NYC Shooting Incident Analysis

KK

2024-06-08

```r
# Clear environment
rm(list = ls())
cat("\014")  # ctrl+L
```

```r
# import Libraries
library(tidyverse)
library(ggplot2)
library("cowplot")
library(skimr)
library(lubridate)
library(plotly)
library(forecast)
library(tseries)
```

Data:

The data description for this project is detailed on this website. https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Year-To-Date-/5ucz-vwe8/about_data

```r
# Load the dataset
df <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD",show_col
# Print the number of rows and columns

skim(df)
```

Table 1: Data summary

Name	df
Number of rows	28562
Number of columns	21
Column type frequency:	
character	12
difftime	1
logical	1
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
OCCUR_DATE	0	1.00	10	10	0	6095	0
BORO	0	1.00	5	13	0	5	0
LOC_OF_OCCUR_DESC	25596	0.10	6	7	0	2	0
LOC_CLASSFCTN_DESC	25596	0.10	5	11	0	10	0
LOCATION_DESC	14977	0.48	3	25	0	40	0
PERP_AGE_GROUP	9344	0.67	3	7	0	11	0
PERP_SEX	9310	0.67	1	6	0	4	0
PERP_RACE	9310	0.67	5	30	0	8	0
VIC_AGE_GROUP	0	1.00	3	7	0	7	0
VIC_SEX	0	1.00	1	1	0	3	0
VIC_RACE	0	1.00	5	30	0	7	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Lon_Lat	59	1.00	24	45	0	13403	0

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
OCCUR_TIME	0	1	0 secs	86340 secs	54900 secs	1423

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
STATISTICAL_MURDER_FLAG	0	1	0.19	FAL: 23036, TRU: 5526

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INCIDENT_KEY	0	1	127405824.78	68043401.99	9953245.00	65439914.00	92711253.50	203131993.00	279758069.00	
PRECINCT	0	1	65.50	27.34	1.00	44.00	67.00	81.00	123.00	
JURISDICTION_CODE	2	1	0.32	0.73	0.00	0.00	0.00	0.00	2.00	
X_COORD_CD	0	1	1009424.37	18301.11	914928.06	1000068.09	1007772.44	1016807.06	1066815.38	
Y_COORD_CD	0	1	208380.08	31917.51	125756.72	182912.00	194901.39	239814.23	271127.69	
Latitude	59	1	40.74	0.09	40.51	40.67	40.70	40.82	40.91	
Longitude	59	1	-73.91	0.07	-74.25	-73.94	-73.92	-73.88	-73.70	

Convert column names to lowercase.

```r
# Assuming your data frame is named df
names(df) <- tolower(names(df))

# Verify the changes
print(names(df))
```

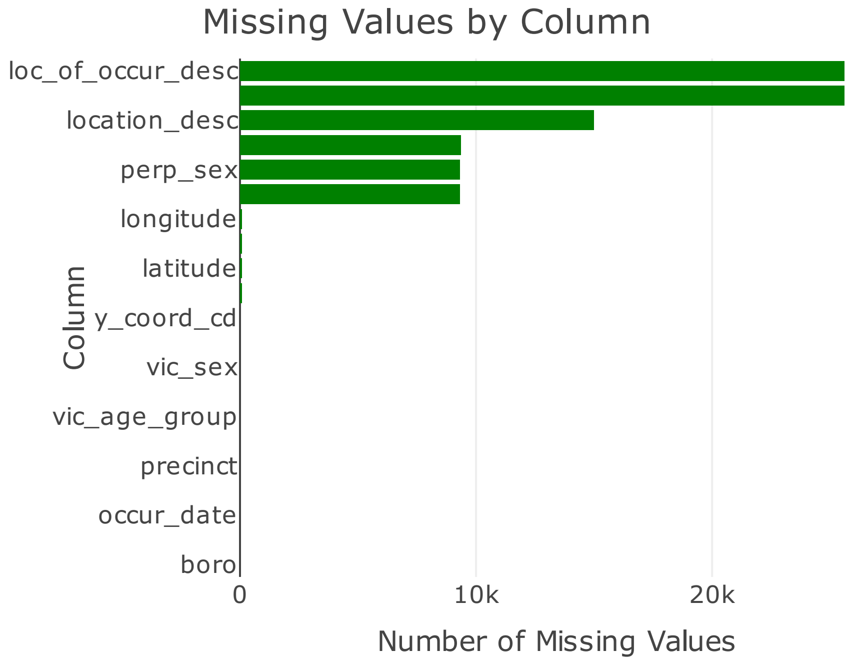
```
##  [1] "incident_key"            "occur_date"
##  [3] "occur_time"              "boro"
##  [5] "loc_of_occur_desc"       "precinct"
##  [7] "jurisdiction_code"       "loc_classfctn_desc"
##  [9] "location_desc"           "statistical_murder_flag"
## [11] "perp_age_group"          "perp_sex"
## [13] "perp_race"               "vic_age_group"
## [15] "vic_sex"                 "vic_race"
## [17] "x_coord_cd"              "y_coord_cd"
## [19] "latitude"                "longitude"
## [21] "lon_lat"
```

