NYPD Shooting Incidents Report

Parnia Shahpari

2024-08-20

This dataset, published by NYC Open Data, covers all NYC shooting incidents from 2006 through the end of the last calendar year. Each entry includes details on the event, such as location, time, and demographics of both suspects and victims. Dataset can be found at https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8/about_data

The following are the libraries used in my analysis:

- Data manipulation and cleaning (dplyr, tidyr, data.table)
- Data visualization (ggplot2, ggmap)
- Modeling (mgcv, nlme)
- Time series analysis (zoo, lubridate)
- Working with different data formats (readr, jsonlite)

NOTE: A personal API key from Stadia Maps is required to visualize the spatial distribution of incidents in NYC

```
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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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(lubridate)
library(stringr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# Identify the data source
shooting_data_url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"

# Read in CSV file
shooting_data <- read_csv(shooting_data_url)
```

```
## Rows: 28562 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Remove columns that are mostly empty or redundant
shooting_data <- shooting_data %>%
    select(-c(LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION_DESC, X_COORD_CD, Y_COORD_CD))

# Convert data types
shooting_data <- shooting_data %>%
mutate(OCCUR_DATE = mdy(OCCUR_DATE),
OCCUR_TIME = hms::as_hms(OCCUR_TIME),
across(c(BORO, STATISTICAL_MURDER_FLAG, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VI

# Group shootings by borough and year for further analysis
shooting_data <- shooting_data %>% mutate(YEAR = year(OCCUR_DATE))

shooting_by_year <- shooting_data %>%
    group_by(BORO, YEAR) %>%
    summarize(INCIDENTS = n()) %>%
    ungroup()
```

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

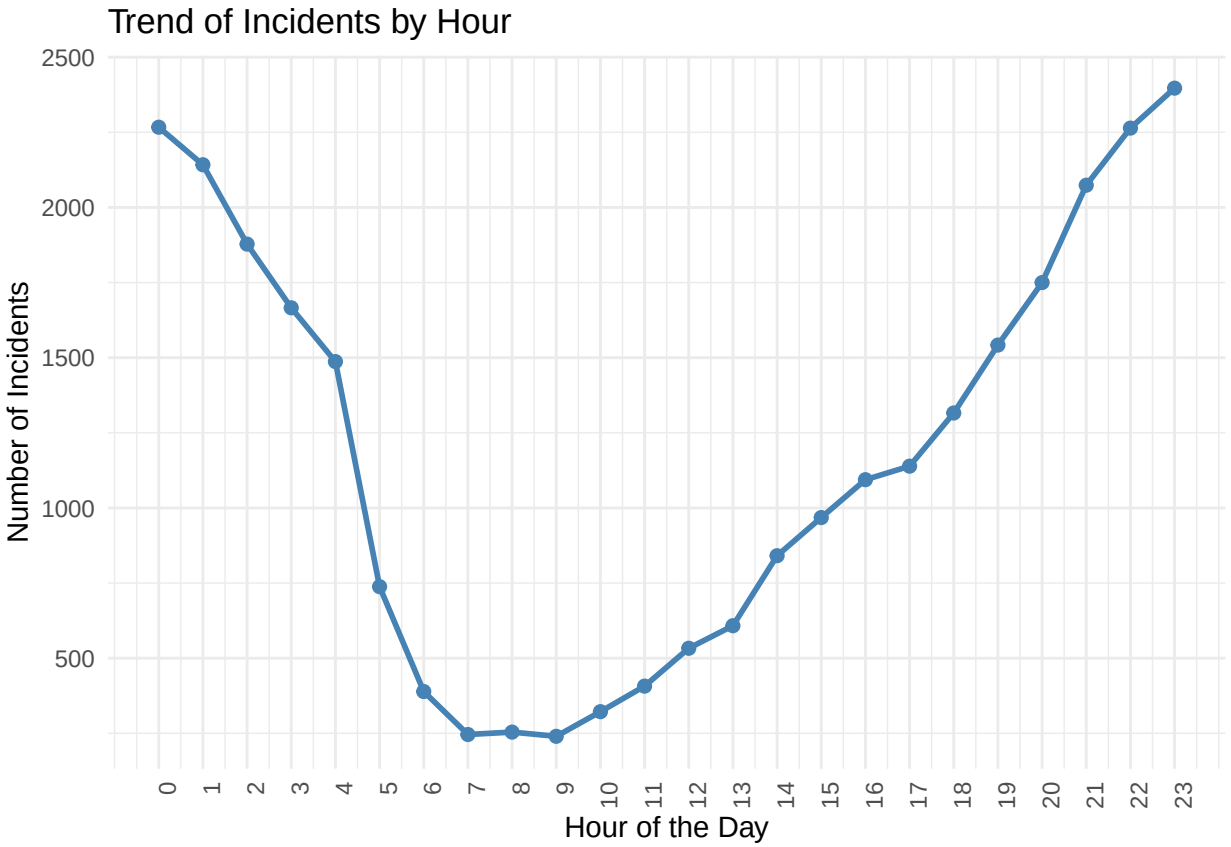
```r
# Identify the data source
population_data_url <- "https://data.cityofnewyork.us/api/views/xywu-7bv9/rows.csv?accessType=DOWNLOAD"

# Read in CSV file
population_data <- read_csv(population_data_url)
```

```
## Rows: 6 Columns: 22
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Age Group, Borough
## dbl (20): 1950, 1950 - Boro share of NYC total, 1960, 1960 - Boro share of N...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Remove unnecessary columns and rows, rename Borough to match naming in shooting incident data, format
population_data <- population_data %>%
filter(Borough != "NYC Total") %>%
select(-contains("Boro share of NYC total"), -"Age Group") %>%
pivot_longer(cols = -Borough, names_to = "Year", values_to = "Population") %>%
mutate(Year = as.integer(Year), Population = as.integer(Population), Borough = str_to_upper(Borough))

# Filter for the years 2000 to 2023, and interpolate the population size for each borough
pop_intrpl_by_bor<- population_data %>%
    filter(Year >=2000 & Year <= 2030) %>%
    group_by(Borough) %>%
    complete(Year = seq(min(Year), max(Year), by = 1)) %>%
    arrange(Borough, Year) %>%
    mutate(Population = na.approx(Population, na.rm = FALSE)) %>%
    ungroup()

# Merge shooting incidents data with population data and add an incidents per thousand column
shooting_by_year <- shooting_by_year %>%
    left_join(pop_intrpl_by_bor, by = c("BORO" = "Borough", "YEAR" = "Year")) %>%
    rename(POPULATION = "Population") %>%
    mutate(INCIDENTS_PER_THOU = INCIDENTS * 1000 / POPULATION)
```

