Analysis of NYPD Historic Shooting Incidents using R

Marco F

2025-06-15

Contents

0.1 Introduction

This report conducts an analysis of the NYPD Shooting Incident Data (Historic). The primary goal is to explore patterns, trends, and characteristics of shooting incidents in New York City. We will follow a standard data analysis workflow:

1. **Importing Data**: Loading the dataset into R.
2. **Data Cleaning**: Preparing the data for analysis by handling missing values, correcting data types, and transforming variables.
3. **Visualizing Data**: Creating visualizations to understand distributions and relationships.
4. **Analyzing and Modeling**: Performing statistical analysis and building a predictive model.
5. **Bias Discussion**: Identifying potential biases in the data and analysis.
6. **Conclusion**: Summarizing key findings and limitations.

The dataset is publicly available from NYC OpenData.

0.2 1. Importing Data

First, we load the necessary R packages and import the dataset. We'll use `tidyverse` for general data manipulation and visualization, `lubridate` for date-time operations, `skimr` for quick summaries, `rpart` and `rpart.plot` for decision tree modeling.

```
# Install packages if not already installed
# install.packages(c("tidyverse", "lubridate", "skimr", "rpart", "rpart.plot", "kableExtra"))

library(tidyverse)
library(lubridate)
library(skimr)
library(rpart)
library(rpart.plot)
library(kableExtra)
library(broom)

# URL for the NYPD Shooting Incident Data (Historic) - Same as https://data.cityofnewyork.us/api/views/
data_url <- "https://raw.githubusercontent.com/marcofanti/Final-Project-1-Data-Science-NYPD-Shooting-In

# Load the dataset
nypd_shootings_raw <- read_csv(data_url, show_col_types = FALSE)

cat("Data loaded successfully.\n")
```

```
## Data loaded successfully.
```

```
cat(paste("Dataset dimensions:", paste(dim(nypd_shootings_raw), collapse = " x "), "\n\n"))
```

```
## Dataset dimensions: 29744 x 21
```

```
cat("First few rows of raw data:\n")
```

```
## First few rows of raw data:
```

```
kable(head(nypd_shootings_raw))
```

INCI																				Lat
2319	08/		BRON	X	40	0	NA	NA	FALSE	NA	NA	NA	18-24	M	BLA	1			40.8	POINT (73.92019278899994 40.80967347200004)
1779	04/		BROO	LYN	79	0	NA	NA	TRUE	25-44	M	WHITE HIS-PANIC	25-44	M	BLA	1			40.6	POINT (73.94291302299996 40.685609672000055)
2550	12/		BRON	X	17	0	STREET	GROCERY STORE	FALSE	(null)	(null)	(null)	25-44	M	BLA	1			40.8	POINT (73.868233 40.872349)
2538	11/		BROO	LYN	66	0	NA	PVT HOUSE	TRUE	UNKNOWN	UNKNOWN	UNKNOWN	18-24	M	BLA	1			40.6	POINT (73.99691224999998 40.642489932000046)

INCI	OCC	BORO	LOC	PRE	JURIS	PERP	VIC	VIC	VIC	Longitude	Lat
72616	02/30/2	BRONX	46	0	NA	MULTI-DWELL - APT BUILD	TRUE	25-44	M	BLACK	18	F	BLACK	1089457	402.64598	POINT (-73.90746 40.84598
85875	07/22/2	BRONX	42	2	NA	MULTI-DWELL - PUB-LIC HOUS	FALSE	18-24	M	BLACK	18-24	M	BLACK	1013084	410.82488	POINT (-73.90318 40.82487

```
cat("\n\nSummary of raw data:\n")
```

```
##
##
## Summary of raw data:
```

```
skim(nypd_shootings_raw) %>%
  select(skim_type, skim_variable, n_missing, complete_rate, character.empty, numeric.mean)
```

Table 2: Data summary

Name	nypd_shootings_raw
Number of rows	29744
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	empty
OCCUR_DATE	0	1.00	0
BORO	0	1.00	0
LOC_OF_OCCUR_DESC	25596	0.14	0
LOC_CLASSFCTN_DESC	25596	0.14	0
LOCATION_DESC	14977	0.50	0
PERP_AGE_GROUP	9344	0.69	0
PERP_SEX	9310	0.69	0
PERP_RACE	9310	0.69	0
VIC_AGE_GROUP	0	1.00	0
VIC_SEX	0	1.00	0

skim_variable	n_missing	complete_rate	empty
VIC_RACE	0	1.00	0
Lon_Lat	97	1.00	0

Variable type: difftime

skim_variable	n_missing	complete_rate
OCCUR_TIME	0	1

Variable type: logical

skim_variable	n_missing	complete_rate
STATISTICAL_MURDER_FLAG	0	1

Variable type: numeric

skim_variable	n_missing	complete_rate	mean
INCIDENT_KEY	0	1	133850951.08
PRECINCT	0	1	65.23
JURISDICTION_CODE	2	1	0.32
X_COORD_CD	0	1	1009442.08
Y_COORD_CD	0	1	208721.99
Latitude	97	1	40.74
Longitude	97	1	-73.91

