NYPD Shooting Incident Analysis

2026-01-21

Introduction

This R-Markdown file looks at the NYPD Shooting Incident dataset for incident patterns. As requested in the assignment, I examined the bias of proximity bias. Proximity bias refers to the tendency for crimes to occur between individuals of similar demographics or within certain geographic areas. Admittedly, I had heard from podcasts that this may be a contributing factor and I wanted to examine if my prior bias could be seen in the data, thereby checking my own potential biases towards the data in the process.

Data Import and Cleaning

```
# Load libraries
library(tidyverse)
library(lubridate)
library(ggplot2)

# Read data directly from the NYC Open Data URL
url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
df <- read.csv(url)

# Date and time cleaning
df <- df %>%
  mutate(
    OCCUR_DATE = mdy(OCCUR_DATE),
    year = year(OCCUR_DATE),
    month = month(OCCUR_DATE, label = TRUE),
    hour = as.numeric(substr(OCCUR_TIME, 1, 2))
  ) %>%
  filter(!is.na(OCCUR_DATE)) %>%
  filter(!is.na(BORO) & BORO != "")
```

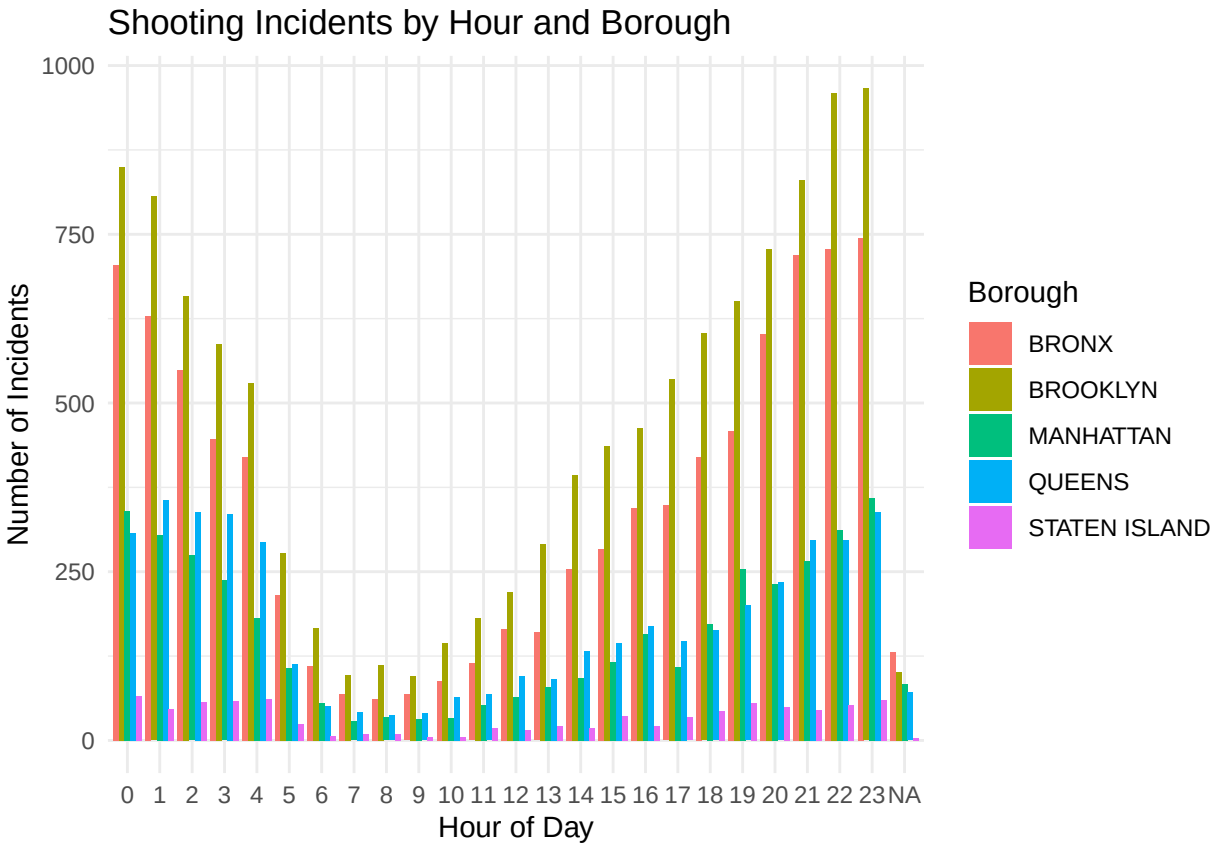
Shooting Incidents by Hour and Borough

```
# Create a new data frame so that we can group without disturbing our original
      # df
shootings_hour_borough <- df %>%
  group_by(hour, BORO) %>%
  summarize(count = n(), .groups = "drop")

# Create a plot for shootings by hour and borough
plot_hour_borough <- ggplot(shootings_hour_borough, aes(x = factor(hour), y = count, fill = BORO)) +
```

```
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Shooting Incidents by Hour and Borough",
       x = "Hour of Day",
       y = "Number of Incidents",
       fill = "Borough") +
  theme_minimal()

print(plot_hour_borough)
```