Exploratory Data Analysis

```
# Compute the count of missing values for each column
missing_counts <- colSums(is.na(df))

# Convert missing_counts to a dataframe
missing_df <- data.frame(Column = names(missing_counts), Missing_Count = missing_counts, row.names = NU

# Print the missing value counts dataframe
#print("Missing values for each column:")
#print(missing_df)

# Create a bar plot to visualize the missing values using Plotly
plot <- plot_ly(
  missing_df,
  x = ~Missing_Count,
  y = ~reorder(Column, Missing_Count),
  type = 'bar',
  orientation = 'h',
  marker = list(color = 'green')
) %>%
  layout(
    title = 'Missing Values by Column',
    xaxis = list(title = 'Number of Missing Values'),
    yaxis = list(title = 'Column'),
    plot_bgcolor = 'white',    # Set plot background to white
    paper_bgcolor = 'white'    # Set paper background to white
  )

# Display the plot
plot
```

Missing Values and How to Deal with Them

```
## Google Chrome was not found. Try setting the `CHROMOTE_CHROME` environment variable to the executabl
```



The dataset of NYC shooting incidents from 2006 to 2023 contains several columns with varying levels of missing values. Key columns like INCIDENT_KEY, OCCUR_DATE, OCCUR_TIME, BORO, PRECINCT, STATISTICAL_MURDER_FLAG, VIC_AGE_GROUP, VIC_SEX, VIC_RACE, X_COORD_CD, and Y_COORD_CD have complete data, making them reliable for analysis. However, columns such as LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, and PERP_RACE have substantial missing values, which may impact their analytical utility. Minimal missing data is found in JURISDICTION_CODE, Latitude, Longitude, and Lon_Lat, which can be handled through imputation or exclusion with negligible effect.

Remove Columns with More Than 50% Missing Data:

```
df <- df %>%
  select(
    -loc_of_occur_desc,
    -loc_classfctn_desc,
    -location_desc
  )
```

Replace ‘unknown’ and ‘null’ values with NA in the entire dataframe

```
df[df == "UNKNOWN" | df == "(null)"] <- NA
```

Remove NA values:

```r
# Remove NA values from the specified vectors
df <- df[complete.cases(df$vic_age_group, df$vic_sex, df$vic_race, df$latitude, df$longitude), ]

# View the structure of the cleaned data
#dim(df)
```

The levels ‘1020’,“1022”, ‘224’, and ‘940’ in the VIC_AGE_GROUP variable do not make sense, so we need to remove them:

```r
# Remove rows with specific levels in PERP_AGE_GROUP and drop unused levels
df <- df %>%
  filter(!(vic_age_group %in% c("1020", "224", "940","1022"))) %>%
  mutate(vic_age_group = fct_drop(vic_age_group))

# Check the levels of PERP_AGE_GROUP after removal
levels(df$vic_age_group)
```

```
## [1] "<18"   "18-24" "25-44" "45-64" "65+"
```

