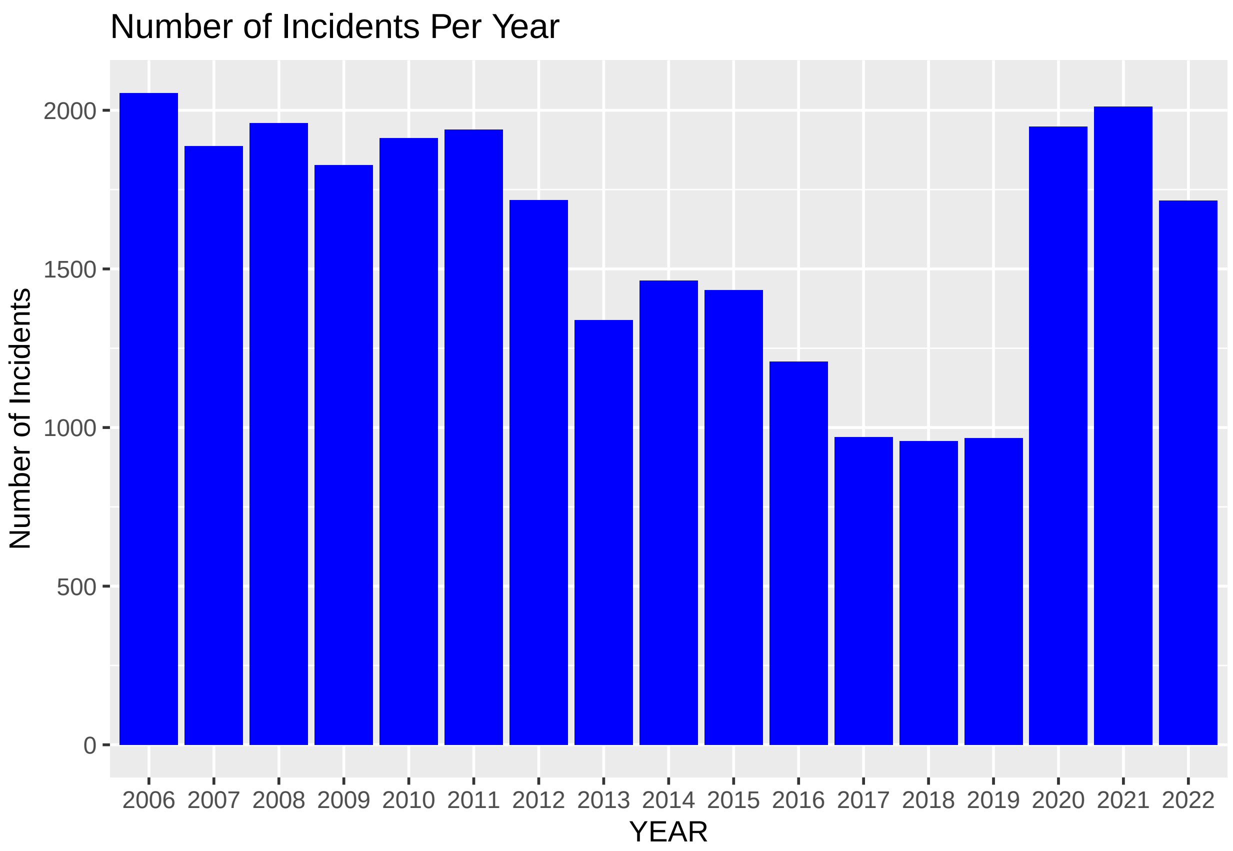
```
##   INCIDENT_KEY         OCCUR_DATE           OCCUR_TIME
##  Min.   :  9953245   Min.   :2006-01-01   Min.   :0S
##  1st Qu.: 63860880   1st Qu.:2009-07-18   1st Qu.:3H 27M 0S
##  Median : 90372218   Median :2013-04-29   Median :15H 11M 0S
##  Mean   :120860536   Mean   :2014-01-06   Mean   :12H 41M 31.7091388400731S
##  3rd Qu.:188810230   3rd Qu.:2018-10-15   3rd Qu.:20H 45M 0S
##  Max.   :261190187   Max.   :2022-12-31   Max.   :23H 59M 0S
##
##            BORO       LOC_OF_OCCUR_DESC     PRECINCT      LOC_CLASSFCTN_DESC
##  BRONX        : 7937   Length:27312       Min.   :  1.00   Length:27312
##  BROOKLYN     :10933   Class :character   1st Qu.: 44.00   Class :character
##  MANHATTAN    : 3572   Mode  :character   Median : 68.00   Mode  :character
##  QUEENS       : 4094                      Mean   : 65.64
##  STATEN ISLAND:  776                      3rd Qu.: 81.00
##                                           Max.   :123.00
##
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP   PERP_SEX
##  Length:27312       Mode :logical                  :9344         : 9310
##  Class :character   FALSE:22046             18-24  :6222   (null):  640
##  Mode  :character   TRUE :5266              25-44  :5687   F     :  424
##                                             UNKNOWN:3148   M     :15439
##                                              <18    :1591   U     : 1499  
##                                              (null) : 640                 
##                                              (Other): 680                 
##           PERP_RACE     VIC_AGE_GROUP   VIC_SEX  
##  BLACK         :11432   <18    : 2839   F: 2615  
##                : 9310   1022   :    1   M:24686  
##  WHITE HISPANIC: 2341   18-24  :10086   U:   11  
##  UNKNOWN       : 1836   25-44  :12281            
##  BLACK HISPANIC: 1314   45-64  : 1863            
##  (null)        :  640   65+    :  181            
##  (Other)       :  439   UNKNOWN:   61            
##                            VIC_RACE         YEAR          
##  AMERICAN INDIAN/ALASKAN NATIVE:   10   Length:27312      
##  ASIAN / PACIFIC ISLANDER      :  404   Class :character  
##  BLACK                         :19439   Mode  :character  
##  BLACK HISPANIC                : 2646                     
##  UNKNOWN                       :   66                     
##  WHITE                         :  698                     
##  WHITE HISPANIC                : 4049                     
```