A line plot illustrating the distribution of incidents across a 24-hour period, aggregated over all days in the dataset.

```r
library(readr)
library(dplyr)
library(lubridate)
library(ggplot2)

#  Tidy and transform the data. Extract the hour from the OCCUR_TIME column and count the number of inc
shooting_data <- shooting_data %>% mutate(HOUR = hour(OCCUR_TIME))
shootings_hourly <- shooting_data %>%
  group_by(HOUR) %>%
  summarize(INCIDENT_COUNT = n())

# Create a line plot to visualize the occurrence of incidents over 24 hours
shootings_hourly %>% ggplot(aes(x = HOUR, y = INCIDENT_COUNT)) +
  geom_line(color = "steelblue", linewidth = 1) +
  geom_point(color = "steelblue", size = 2) +
  scale_x_continuous(breaks = seq(0, 23, by = 1),
                     labels = seq(0, 23, by = 1),
                     limits = c(0, 23)) +
  labs(title = "Trend of Incidents by Hour",
       x = "Hour of the Day",
       y = "Number of Incidents") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Key Observations from the Plot

- There's a pattern in the occurrence of shooting incidents throughout the day.
- The number of incidents is lowest in the early morning hours (around 7-9 AM).
- The incidents start to increase from the late morning and continue to rise throughout the day.
- The peak of incidents occurs late at night, with the highest number of shootings happening around 11 PM to midnight.
- There's a sharp decline in incidents in the early morning hours (2-7 AM).

A scatter plot of all of the incidents which occurred, color-coded by borough, overlaying a map of New York