Descriptive Analysis

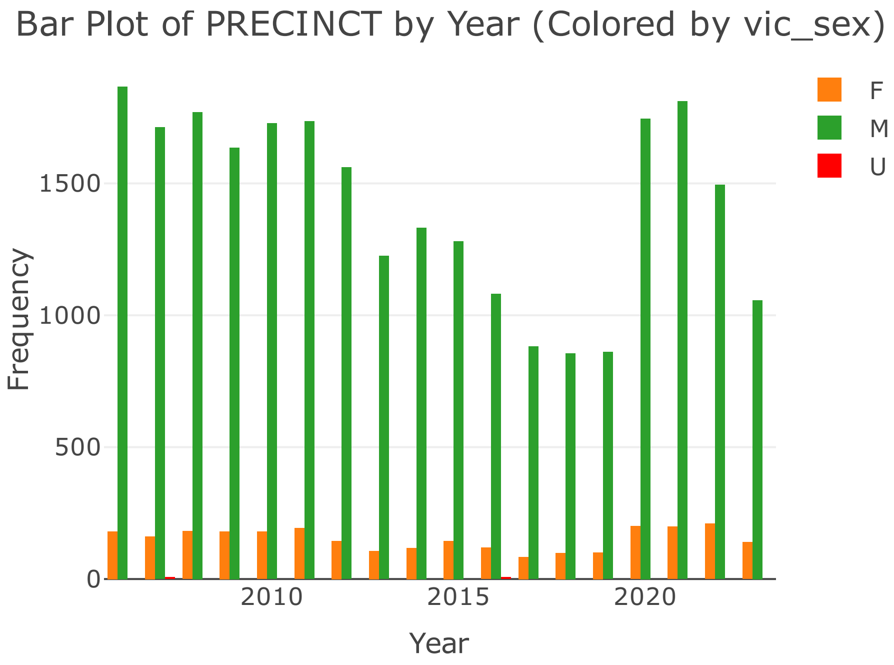
```r
# Convert OCCUR_DATE to Date type
df$occur_date <- as.Date(df$occur_date, format="%m/%d/%Y")

# Extract the year from OCCUR_DATE
df$Year <- year(df$occur_date)
```

Q1: What are the observed trends and gender disparities in NYC shooting incidents from 2006 to 2023?

```r
# Summarize the data by YEAR and VIC_SEX, counting the occurrences of PRECINCT
df_summary <- df %>%
  group_by(Year, vic_sex) %>%
  summarise(total_precincts = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

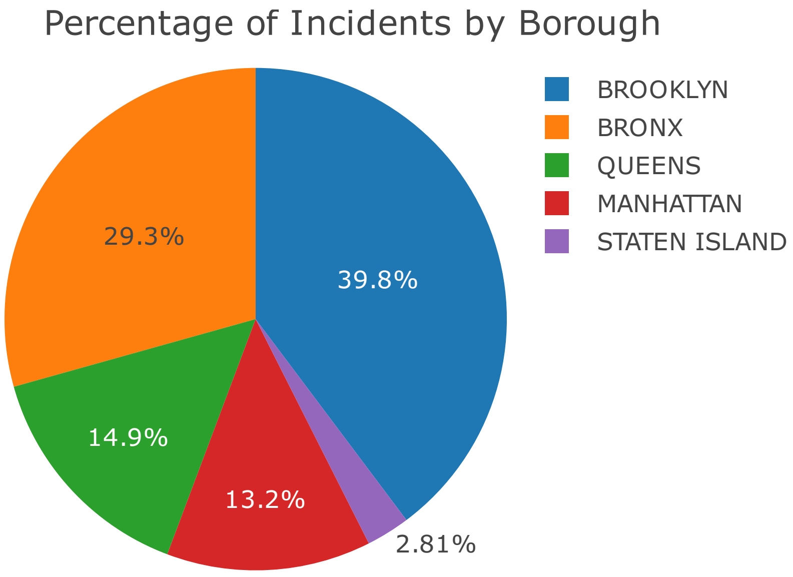
```r
# Display the summary dataframe
#print(df_summary)

# Define custom colors
custom_colors <- c('#ff7f0e', '#2ca02c', 'red')  # Orange, green, red

# Create the bar plot using Plotly
plot <- plot_ly(df_summary,
                x = ~Year,
                y = ~total_precincts,
                color = ~vic_sex,
                type = 'bar',
                hoverinfo = "y",
                colors = custom_colors) %>%
  layout(title = 'Bar Plot of PRECINCT by Year (Colored by vic_sex)',
         xaxis = list(title = 'Year'),
         yaxis = list(title = 'Frequency'),
         barmode = 'group',
         plot_bgcolor = 'white',     # Set plot background to white
         paper_bgcolor = 'white')    # Set paper background to white