From the above output, it is evident that there are no null values present in the dataset, which means the dataset is cleaned and ready to use.

Visualizaton and Analysis

Data visualization is the graphical representation of information and data. First, I would like to visualize the total number of incidents occurred in each year from (2006-2022)

```r
year_of_incidents <- nypd_data %>% count(YEAR)
ggplot(data=year_of_incidents, aes(x=YEAR, y=n)) +
       geom_bar(stat="identity", fill="blue") +
       labs(title = str_c("Number of Incidents Per Year"),
            y = str_c("Number of Incidents"),
            x = str_c("YEAR"))
```

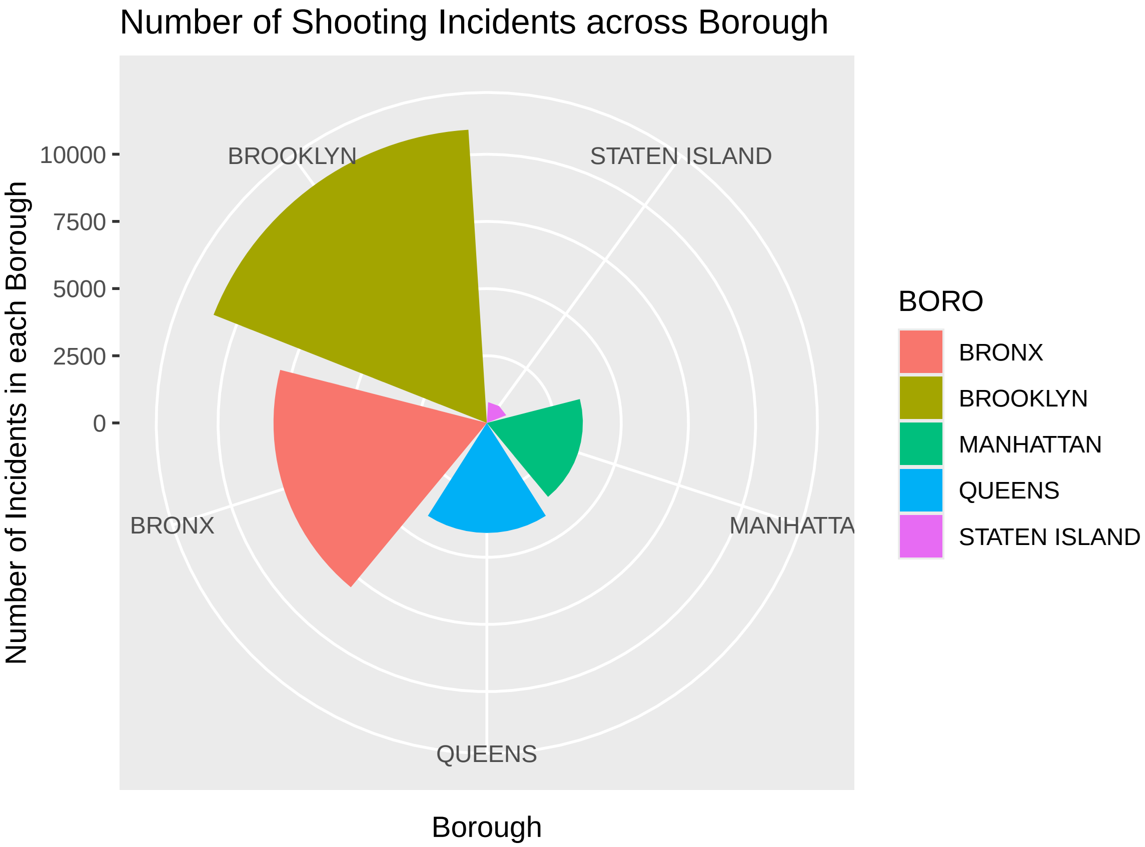


From this bar chart, we could see the number of incidents that occurred in each year from 2006 to 2022. We could see a decrease in the incident rate from 2017 to 2019.

Distribution of Incidents Across Boroughs

The below graph shows the distribution of incidents in each borough in New York.

```r
nypd_data %>%
  count(BORO) %>%
  ggplot(aes(x = reorder(BORO, n), y = n, fill = BORO)) +
  geom_bar(stat = "identity") +
  scale_colour_viridis_b() +
  labs(title = "Number of Shooting Incidents across Borough",
       x = "Borough",
       y = "Number of Incidents in each Borough") +
  coord_polar()
```