0.3 2. Data Cleaning

Data cleaning is crucial for reliable analysis. This section involves:

Standardizing column names.

Converting data types (e.g., dates, times).

Handling missing values.

Feature engineering (e.g., extracting year, month, hour).

```
nypd_shootings_clean <- nypd_shootings_raw %>%
  # Standardize column names (lowercase and replace spaces with underscores)
  rename_with(tolower) %>%
  rename_with(~gsub(" ", "_", .x))

# Convert OCCUR_DATE to Date object
nypd_shootings_clean <- nypd_shootings_clean %>%
  mutate(occur_date = mdy(occur_date)) # Assumes MM/DD/YYYY format

# Extract year, month, day_of_week
nypd_shootings_clean <- nypd_shootings_clean %>%
  mutate(
```

```
    year = year(occur_date),
    month = month(occur_date, label = TRUE, abbr = FALSE),
    day_of_week = wday(occur_date, label = TRUE, abbr = FALSE)
  )

# Handle OCCUR_TIME - convert to hour
nypd_shootings_clean <- nypd_shootings_clean %>%
  mutate(
    occur_hour = case_when(
      !is.na(occur_time) ~ hour(hms(occur_time, quiet = TRUE)), # Try to parse HH:MM:SS
      TRUE ~ NA_integer_ # If parsing fails or is NA, keep as NA
    )
  )

# Impute missing occur_hour with the median hour
median_hour <- median(nypd_shootings_clean$occur_hour, na.rm = TRUE)
nypd_shootings_clean <- nypd_shootings_clean %>%
  mutate(occur_hour = ifelse(is.na(occur_hour), median_hour, occur_hour))

# Handle missing values for key categorical features by replacing with 'UNKNOWN'
cols_to_fill_unknown <- c(
  "boro", "perp_age_group", "perp_sex", "perp_race",
  "vic_age_group", "vic_sex", "vic_race",
  "loc_of_occur_desc", "loc_classfctn_desc", "location_desc"
)

# Ensure these columns exist before trying to mutate them
existing_cols_to_fill <- intersect(cols_to_fill_unknown, names(nypd_shootings_clean))

nypd_shootings_clean <- nypd_shootings_clean %>%
  mutate(across(all_of(existing_cols_to_fill), ~replace_na(as.character(.x), "UNKNOWN"))) %>%
  mutate(across(all_of(existing_cols_to_fill), ~ifelse(.x %in% c("", "(null)"), "UNKNOWN", .x)))


# Convert STATISTICAL_MURDER_FLAG to a factor (0 for No, 1 for Yes)
# The column might be logical (TRUE/FALSE) or character ("true"/"false")
nypd_shootings_clean <- nypd_shootings_clean %>%
  mutate(
    statistical_murder_flag = case_when(
      is.logical(statistical_murder_flag) ~ factor(statistical_murder_flag, levels = c(FALSE, TRUE), lab
      is.character(statistical_murder_flag) ~ factor(tolower(statistical_murder_flag), levels = c("false
      TRUE ~ factor(NA, levels = c("No", "Yes")) # Handle other cases or if column doesn't exist as exp
    ),
    # If it was already T/F, ensure NAs are handled (e.g. replace with "No" or a specific category)
    statistical_murder_flag = replace_na(statistical_murder_flag, "No")
  )

# The following variables should be treated as factors:
#
# - boro
# - perp_age_group
# - perp_sex
# - perp_race
```

```r
# - vic_age_group
# - vic_sex
# - vic_race
# - statistical_murder_flag

nypd_shootings_clean$boro <- factor(nypd_shootings_clean$boro)
nypd_shootings_clean$perp_age_group <- factor(nypd_shootings_clean$perp_age_group)
nypd_shootings_clean$perp_sex <- factor(nypd_shootings_clean$perp_sex)
nypd_shootings_clean$perp_race <- factor(nypd_shootings_clean$perp_race)
nypd_shootings_clean$vic_age_group <- factor(nypd_shootings_clean$vic_age_group)
nypd_shootings_clean$vic_sex <- factor(nypd_shootings_clean$vic_sex)
nypd_shootings_clean$vic_race <- factor(nypd_shootings_clean$vic_race)
nypd_shootings_clean$statistical_murder_flag <- factor(nypd_shootings_clean$statistical_murder_flag)

# Drop columns not immediately useful for this analysis or with too many unique values for simple model
# For example, specific coordinates if lat/long are used, or high-cardinality IDs.
# `incident_key` is an ID, `lon_lat` is a point geometry.
cols_to_drop <- c("x_coord_cd", "y_coord_cd", "lon_lat", "jurisdiction_code", "incident_key", "patch_co
# Ensure columns exist before trying to drop
existing_cols_to_drop <- intersect(cols_to_drop, names(nypd_shootings_clean))
if (length(existing_cols_to_drop) > 0) {
  nypd_shootings_clean <- nypd_shootings_clean %>%
    select(-all_of(existing_cols_to_drop))
}

cat("\nCleaned data summary:\n")
```

```
##
## Cleaned data summary:
```

```r
skim(nypd_shootings_clean) %>%
  select(skim_type, skim_variable, n_missing, complete_rate, character.empty, numeric.mean, factor.orde
```