```
library(ggmap)
```

```
## i Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```
# Access and Register Stadia Maps API key
stadia_api_key <- Sys.getenv("STADIA_API_KEY")
register_stadiamaps(key = stadia_api_key)
```

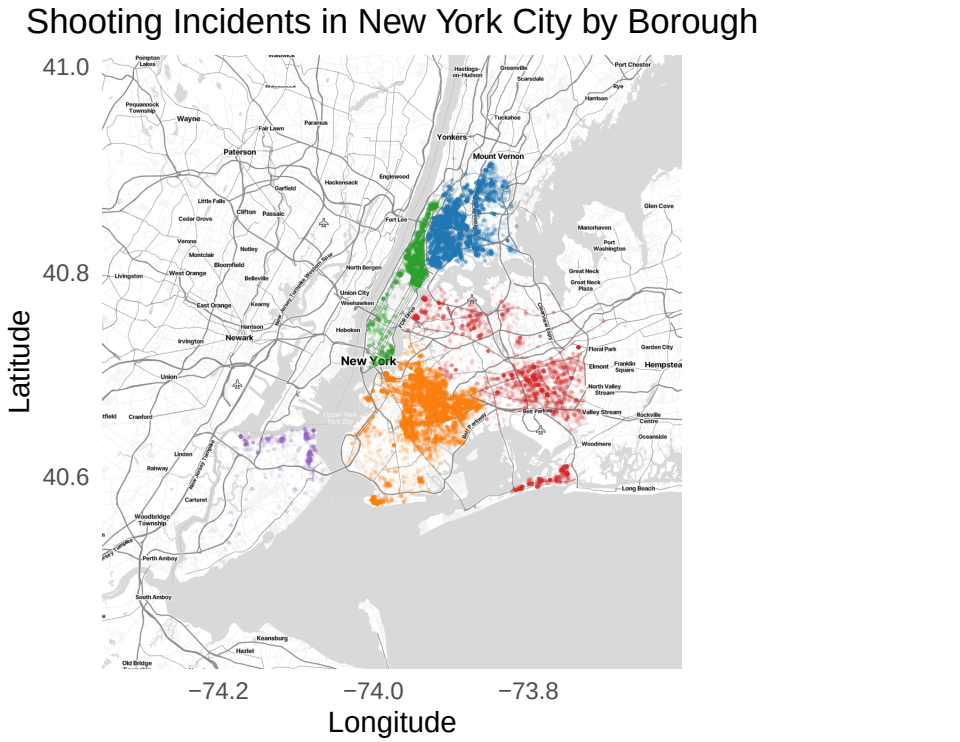
```r
# Tidy and transform the data; remove rows with missing latitude or longitude
shooting_lat_long <- shooting_data %>%
    filter(!is.na(Latitude) & !is.na(Longitude))

# Create a color palette
borough_colors <- c("BRONX" = "#1f77b4", "BROOKLYN" = "#ff7f0e", "MANHATTAN" = "#2ca02c", "QUEENS" = "#

# Define the bounding box
bbox <- c(
  left = min(shooting_lat_long$Longitude) - 0.1,
  bottom = min(shooting_lat_long$Latitude) - 0.1,
  right = max(shooting_lat_long$Longitude) + 0.1,
  top = max(shooting_lat_long$Latitude) + 0.1
)
print(tempdir())  # See if tempdir() returns a valid, writable directory
```

```
## [1] "/var/folders/90/rgvcgnfn2lsbc4fprz790z8c0000gn/T//RtmptcOZbH"
```

```r
# Get the map using get_stadiamap
nyc_map <- get_stadiamap(
  bbox = bbox,
  zoom = 11,
  maptype = "stamen_toner_lite",
  crop = TRUE,
  messaging = FALSE,
  urlonly = FALSE,
  color = "color",
  force = FALSE,
  where = tempdir()
)
```

```
## i © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

```r
# Create a scatter plot overlaying map of NYC
shooting_lat_long_map <- ggmap(nyc_map) +
  geom_point(data = shooting_lat_long, aes(x = Longitude, y = Latitude, color = BORO), alpha = 0.1, size
  scale_color_manual(values = borough_colors) +
  theme_minimal() +
  labs(title = "Shooting Incidents in New York City by Borough",
       x = "Longitude",
       y = "Latitude",
       color = "Borough") +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "bottom") +
    guides(color = guide_legend(override.aes = list(size = 1.5, alpha = 1)))
print(shooting_lat_long_map)
```