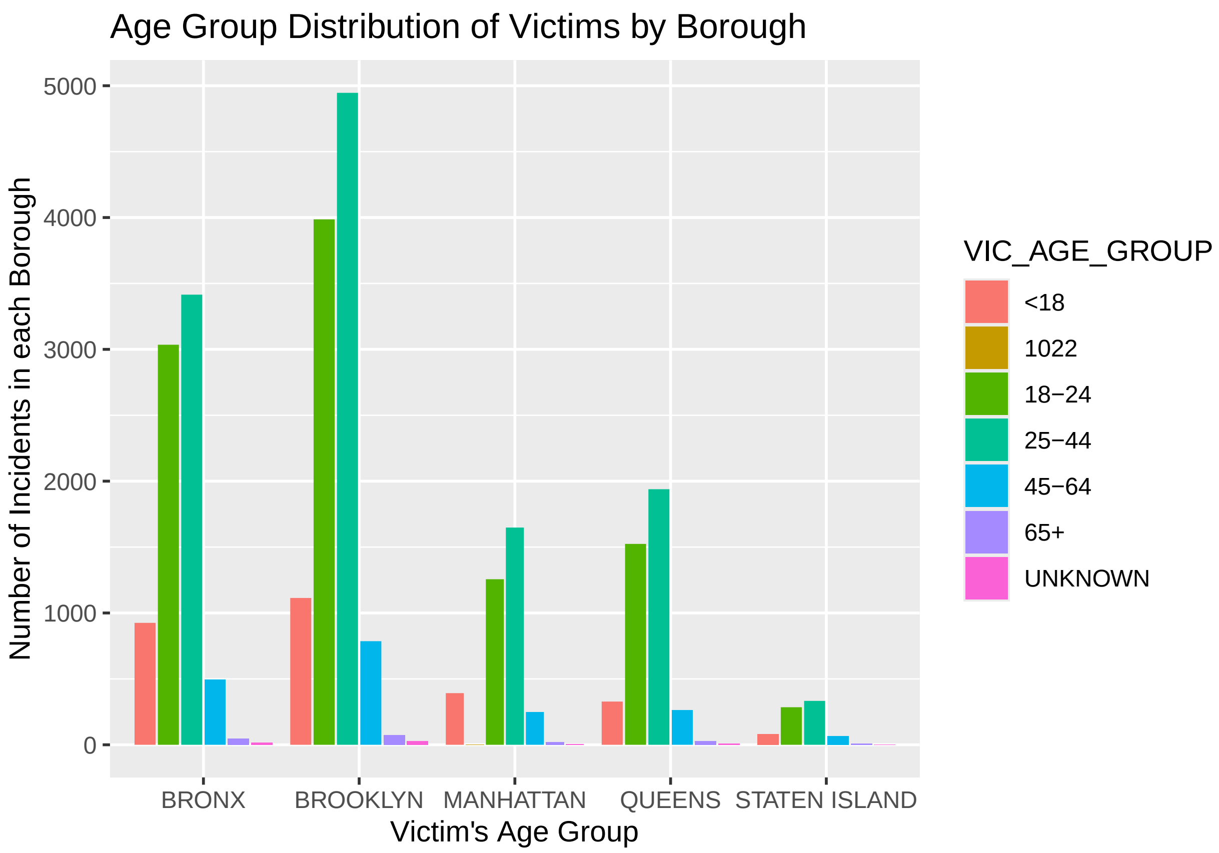


In this graph, we could see that the “Brooklyn” borough in NY has the highest number of incidents reported when compared to other boroughs. Also, “Staten Island” borough has reported the least number of incidents.

Age Group Distribution of Victims by Borough

Here we can visualize the age group distribution of victims by Borough. The below horizontal bar chart represents this visually.

```
ggplot(nypd_data, aes(x = factor(BORO), fill = VIC_AGE_GROUP)) +
  geom_bar(position = "dodge2") +
  labs(title = "Age Group Distribution of Victims by Borough",
       x = "Victim's Age Group",
       y = "Number of Incidents in each Borough") +
  coord_trans()
```