Table 7: Data summary

Name	nypd_shootings_clean
Number of rows	29744
Number of columns	20
Column type frequency:	
character	3
Date	1
difftime	1
factor	10
numeric	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	empty
loc_of_occur_desc	0	1	0
loc_classfctn_desc	0	1	0
location_desc	0	1	0

Variable type: Date

skim_variable	n_missing	complete_rate
occur_date	0	1

Variable type: difftime

skim_variable	n_missing	complete_rate
occur_time	0	1

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique
boro	0	1	FALSE	5
statistical_murder_flag	0	1	FALSE	2
perp_age_group	0	1	FALSE	11
perp_sex	0	1	FALSE	4
perp_race	0	1	FALSE	7
vic_age_group	0	1	FALSE	7
vic_sex	0	1	FALSE	3
vic_race	0	1	FALSE	7
month	0	1	TRUE	12
day_of_week	0	1	TRUE	7

Variable type: numeric

skim_variable	n_missing	complete_rate	mean
precinct	0	1	65.23
latitude	97	1	40.74
longitude	97	1	-73.91
year	0	1	2014.31
occur_hour	0	1	12.30

```
cat("\nFirst few rows of cleaned data:\n")
```

```
##
## First few rows of cleaned data:
```

```
kable(head(nypd_shootings_clean))
```

occur_date	occur_time	loc_of_occur_desc	precinct	juris_desc	jurisdiction_code	loc_classfctn_desc	location_desc	statistical_murder_flag	perp_age_group	perp_sex	perp_race	vic_age_group	vic_sex	vic_race	latitude	longitude	year	month	day_of_week	hour
2021-08-09 01:06:00	BRONX	NA	40	UNKNOWN	0	UNKNOWN	UNKNOWN	No	UNKNOWN	U	UNKNOWN	18-24	M	BLACK	40.80967	-73.92019	2021	August	Monday	1
2018-04-07 19:48:00	BRONX	NA	49	UNKNOWN	0	UNKNOWN	UNKNOWN	Yes	25-44	M	WHITE HISPANIC	25-44	M	BLACK	40.86856	-73.94291	2018	April	Saturday	19
2022-12-02 22:57:00	BRONX	47	47	PATROL BORO BRONX	107	GROCERY/BODEGA	47 STREET GROCERY	No	UNKNOWN	U	UNKNOWN	25-44	M	BLACK	40.87235	-73.86823	2022	December	Friday	22
2006-11-19 01:50:00	BRONX	NA	46	UNKNOWN	0	OTHER	OTHER HOUSE	Yes	UNKNOWN	U	UNKNOWN	18-24	M	BLACK	40.84249	-73.99691	2006	November	Sunday	1
2010-05-09 01:58:00	BRONX	NA	46	UNKNOWN	0	MULTI DWELL - APT BUILD	Yes	25-44	M	BLACK	18-24	F	BLACK	40.84598	-73.90746	2010	May	Sunday	1
2012-07-22 21:35:00	BRONX	NA	42	UNKNOWN	0	MULTI DWELL - PUBLIC HOUS	No	18-24	M	BLACK	18-24	M	BLACK	40.82488	-73.90318	2012	July	Sunday	21

```
cat("\nMissing values count per column after cleaning:\n")
```

```
##
## Missing values count per column after cleaning:
```

```
sapply(nypd_shootings_clean, function(x) sum(is.na(x))) %>%
  data.frame(missing_count = .) %>%
  rownames_to_column("column_name") %>%
  filter(missing_count > 0) %>%
  kable()
```

column_name	missing_count
latitude	97
longitude	97

0.4 3. Visualizing Data

Visualizations help in understanding trends, distributions, and relationships.