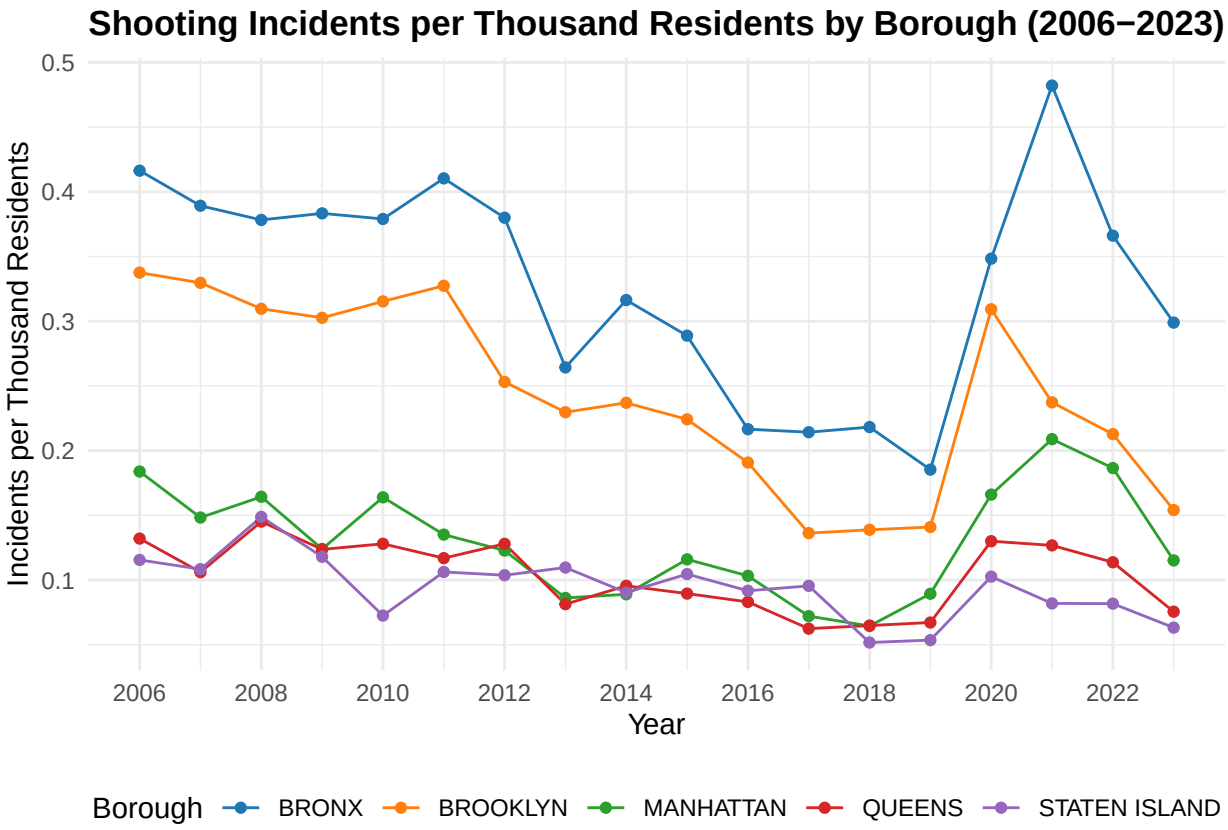


Key Observations from the Map

- This map provides a geographical visualization of the distribution of shooting incidents across different boroughs in New York City. Areas with a greater number of incidents have higher color opacity.

A line plot of incidents per thousand residents by year and borough with summary stats

```
# Create a line plot
ggplot(shooting_by_year, aes(x = YEAR, y = INCIDENTS_PER_THOU, color = BORO)) +
  geom_line() +
  geom_point() +
  scale_color_manual(values = borough_colors) +
  labs(title = "Shooting Incidents per Thousand Residents by Borough (2006-2023)",
       x = "Year",
       y = "Incidents per Thousand Residents",
       color = "Borough") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  scale_x_continuous(breaks = seq(2006, 2023, by = 2))
```




```r
# Summary statistics
shooting_by_year %>%
  group_by(BORO) %>%
  summarize(
    Mean = mean(INCIDENTS_PER_THOU),
    Median = median(INCIDENTS_PER_THOU),
    Min = min(INCIDENTS_PER_THOU),
    Max = max(INCIDENTS_PER_THOU)
  ) %>%
  arrange(desc(Mean))
```

```
## # A tibble: 5 x 5
##   BORO            Mean Median    Min   Max
##   <chr>          <dbl>  <dbl>  <dbl> <dbl>
## 1 BRONX         0.330  0.357  0.185  0.482
## 2 BROOKLYN      0.244  0.237  0.136  0.338
## 3 MANHATTAN     0.130  0.123  0.0645 0.209
## 4 QUEENS        0.104  0.110  0.0624 0.145
## 5 STATEN ISLAND 0.0944 0.0991 0.0517 0.149
```

Key Observations from the Plot

- The Bronx consistently has the highest rate of shooting incidents per thousand residents.
- Brooklyn generally has the second-highest rate.
- Manhattan, Queens, and Staten Island cluster together at the bottom of the graph.