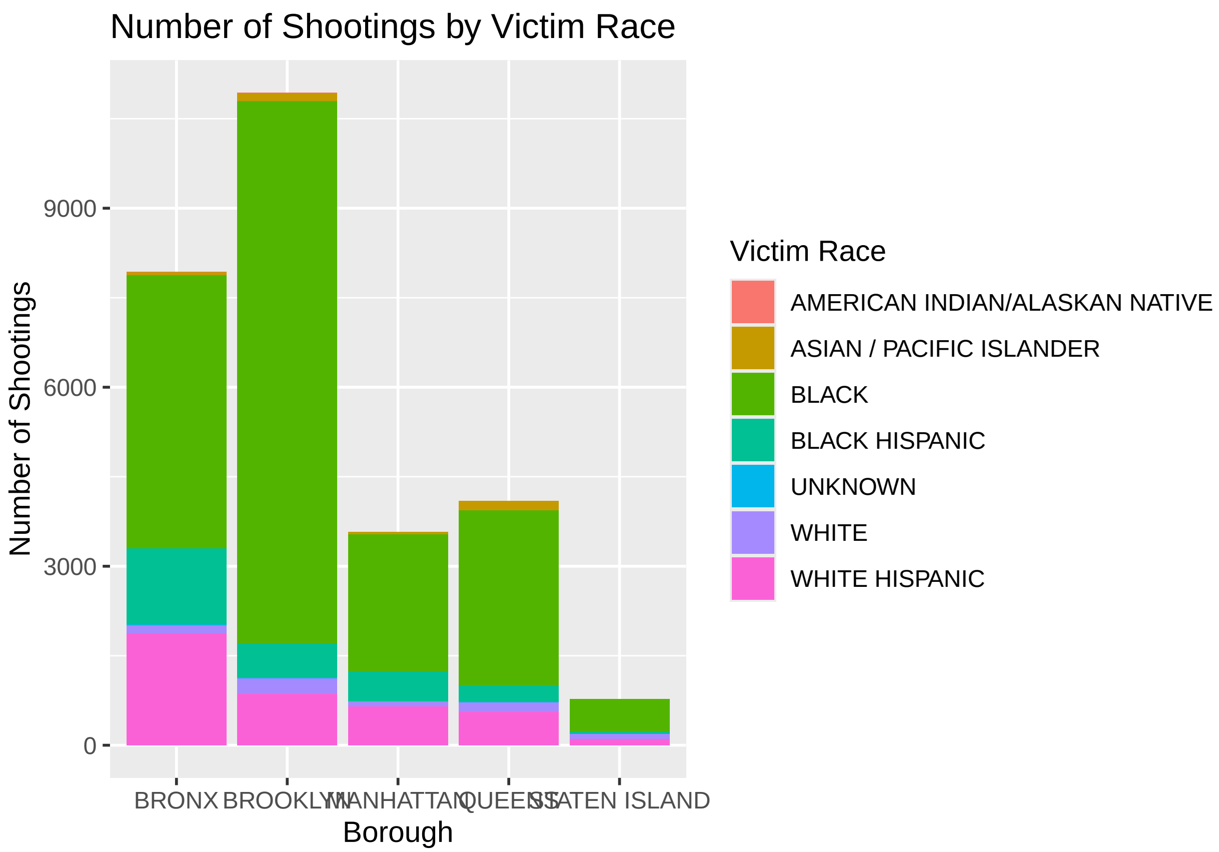


From the above graph, we could see that the age group of "25-44" has the highest number of victims in each borough. Also, the age group above 65+ has a smaller number of victims.

Distribution of Number of Shooting Incidents by Victim Race

Let us now visualize the distribution of number of shooting incidents by Victim's race.

```r
nypd_data %>%
  ggplot(aes(x = factor(BORO), fill = VIC_RACE)) +
  geom_bar(position = "stack") +
  labs(x = "Borough",
       y = "Number of Shootings",
       fill = "Victim Race",
       title = "Number of Shootings by Victim Race")
```



From this visualization, we could predict that most of the victims are BLACK, which has a higher number in each borough. Also, we could see that WHITE HISPANIC's are less significant victims in each borough.