0.4.1 Visualization 1: Shooting Incidents Over Years

This plot shows the trend of shooting incidents annually.

```r
if (nrow(nypd_shootings_clean) > 0 && "year" %in% names(nypd_shootings_clean)) {
  incidents_by_year <- nypd_shootings_clean %>%
    filter(!is.na(year)) %>% # Ensure year is not NA
    group_by(year) %>%
    summarise(count = n(), .groups = 'drop') %>%
    filter(year >= min(nypd_shootings_clean$year, na.rm=TRUE) & year <= max(nypd_shootings_clean$year,

  ggplot(incidents_by_year, aes(x = year, y = count)) +
    geom_line(color = "dodgerblue", size = 1) +
    geom_point(color = "dodgerblue", size = 2) +
    labs(
      title = "Total Shooting Incidents per Year",
      x = "Year",
      y = "Number of Incidents"
    ) +
    theme_minimal(base_size = 12) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    scale_x_continuous(breaks = seq(min(incidents_by_year$year, na.rm=TRUE), max(incidents_by_year$year
} else {
  cat("Skipping yearly trend visualization as data is empty or 'year' column is missing.\n")
}
```

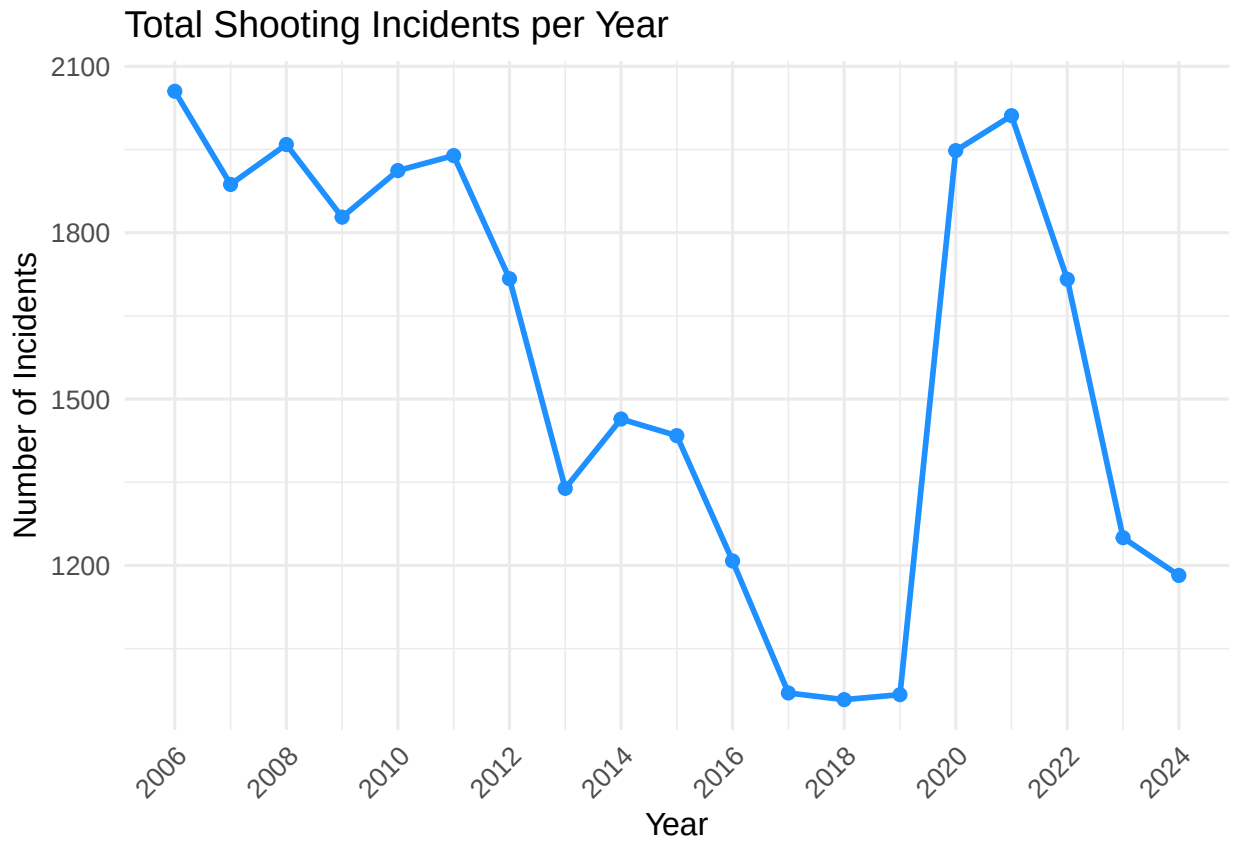


Figure 1: Number of Shooting Incidents per Year

Interpretation: This line chart illustrates the annual frequency of shooting incidents. It can reveal long-term

trends, such as increases, decreases, or periods of stability in shooting occurrences.

0.4.2 Visualization 2: Shooting Incidents by Borough

This bar chart shows the distribution of shooting incidents across NYC boroughs.

```
if (nrow(nypd_shootings_clean) > 0 && "boro" %in% names(nypd_shootings_clean)) {
  incidents_by_boro <- nypd_shootings_clean %>%
    filter(!is.na(boro) & boro != "UNKNOWN") %>% # Exclude NA or UNKNOWN for a cleaner plot