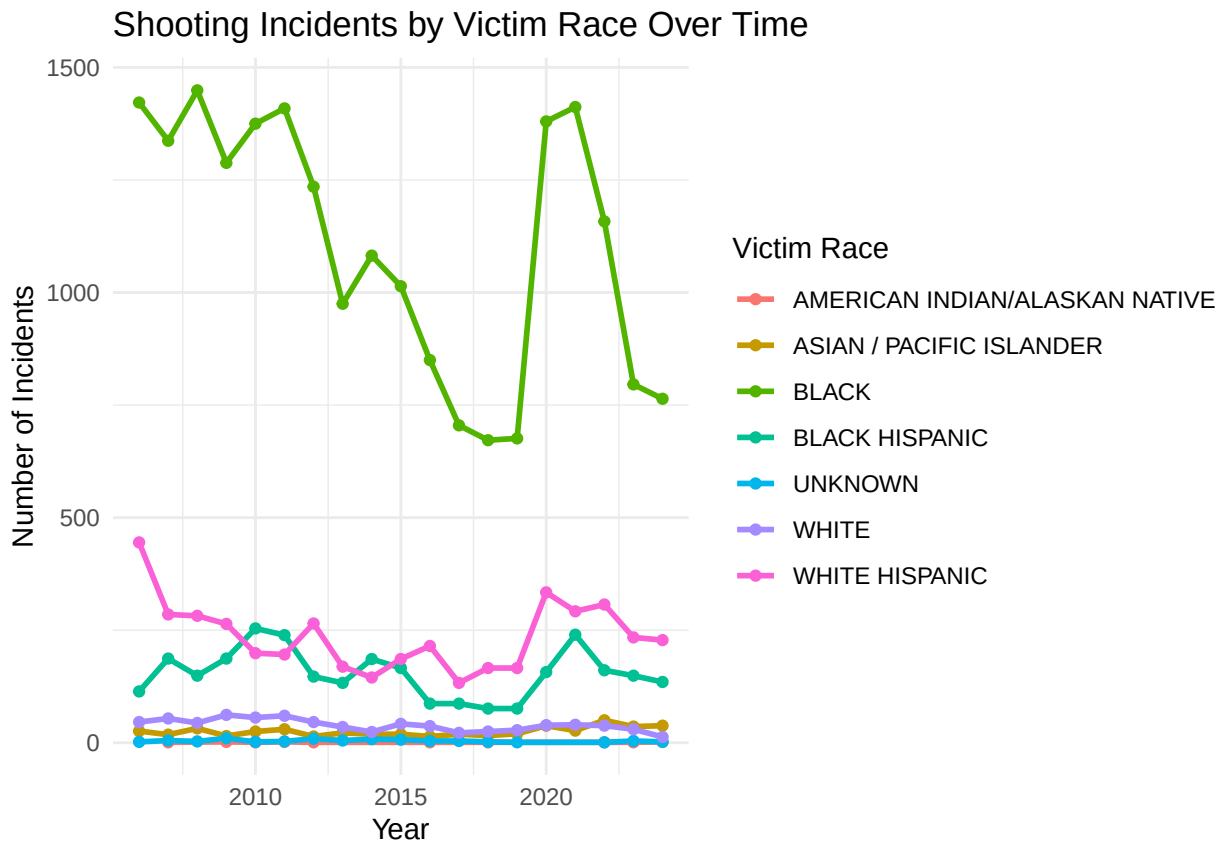


Shooting Incidents by Victim Race Over Time

```
# Similar mutations as before, but this time we will group by victim race
shootings_race_year <- df %>%
  filter(!is.na(VIC_RACE) & VIC_RACE != "") %>%
  group_by(year, VIC_RACE) %>%
  summarize(count = n(), .groups = "drop")

# Create a plot of incidents by victim race over time
plot_race_year <- ggplot(shootings_race_year, aes(x = year, y = count, color = VIC_RACE)) +
  geom_line(size = 1) +
  geom_point() +
  labs(title = "Shooting Incidents by Victim Race Over Time",
       x = "Year",
       y = "Number of Incidents",
       color = "Victim Race") +
  theme_minimal()
```

```
print(plot_race_year)
```



Percentage of Victim Race by Perpetrator Race

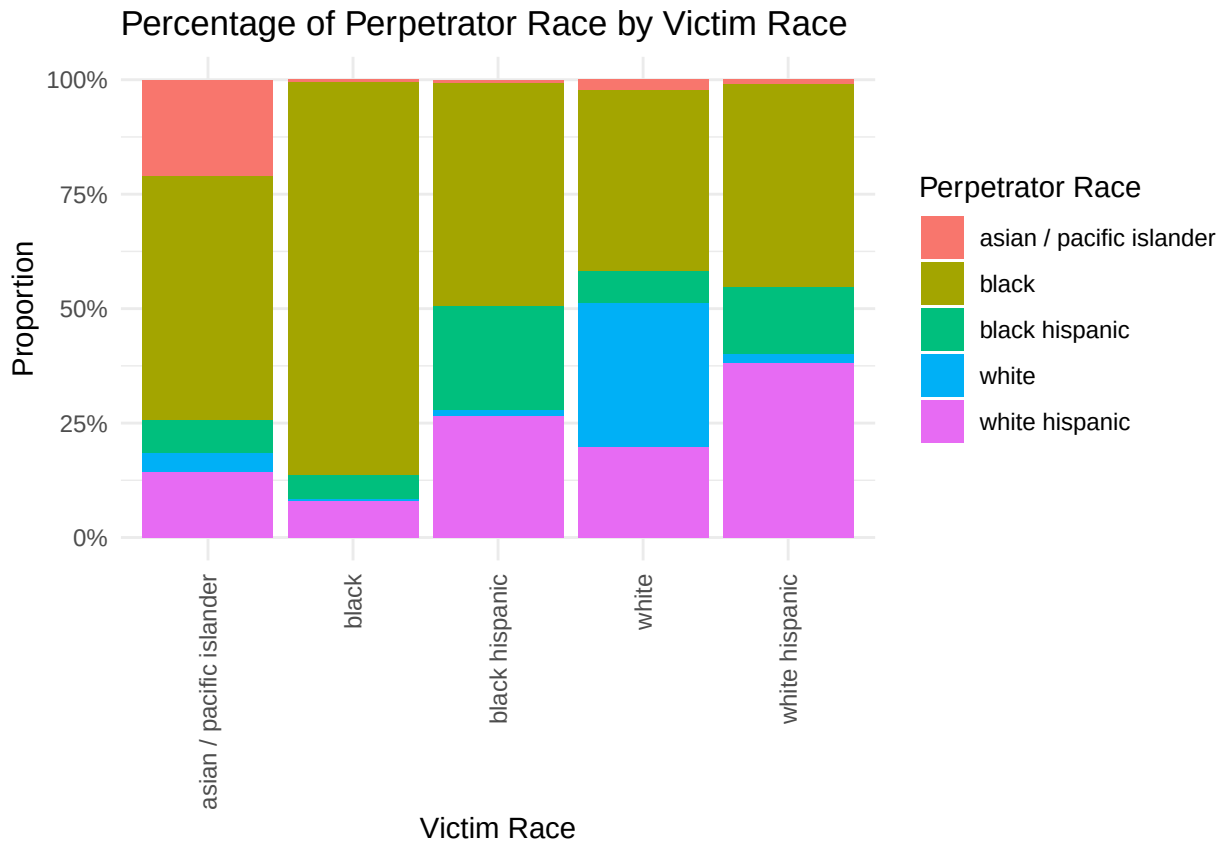
```
# Convert race columns to character, trim whitespace, and convert to lower case.
df_vic <- df %>%
  mutate(PERP_RACE = as.character(PERP_RACE),
         VIC_RACE = as.character(VIC_RACE)) %>%
  mutate(PERP_RACE = str_to_lower(str_trim(PERP_RACE)),
         VIC_RACE = str_to_lower(str_trim(VIC_RACE))) %>%
  mutate(PERP_RACE = na_if(PERP_RACE, ""),
         VIC_RACE = na_if(VIC_RACE, ""))

# Filter the data to drop rows with unwanted values
df_vic <- df_vic %>%
  filter(!is.na(PERP_RACE) & !is.na(VIC_RACE)) %>%
  filter(!(PERP_RACE %in% c("(null)", "american indian/alaskan native", "unknown")),
         !(VIC_RACE %in% c("(null)", "american indian/alaskan native", "unknown"))) %>%
  mutate(PERP_RACE = droplevels(as.factor(PERP_RACE)),
         VIC_RACE = droplevels(as.factor(VIC_RACE)))

# Create plot of percent perpetrator race by victim race to see if there is an
    # Outsized effect for given race by that race.
plot_stack <- ggplot(df_vic, aes(x = VIC_RACE, fill = PERP_RACE)) +
  geom_bar(position = "fill") +
  labs(title = "Percentage of Perpetrator Race by Victim Race",
```

```
       x = "Victim Race",
       y = "Proportion",
       fill = "Perpetrator Race") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))

print(plot_stack)
```