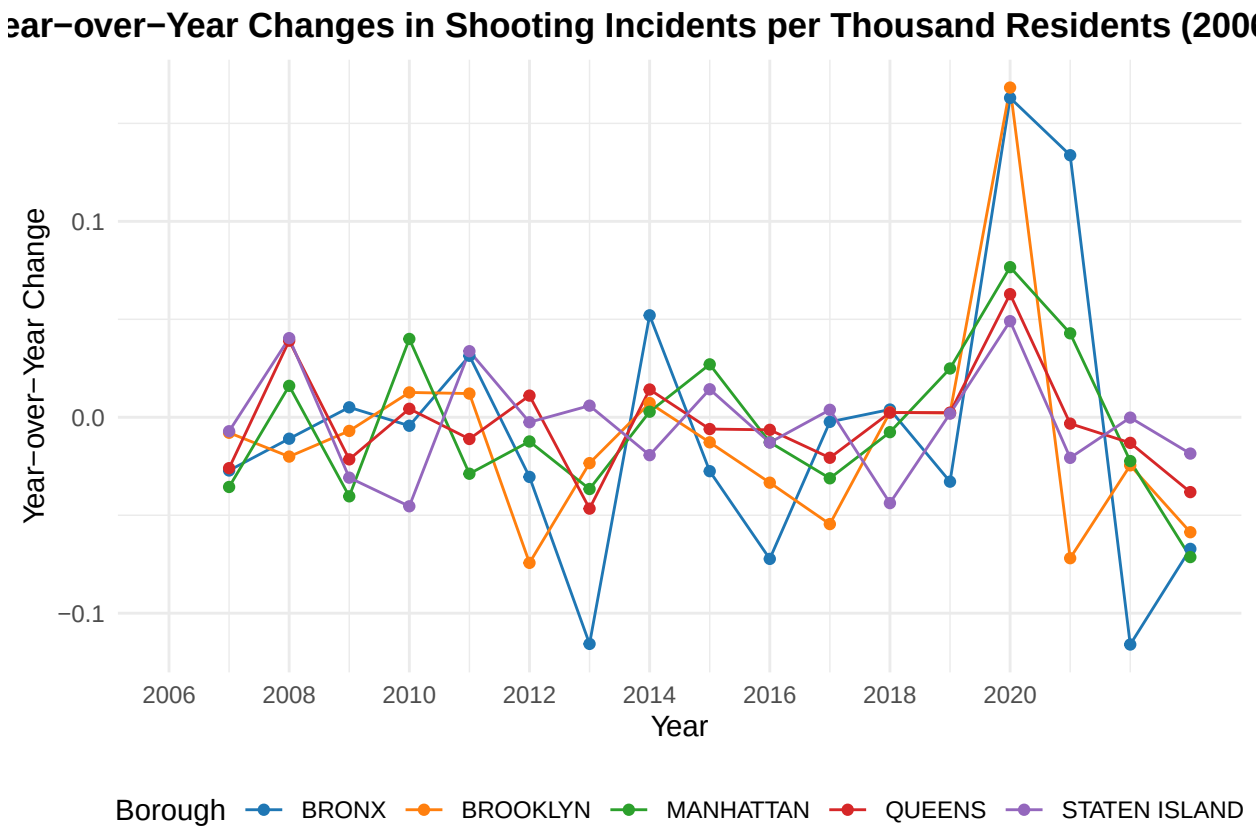
- There's a noticeable spike in incidents for all boroughs around 2019, probably related to the COVID-19 pandemic and associated social impacts.
- After the spike, rates seem to be decreasing but haven't returned to pre-spike levels.