# Display the plot
plot
```



The dataset on NYC shooting incidents from 2006 to 2023 shows that male victims consistently outnumber female and “U” victims each year. While the annual number of incidents varies, a significant peak is observed in 2006 with 1867 male and 181 female victims. After a general decline from 2007 to 2019, incidents sharply increase in 2020, with 1746 male and 201 female victims,and in 2021 before decreasing again in subsequent years, ending with 1057 male and 140 female victims in 2023. Incidents involving “U” victim sex are rare, recorded only in 2007 and 2016. This data highlights the persistent gender disparity and fluctuating trends in shooting incidents over the analyzed period.

Q2: What are the relative proportions of shooting incidents across different boroughs in New York City?

```r
# Summarize data by borough
borough_counts <- df %>%
  group_by(boro) %>%
  summarise(n = n())

# Calculate percentage
borough_counts <- borough_counts %>%
  mutate(percent = n / sum(n) * 100)

# Plotting the pie chart
plot_ly(borough_counts, labels = ~boro, values = ~percent, type = 'pie') %>%
  layout(title = "Percentage of Incidents by Borough",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = TRUE)
```

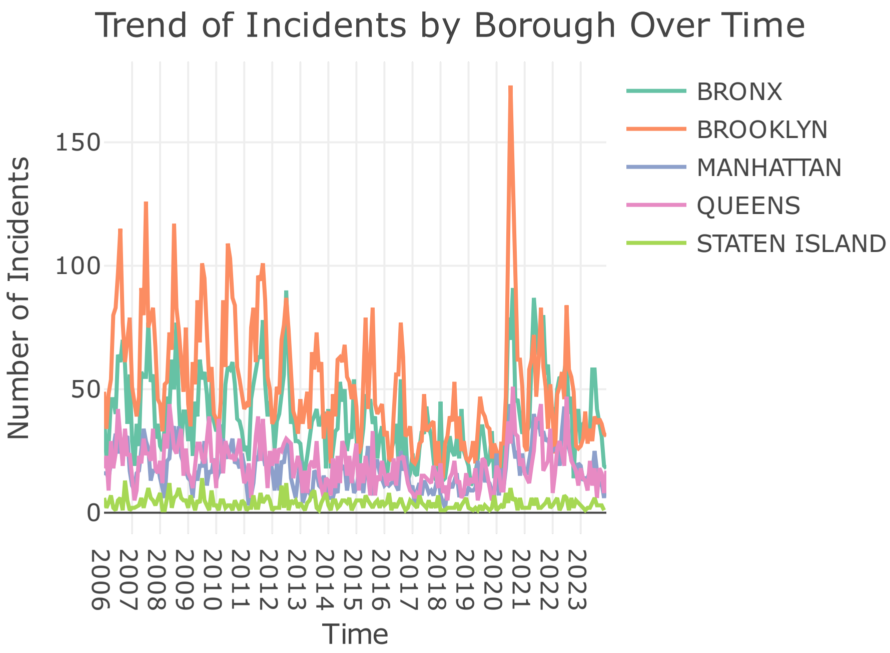


The breakdown of shooting incidents in New York City by borough reveals that Brooklyn has the highest incidence rate, followed by the Bronx. Manhattan, Queens, and Staten Island have comparatively lower rates of incidents.

Q3 : How do the trends in incidents vary across different boroughs of New York City over time?

```r
# Extract Year-Month from OCCUR_DATE
df$YearMonth <- format(df$occur_date, "%Y-%m")
```

```
# Aggregate data by YearMonth and BORO
incidents_by_time <- df %>%
  group_by(YearMonth, boro) %>%
  summarise(n = n(), .groups = "drop")  # Specify `.groups = "drop"` to override the grouped output

# Convert YearMonth to Date type for plotting
incidents_by_time$Date <- as.Date(paste0(incidents_by_time$YearMonth, "-01"))