Linear Model: Fatalities vs. Shootings

```
# Now we are going to create a linear model but we have to clean the data up first.
      # We need to drop rows with missing/empty OCCUR_TIME
df_lm <- df %>%
  filter(!is.na(OCCUR_TIME) & OCCUR_TIME != "") %>%
  mutate(OCCUR_DATE = as.Date(OCCUR_DATE, format = "%m/%d/%Y"),
         year = year(OCCUR_DATE),
         month = month(OCCUR_DATE, label = TRUE),
         hour = as.numeric(substr(OCCUR_TIME, 1, 2)),
         fatality_flag = ifelse(tolower(as.character(STATISTICAL_MURDER_FLAG)) == "true", 1, 0))

# Aggregate by year & month
monthly_data <- df_lm %>%
  group_by(year, month) %>%
  summarise(shootings = n(),
            fatalities = sum(fatality_flag, na.rm = TRUE),
```

```
            .groups = "drop")

print(monthly_data)
```

```
## # A tibble: 228 x 4
##     year month shootings fatalities
##    <dbl> <ord>     <int>      <dbl>
##  1  2006 Jan         129         29
##  2  2006 Feb          97         27
##  3  2006 Mar         102         14
##  4  2006 Apr         156         37
##  5  2006 May         173         40
##  6  2006 Jun         180         36
##  7  2006 Jul         233         47
##  8  2006 Aug         245         46
##  9  2006 Sep         196         44
## 10  2006 Oct         199         38
## # i 218 more rows
```

```
# Fit a simple lm: fatalities ~ shootings
model_lm_simple <- lm(fatalities ~ shootings, data = monthly_data)
summary(model_lm_simple)
```