A line plot of year-over-year changes in shooting incidents per thousand residents by borough with summary stats

```
# Calculate year-over-year changes in incidents per thousand
shooting_changes_yoy <- shooting_by_year %>%
  group_by(BORO) %>%
  arrange(YEAR) %>%
  mutate(YOY_CHANGE = INCIDENTS_PER_THOU - lag(INCIDENTS_PER_THOU)) %>%
  ungroup()

# Create the plot
ggplot(shooting_changes_yoy, aes(x = YEAR, y = YOY_CHANGE, color = BORO)) +
  geom_line() +
  geom_point() +
  scale_color_manual(values = borough_colors) +
  labs(title = "Year-over-Year Changes in Shooting Incidents per Thousand Residents (2006-2023)",
       x = "Year",
       y = "Year-over-Year Change",
       color = "Borough") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  scale_x_continuous(breaks = seq(2006, 2020, by = 2))
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```r
# Summary statistics
shooting_changes_yoy %>%
  group_by(BORO) %>%
  summarize(
    Mean_YOY_Change = mean(YOY_CHANGE, na.rm = TRUE),
    Median_YOY_Change = median(YOY_CHANGE, na.rm = TRUE),
    Min_YOY_Change = min(YOY_CHANGE, na.rm = TRUE),
    Max_YOY_Change = max(YOY_CHANGE, na.rm = TRUE)
  ) %>%
  arrange(desc(Mean_YOY_Change))
```

```
## # A tibble: 5 x 5
##   BORO          Mean_YOY_Change Median_YOY_Change Min_YOY_Change Max_YOY_Change
##   <chr>                   <dbl>             <dbl>          <dbl>          <dbl>
## 1 STATEN ISLAND        -0.00308          -0.00253        -0.0454         0.0491
## 2 QUEENS               -0.00332          -0.00601        -0.0466         0.0629
## 3 MANHATTAN            -0.00404          -0.0123         -0.0714         0.0766
## 4 BRONX                -0.00691          -0.0109         -0.116          0.163
## 5 BROOKLYN             -0.0108           -0.0128         -0.0743         0.168
```

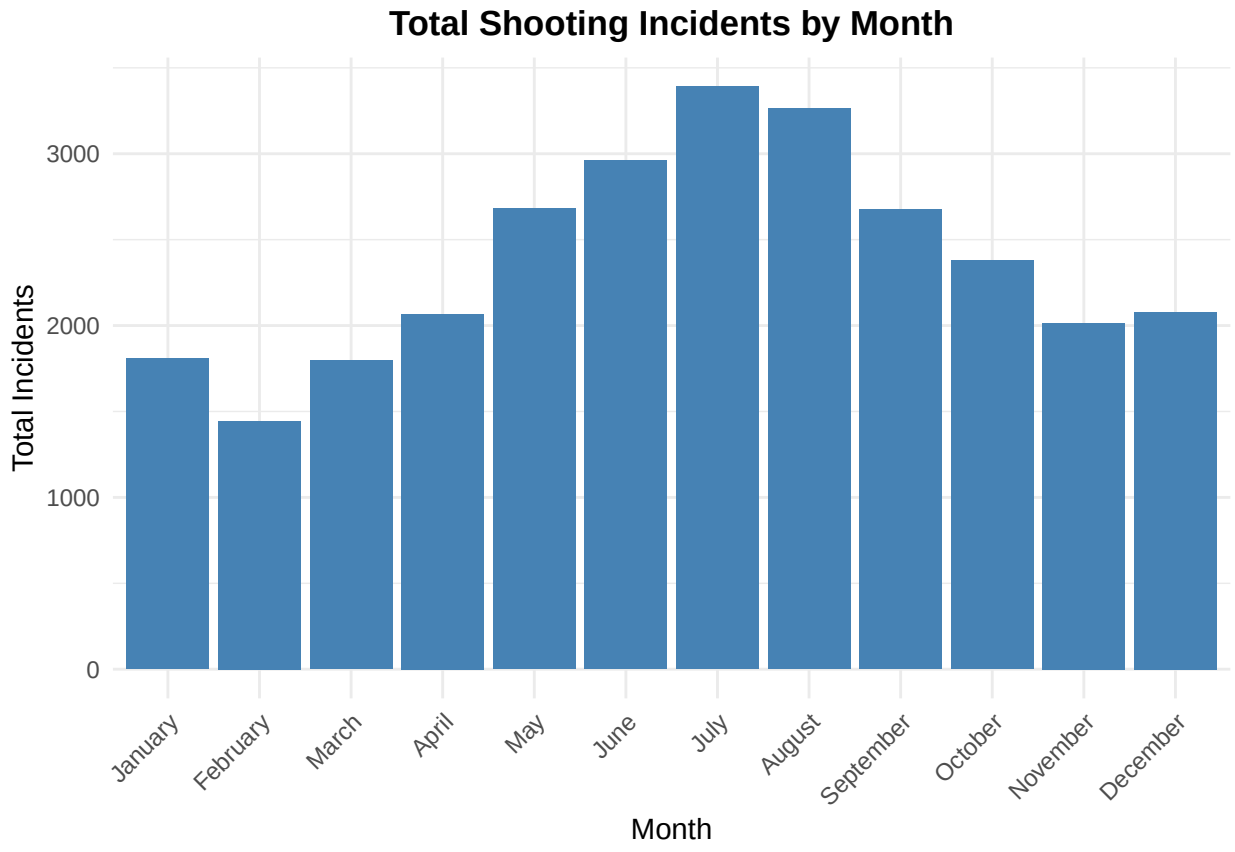
Key Observations from the Plot

- Most boroughs show a similar pattern of changes, with peaks and troughs often occurring in the same years.
- There's a notable spike in changes around 2019, likely corresponding to the COVID-19 pandemic.