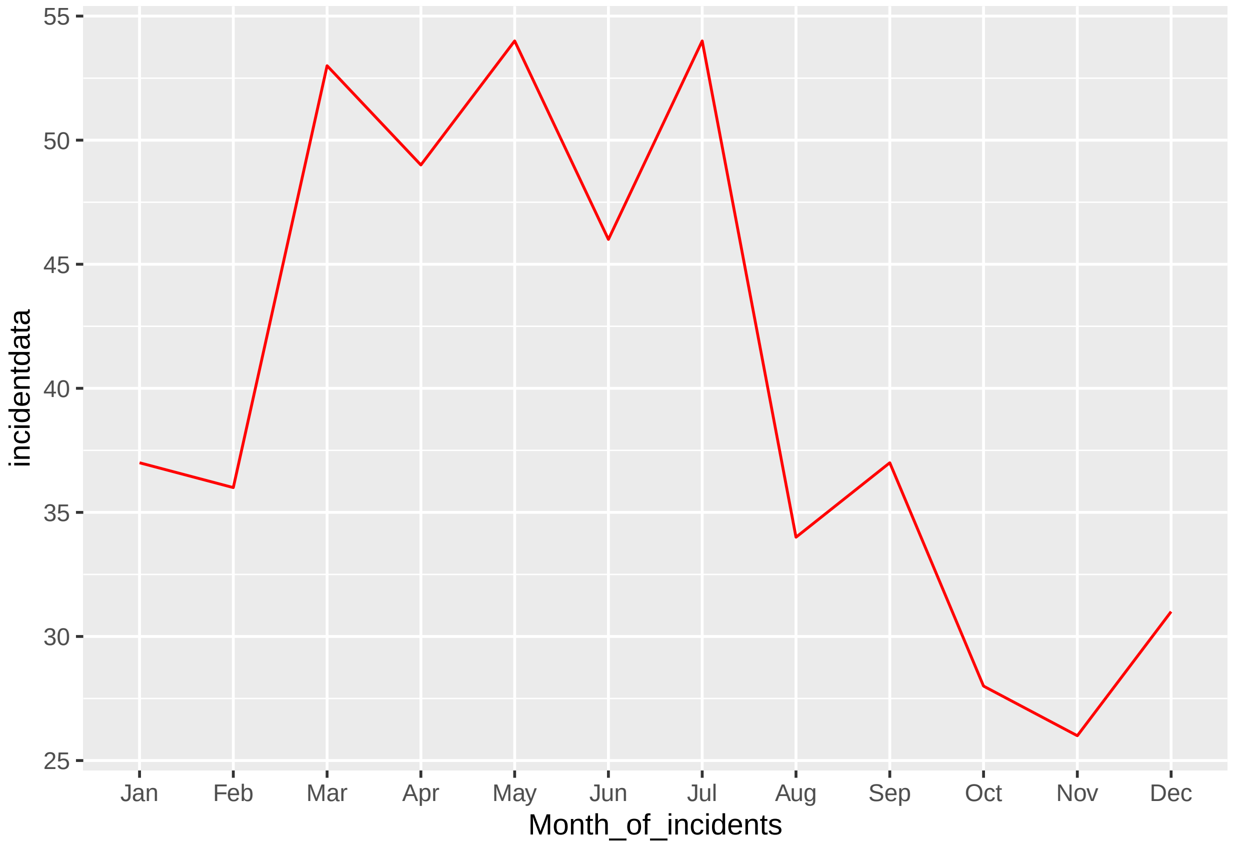
ANALYSING AND MODELING

Let us now analyze the data deeper to examine the rate at which the incidents increased or decreased over the past year for the age group that had the highest number of victims. Also, we can check if there is a huge difference between the months. By this, we can get which month has the highest incidents reported and which month has the least reported.

```
ny_incidents <- nypd_data %>%
filter(year(OCCUR_DATE)==2022 & PERP_AGE_GROUP=="25-44")

ny_incidents <- ny_incidents %>%
mutate(OCCUR_Month = month(OCCUR_DATE))

ny_incidents <- ny_incidents %>%
group_by(OCCUR_Month)%>%
summarise(incidentdata = n(), deaths = sum(STATISTICAL_MURDER_FLAG))

ny_incidents <- ny_incidents %>%
mutate(Month_of_incidents = month(OCCUR_Month,label = TRUE, abbr = TRUE))

ny_incidents
```

```
## # A tibble: 12 x 4
##    OCCUR_Month incidentdata deaths Month_of_incidents
##          <dbl>        <int>  <int> <ord>
##  1           1           37      8 Jan
##  2           2           36     14 Feb
##  3           3           53     11 Mar
##  4           4           49      4 Apr
##  5           5           54     22 May
##  6           6           46      5 Jun
##  7           7           54     15 Jul
##  8           8           34      5 Aug
##  9           9           37      3 Sep
## 10          10           28      8 Oct
## 11          11           26      5 Nov
## 12          12           31     10 Dec
```

```r
ny_incidents %>% ggplot(aes(x=Month_of_incidents, y=incidentdata, group = 1))+geom_line(color="red")
```