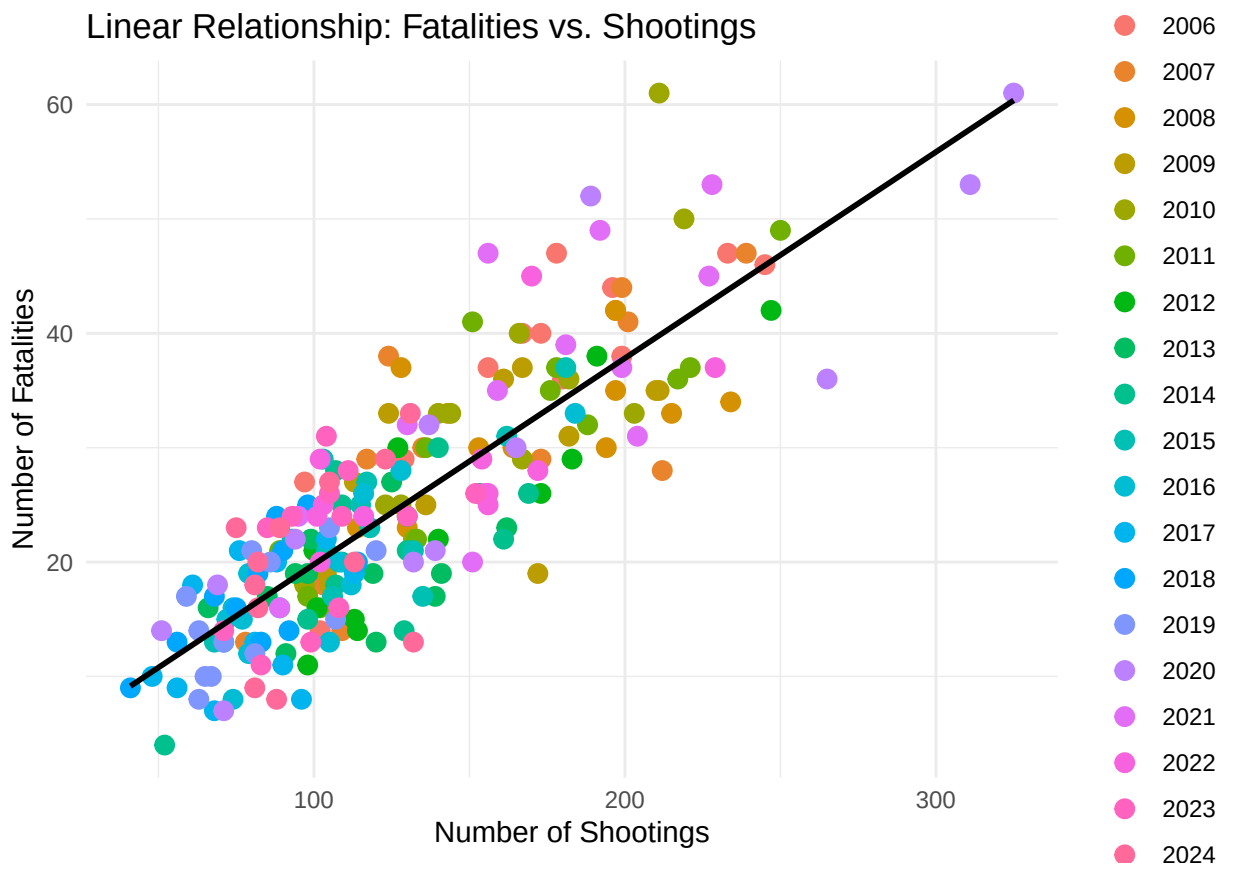
```
##
## Call:
## lm(formula = fatalities ~ shootings, data = monthly_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7756  -4.2070  -0.1217   3.9161  21.1925
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.763234   1.076856   1.637    0.103
## shootings   0.180305   0.007689  23.450   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.915 on 226 degrees of freedom
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.7074
## F-statistic: 549.9 on 1 and 226 DF,  p-value: < 2.2e-16
```

```
# Predicted fatalities from the simple model
monthly_data <- monthly_data %>%
  mutate(predicted_fatalities = predict(model_lm_simple, newdata = .))

# Display plot with regression line
plot_model <- ggplot(monthly_data, aes(x = shootings, y = fatalities)) +
  geom_point(aes(color = as.factor(year)), size = 3) +
  geom_smooth(method = "lm", se = FALSE, color = "black", size = 1) +
  labs(title = "Linear Relationship: Fatalities vs. Shootings",
       x = "Number of Shootings",
```

```
       y = "Number of Fatalities",
       color = "Year") +
  theme_minimal()

print(plot_model)
```



References:

Assistance provided by Claude 4.5 Sonnet for generating code for charts, troubleshooting R Markdown knitting errors and fixing dataset connection issues. January 14, 2026.