    group_by(boro) %>%
    summarise(count = n(), .groups = 'drop') %>%
    arrange(desc(count))

  ggplot(incidents_by_boro, aes(x = reorder(boro, -count), y = count, fill = boro)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    geom_text(aes(label = count), vjust = -0.5, size = 3.5) +
    labs(
      title = "Shooting Incidents by Borough",
      x = "Borough",
      y = "Number of Incidents"
    ) +
    scale_fill_brewer(palette = "Set2") +
    theme_minimal(base_size = 12) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
} else {
  cat("Skipping borough distribution visualization as data is empty or 'boro' column is missing.\n")
}
```

Interpretation: This bar chart highlights which boroughs experience the highest and lowest numbers of shooting incidents, providing insight into the geographical distribution of these events.

0.4.3 Visualization 3: Shooting Incidents by Borough Per 100K Residents

This bar chart shows the distribution of shooting incidents across NYC boroughs per 100K residents.

```
# Load population data for 2020 - Same as "https://data.cityofnewyork.us/resource/xywu-7bv9.csv")
pop_url <- "https://raw.githubusercontent.com/marcofanti/Final-Project-1-Data-Science-NYPD-Shooting-Inci

population <- read_csv(pop_url, show_col_types = FALSE)

population <- population %>%
  # Standardize column names (lowercase and replace spaces with underscores)
  rename_with(tolower) %>%
  rename_with(~gsub(" ", "_", .x))

population_2020 <- population %>%
        filter(borough != "NYC Total") %>%
        select(borough, `2020`) %>%
        rename(boro = borough, Population = `2020`) %>%
        mutate(boro = toupper(boro))
```