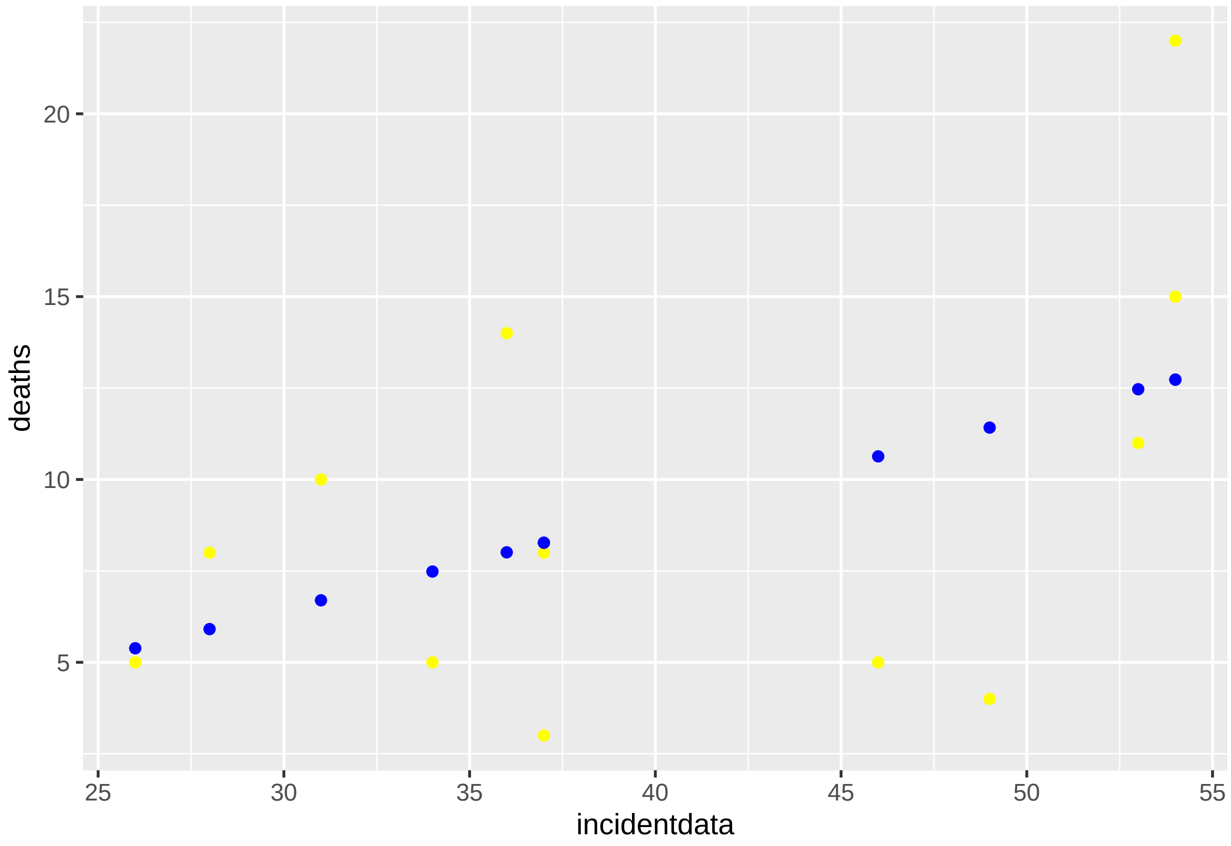
From the above graph, we could see there was an increase in crime rates in the months of May and July.

Linear regression is a fundamental statistical technique used for modeling the relationship between a dependent variable (response) and one or more independent variables (predictors). Likewise, we can show the **rate of incidents with respect to death rate**

```r
mod <- lm(deaths ~ incidentdata, data = ny_incidents)

summary(mod)
```

```
##
## Call:
## lm(formula = deaths ~ incidentdata, data = ny_incidents)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4187 -3.1799 -0.3272  2.5281  9.2694
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.4375     6.2733  -0.229    0.823
## incidentdata   0.2624     0.1508   1.740    0.112
##
## Residual standard error: 5.151 on 10 degrees of freedom
## Multiple R-squared:  0.2324, Adjusted R-squared:  0.1556
## F-statistic: 3.027 on 1 and 10 DF,  p-value: 0.1125
```

```
inc_pred <- ny_incidents %>%
mutate(pred = predict(mod))

inc_pred %>% ggplot() + geom_point(aes(x=incidentdata, y=deaths),color="yellow") + geom_point(aes(x=inc
```



In the above linear model, we can see that as the incident rate increases, the death rate also increases.

Bias Identification and Conclusion

We discovered the number of shooting incidents in each borough in New York City. We discussed the trend of shooting incidents over the past 16 years. The data shown here is likely only data documented by police reports. It is possible that shootings occur that go unreported. The potential sources of bias for datasets like this are numerous and often significant. In analyzing the NYPD shooting incident data, it is crucial to acknowledge the presence of missing values, as they inherently introduce uncertainty and potential bias in our analysis and subsequent models. The given data model and related plot are based on specific aspects of the NYPD data. There are certain limitations that prevent us from drawing useful conclusions from this analysis. A source of bias is that there could potentially be unreported shooting incidents that the police department has no information on. Possible sources of bias could be missing data and data noise.There is a significant challenge when it comes to analyzing or modeling perpetrator-related attributes due to the substantial amount of missing data.

Personal bias

These data contain age, sex, and race data in the PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, and VIC_RACE columns. Some records were based on incomplete reports where this information was listed as “unknown.”. While the values of these columns weren’t factored into the analysis, records based on incomplete reports were removed.

Also, these data contain a raw number of reported cases and do not consider population density, so violence per capita data could yield different results.