- The Bronx has the largest magnitude of changes, both positive and negative.

Plot of all incidents by month

```
# Aggregate the data by month
shooting_data <- shooting_data %>%  mutate(MONTH = month(OCCUR_DATE, label = TRUE, abbr = FALSE))

shooting_by_month <- shooting_data %>%
  group_by(MONTH) %>%
  summarize(TOTAL_INCIDENTS = n()) %>%
  arrange(match(MONTH, month.name))

# Create a bar plot
ggplot(shooting_by_month, aes(x = MONTH, y = TOTAL_INCIDENTS)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Total Shooting Incidents by Month",
       x = "Month",
       y = "Total Incidents") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5, face = "bold"))
```



Key Observations from the Graph

- There appears to be seasonal patterns in shooting incidents.
- The number of incidents varies across the months, with summer months experiencing higher incidents than others.

Model the data

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

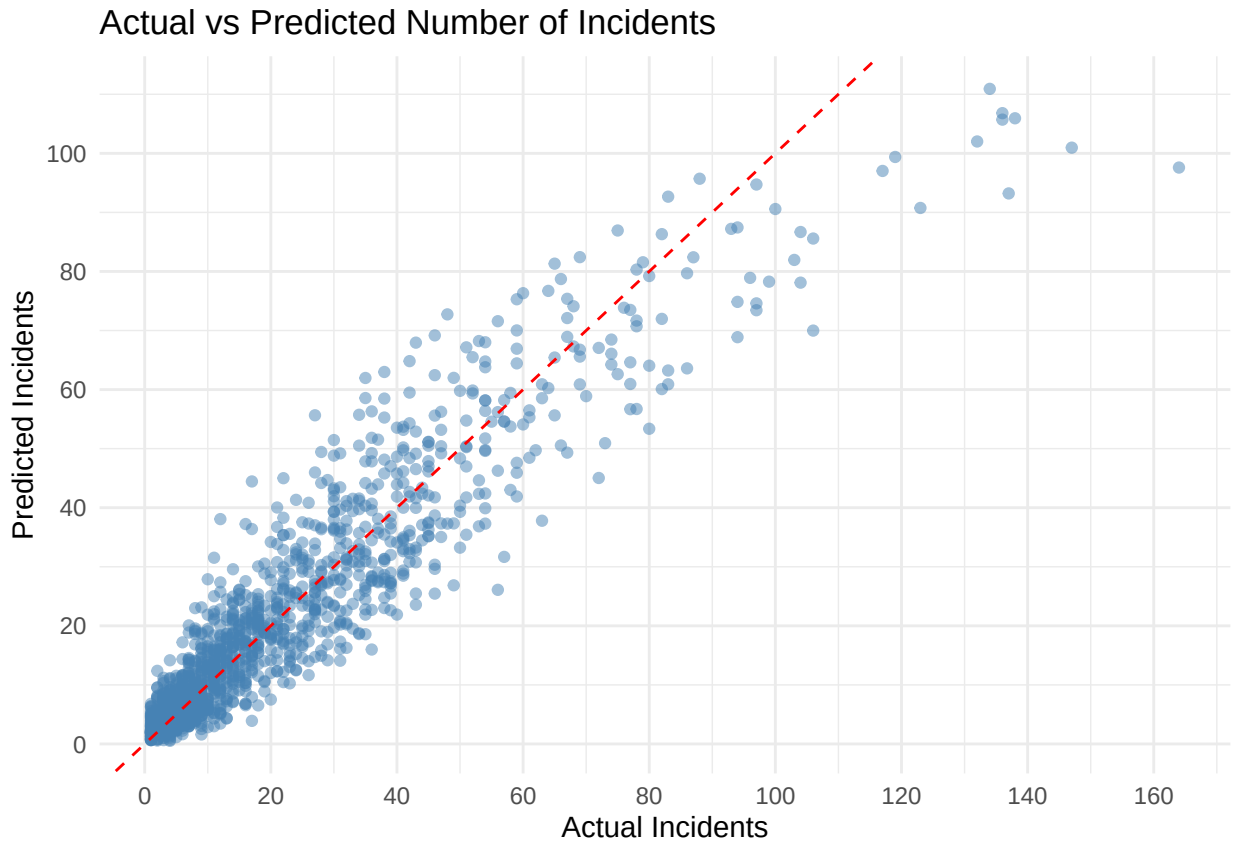
```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
# Prepare the data
model_data <- shooting_data %>%
  mutate(MONTH = as.numeric(MONTH)) %>%
  group_by(HOUR, MONTH, BORO) %>%
  summarize(Incidents = n()) %>%
  ungroup()
```