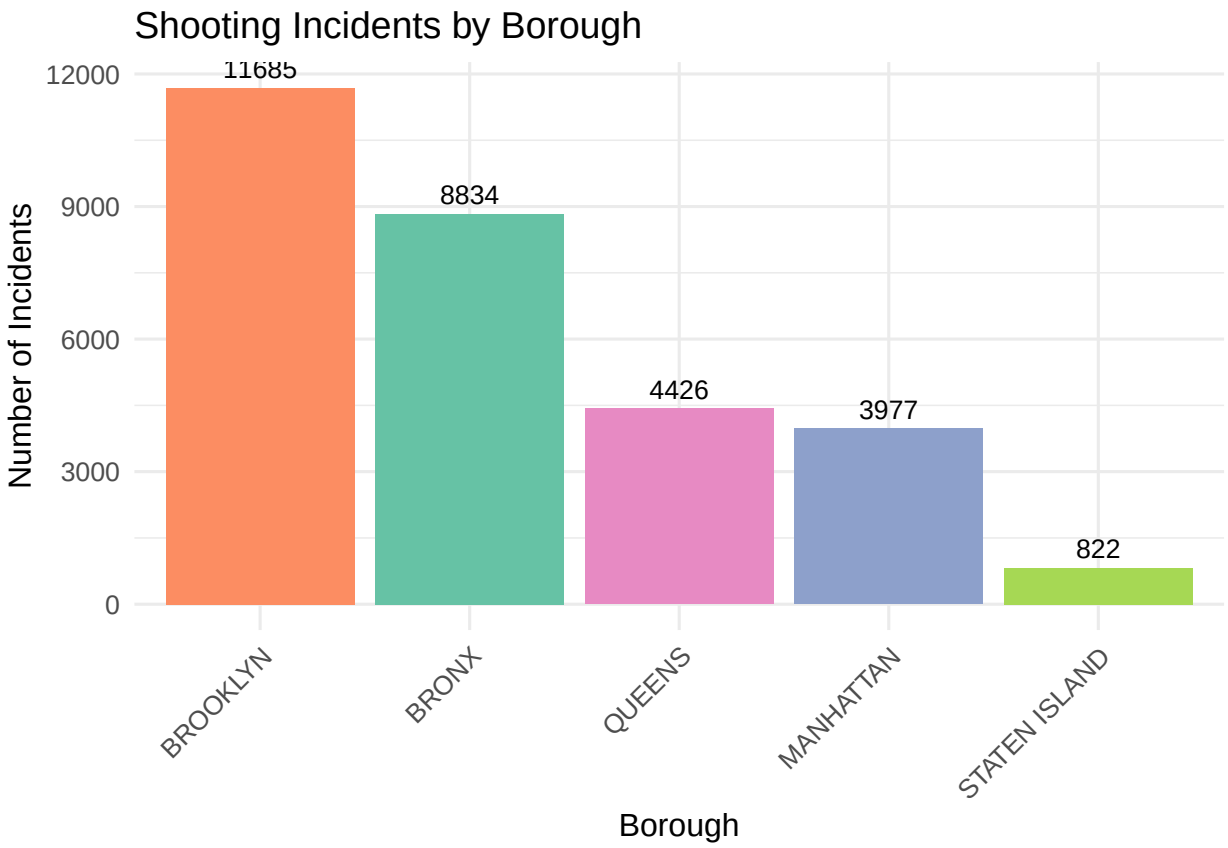
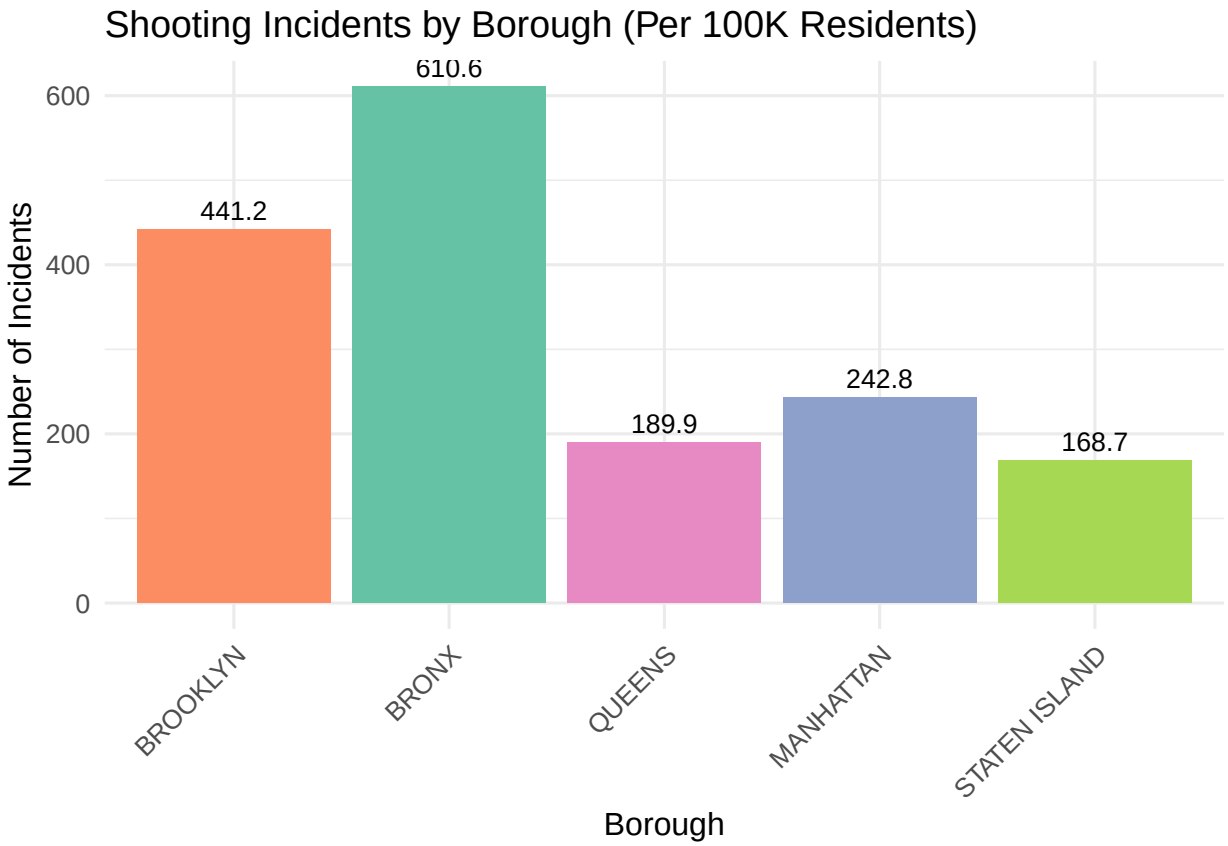