# Plotting the count by time in each borough with explicit date breaks
plot_ly(incidents_by_time, x = ~Date, y = ~n, color = ~boro, type = 'scatter', mode = 'lines') %>%
  layout(title = "Trend of Incidents by Borough Over Time",
         xaxis = list(title = "Time", type = "date", tickmode = "linear", tick0 = "2005-01-01", dtick =
         yaxis = list(title = "Number of Incidents"),
         legend = list(title = ""),
         showlegend = TRUE)
```



By examining the plot of the trend of incidents by borough over time, we notice that Brooklyn has the highest number of incidents over time, followed by the Bronx, Queens, and Manhattan. Staten Island has the fewest incidents over time.

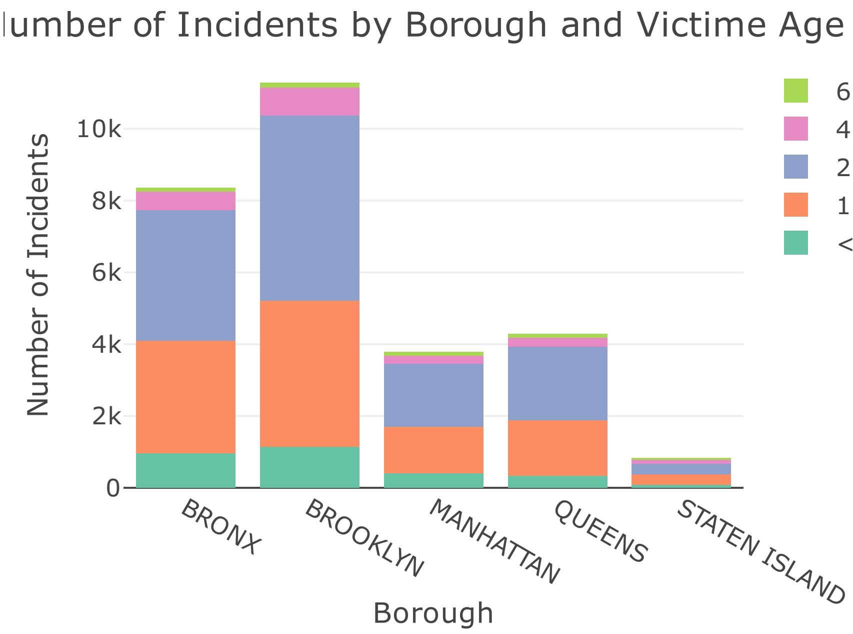
Q4: What are the patterns and disparities in shooting incidents across different age groups in New York City boroughs?

```
# Aggregate data by Borough and VIC_AGE_GROUP
# Display the plot
```

```
borough_age_counts <- df %>%
  group_by(boro, vic_age_group) %>%
  summarise(n = n())
```

```
## `summarise()` has grouped output by 'boro'. You can override using the
## `.groups` argument.
```

```
# Create the plotly plot
plot <- plot_ly(borough_age_counts, x = ~boro, y = ~n, color = ~vic_age_group, type = "bar") %>%
  layout(title = "Number of Incidents by Borough and Victime Age Group",
         xaxis = list(title = "Borough"),
         yaxis = list(title = "Number of Incidents"),
         barmode = "stack")

# Display the plot
plot
```



The data presents a breakdown of shooting incidents across various age groups in each borough of New York City. It reveals the number of incidents involving victims under 18, aged 18-24, 25-44, 45-64, and 65+, in the Bronx, Brooklyn, Manhattan, Queens, and Staten Island. For instance, Brooklyn recorded the highest number of incidents across all age groups, followed by the Bronx, Queens, Manhattan, and Staten Island. This summary offers valuable insights into the distribution of incidents by age group within each borough, aiding in the understanding of trends and patterns in shooting incidents across New York City.

Modeling

Two Way Anova Model

Model with Interaction:

```
model_interaction <- aov(precinct ~ boro + vic_age_group + boro * vic_age_group, data = df)
summary(model_interaction)
```

```
##                         Df   Sum Sq Mean Sq   F value   Pr(>F)    
## boro                     4 20205304 5051326 1.449e+05  < 2e-16 ***
## vic_age_group            4      226      56 1.620e+00  0.16611    
## boro:vic_age_group      16     1276      80 2.288e+00  0.00239 ** 
## Residuals            28362   988681      35                       
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpration The two-way ANOVA analysis indicates that boro has a highly significant effect on precinct (p < 2.2e-16), while vic_age_group does not show a significant main effect (p = 0.166). However, there is a significant interaction between boro and vic_age_group (p = 0.002388), suggesting that the impact of vic_age_group on precinct varies depending on the boro. This implies that precinct values differ substantially across different boro levels, and the influence of vic_age_group on precinct is contingent on the specific boro.