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

```
# Create GAM model
gam_model <- gam(Incidents ~ s(HOUR) + s(MONTH) + BORO, data = model_data, family = poisson())

# Predict using GAM model
model_data <- model_data %>%
  mutate(Predicted = predict(gam_model, newdata = model_data, type = "response"))

# Define the breaks for x and y axes
breaks_seq <- seq(0, max(c(model_data$Incidents, model_data$Predicted)), by = 20)

# Create a plot of actual vs predicted incidents
actual_vs_predicted_plot <- ggplot(model_data, aes(x = Incidents, y = Predicted)) +
  geom_point(alpha = 0.5, color = "steelblue") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs Predicted Number of Incidents",
       x = "Actual Incidents",
       y = "Predicted Incidents") +
  theme_minimal() +
  scale_x_continuous(breaks = breaks_seq) +
```

```
  scale_y_continuous(breaks = breaks_seq)

print(actual_vs_predicted_plot)
```




```
# Calculate and display R-squared value for the GAM model
gam_r2 <- 1 - sum((model_data$Incidents - model_data$Predicted)^2) /
  sum((model_data$Incidents - mean(model_data$Incidents))^2)
cat("GAM R-squared:", gam_r2)
```

```
## GAM R-squared: 0.8686003
```

Key Observations from the Plot

- Most points cluster around the red line, demonstrating that the model's predictions are generally good.
- There's some spread around the line for higher numbers of incidents, indicating that the model's accuracy may decrease for very high values.

Key Observations from the Model's Performance

- The R-squared value of 0.8686 means that our GAM model explains about 86.86% of the variance in the number of shooting incidents. This high R-squared value suggests that the model fits the data well and has good predictive power. Thus, the factors time of day, time of year, and borough are strong predictors of shooting incidents in New York City.

- While the model is good, there is room for improvement. There are likely socio-economic factors not included in the model that play a role in the occurrence of shooting incidents.

Conclusion

This project analyzed shooting incident data in New York City, examining trends across boroughs in relation to various parameters such as month and hour of the day. Our analysis revealed several key findings that could be valuable for stakeholders, such as law enforcement and policymakers, in developing targeted strategies to reduce gun violence in New York City:

- Shooting incidents vary across boroughs, with the Bronx consistently showing the highest rate of incidents per thousand residents, followed by Brooklyn.
- There's a seasonal pattern in shooting incidents, with summer months generally having higher numbers of incidents compared to winter months.
- The COVID-19 era (around 2019) saw a significant spike in shooting incidents across all boroughs, followed by a decline, though it has not yet returned to pre-pandemic levels.
- Our Generalized Additive Model (GAM) demonstrated that hour of the day, month of the year, and borough are strong predictors of number of shooting incidents, explaining about 86.86% of the variance in incident numbers.

Possible Sources of Bias

- **Reporting Bias**: Changes in reporting practices or law enforcement strategies over time could affect the consistency of the data.
- **Data Collection Bias**: Incidents may be underreported in certain communities due to factors such as distrust in law enforcement or fear of retaliation.
- **Demographic Bias**: Our analysis doesn't account for demographic factors such as age, race, or socioeconomic status, which are important in understanding the full context of shooting incidents.

Personal Biases and Mitigation

- **Confirmation Bias**: It is possible I looked for patterns that confirmed my preexisting beliefs around crime. To mitigate this, I recorded any assumptions I may have before beginning my report Furthermore, I advanced my analysis by relying on insights gained from data visualization techniques. I also minimized bias by focusing on objective data measures.
- **Oversimplification Bias**: There is a risk of oversimplify complex social issues into purely statistical terms. I've tried to mitigate this by acknowledging the limitations of my analysis.

Further Analysis

- Include additional socioeconomic and demographic variables in the analysis.
- Conduct a more granular analysis at the community district level.