Figure 2: Distribution of Shooting Incidents by Borough

```r
if (nrow(nypd_shootings_clean) > 0 && "boro" %in% names(nypd_shootings_clean)) {
  incidents_by_boro <- nypd_shootings_clean %>%
    filter(!is.na(boro) & boro != "UNKNOWN") %>% # Exclude NA or UNKNOWN for a cleaner plot
    group_by(boro) %>%
    summarise(count = n(), .groups = 'drop') %>%
    arrange(desc(count))

  # Join with population data
  incidents_population <- incidents_by_boro %>%
        left_join(population_2020, by = "boro") %>%
        mutate(Rate_Per_100k = (count / Population) * 100000)


  ggplot(incidents_population, aes(x = reorder(boro, -count), y = Rate_Per_100k, fill = boro)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    geom_text(aes(label = round(Rate_Per_100k, 1)), vjust = -0.5, size = 3.5) +
    labs(
      title = "Shooting Incidents by Borough (Per 100K Residents)",
      x = "Borough",
      y = "Number of Incidents"
    ) +
    scale_fill_brewer(palette = "Set2") +
    theme_minimal(base_size = 12) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
} else {
  cat("Skipping borough distribution visualization as data is empty or 'boro' column is missing.\n")
}
```



Interpretation: This bar chart highlights which boroughs experience the highest and lowest numbers of shooting incidents per 100K residents. If we look at the incidence rate per 100,000 residents, and compare to the previous visualization, the Bronx has the highest number of shootings per capita, and Manhattan overtakes Queens in this metric.

0.4.4 Visualization 4: Incidents by Hour of Day

This visualization explores the temporal pattern of shootings throughout the day.

```
if (nrow(nypd_shootings_clean) > 0 && "occur_hour" %in% names(nypd_shootings_clean)) {
  incidents_by_hour <- nypd_shootings_clean %>%
    filter(!is.na(occur_hour)) %>%
    group_by(occur_hour) %>%
    summarise(count = n(), .groups = 'drop')

  ggplot(incidents_by_hour, aes(x = occur_hour, y = count)) +
    geom_col(fill = "coral") +
    labs(
      title = "Shooting Incidents by Hour of Day",
      x = "Hour of Day (0-23)",
      y = "Number of Incidents"
    ) +
    scale_x_continuous(breaks = seq(0, 23, by = 2)) +
    theme_minimal(base_size = 12)
} else {
  cat("Skipping hour distribution visualization as data is empty or 'occur_hour' column is missing.\n")
```

```
}
```