Model without Interation:

```
model_without_interaction <- aov(precinct ~ boro + vic_age_group, data = df)
summary(model_without_interaction)
```

```
##                  Df   Sum Sq Mean Sq   F value Pr(>F)    
## boro              4 20205304 5051326 1.448e+05 <2e-16 ***
## vic_age_group     4      226      56 1.619e+00  0.166    
## Residuals     28378   989958      35                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The best model:

```
anova(model_interaction, model_without_interaction)
```

```
## Analysis of Variance Table
## 
## Model 1: precinct ~ boro + vic_age_group + boro * vic_age_group
## Model 2: precinct ~ boro + vic_age_group
##   Res.Df    RSS  Df Sum of Sq      F   Pr(>F)   
## 1  28362 988681                                 
## 2  28378 989958 -16   -1276.3 2.2883 0.002388 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA comparison between two models shows that including the interaction term between boro and vic_age_group significantly improves the model fit for predicting precinct values. The model with the interaction term has a lower residual sum of squares and an F value of 2.2883, with a p-value of 0.002388, indicating that the interaction effect is statistically significant. Therefore, the effect of vic_age_group on precinct varies by boro, justifying the inclusion of the interaction term in the model.

Compare models using MSE:

```r
# Fit the model with only the boro variable
model_boro <- lm(precinct ~ boro, data = df)

# Fit the model with boro and the interaction with vic_age_group
model_boro_interaction <- lm(precinct ~ boro + vic_age_group + boro * vic_age_group, data = df)

# Make predictions using the model with only boro
df$predicted_precinct_boro <- predict(model_boro, newdata = df)

# Make predictions using the model with boro and the interaction with vic_age_group
df$predicted_precinct_boro_interaction <- predict(model_boro_interaction, newdata = df)

# Calculate residuals for the model with only boro
df$residuals_boro <- df$precinct - df$predicted_precinct_boro

# Calculate residuals for the model with boro and the interaction with vic_age_group
df$residuals_boro_interaction <- df$precinct - df$predicted_precinct_boro_interaction

# Calculate the mean squared error (MSE) for the model with only boro
mse_boro <- mean(df$residuals_boro^2)

# Calculate the mean squared error (MSE) for the model with boro and the interaction with vic_age_group
mse_boro_interaction <- mean(df$residuals_boro_interaction^2)

# Print the MSEs
print(paste("MSE for the model with only boro: ", mse_boro))
```

```
## [1] "MSE for the model with only boro:  34.8815793189759"
```

```r
print(paste("MSE for the model with boro and interaction with vic_age_group: ", mse_boro_interaction))
```

```
## [1] "MSE for the model with boro and interaction with vic_age_group:  34.8286613580277"
```

```r
# Create a dataframe containing actual, predicted, date, and boro information

# Extract Year-Month from OCCUR_DATE
df$YearMonth <- format(df$occur_date, "%Y-%m")
actual_vs_predicted <- data.frame(
    boro = df$boro,
    df$YearMonth,
    actual_precinct = df$precinct,
    predicted_precinct = df$predicted_precinct_boro)
head(actual_vs_predicted, 5)
```

```
##        boro df.YearMonth actual_precinct predicted_precinct
## 1 MANHATTAN      2022-05              14           25.31255
## 2     BRONX      2022-07              48           44.97899
## 3    QUEENS      2012-05             103          107.20825
## 4     BRONX      2019-09              42           44.97899
## 5  BROOKLYN      2007-02              83           74.34314
```

Step 4 : Conclusion and Bias

This analysis of NYC shooting incidents uncovers significant spatial and demographic patterns, emphasizing the influence of boroughs and victim age groups on gun violence. Key findings show that specific boroughs and age groups are more vulnerable to incidents, with a significant interaction between these factors. However, potential biases include data collection inconsistencies, temporal limitations, and geographical disparities. My personal bias might stem from preconceived notions about certain areas of NYC or specific demographic groups based on societal stereotypes or media portrayals. These biases were addressed through a data-driven approach, peer review, and transparent methodology. Despite these limitations, the insights gained can guide targeted interventions and policy decisions to enhance public safety and reduce gun violence in NYC.