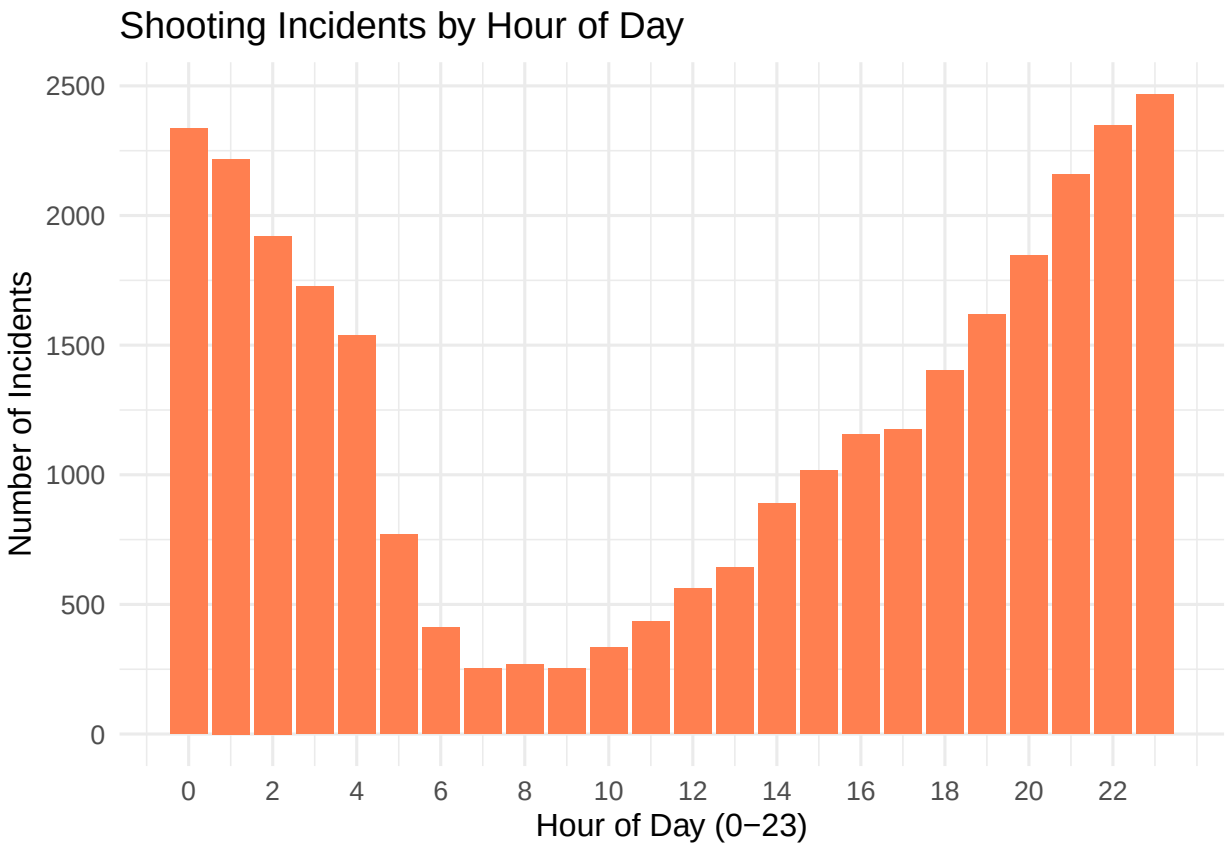


Figure 3: Shooting Incidents by Hour of Day

Interpretation: This plot shows if there are particular times of day when shooting incidents are more frequent.

0.5 4. Analyzing and Modeling

0.5.1 Logistic Regression Analysis of Fatal Shooting Likelihood

We model whether an incident was fatal (`statistical_murder_flag`) based on shooting incident's borough (`boro`), time of occurrence (`occur_hour`), victim's age group, sex, and race.

```
if (nrow(nypd_shootings_clean) > 0 && "statistical_murder_flag" %in% names(nypd_shootings_clean)) {
  # Select features for modeling
  # Ensure all selected features exist in the dataframe
  feature_candidates <- c(
    "boro", "occur_hour", "vic_age_group", "vic_sex", "vic_race"
  )

  model_features <- intersect(feature_candidates, names(nypd_shootings_clean))

  model_data <- nypd_shootings_clean %>%
    select(all_of(model_features)) %>%
```

```
    na.omit() # Remove rows with any NAs in selected features for simplicity

  # Convert character columns to factors for rpart
  model_data <- model_data %>%
    mutate(across(where(is.character), as.factor)) %>%
    mutate(across(where(is.logical), as.factor)) # Ensure logicals are factors too

    cat(paste("Dimensions of data for modeling:", paste(dim(model_data), collapse = " x "), "\n"))
}
```

```
## Dimensions of data for modeling: 29744 x 5
```

```
glm_model <- glm(statistical_murder_flag ~ boro + occur_hour + vic_age_group + vic_sex + vic_race,
                 data = nypd_shootings_clean, family = 'binomial')

summary(glm_model)
```

```
##
## Call:
## glm(formula = statistical_murder_flag ~ boro + occur_hour + vic_age_group +
##     vic_sex + vic_race, family = "binomial", data = nypd_shootings_clean)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -12.907402  89.668090  -0.144   0.8855
## boroBROOKLYN                    -0.022239   0.037019  -0.601   0.5480
## boroMANHATTAN                   -0.117731   0.049540  -2.377   0.0175 *
## boroQUEENS                      -0.035756   0.047439  -0.754   0.4510
## boroSTATEN ISLAND                0.020945   0.091556   0.229   0.8190
## occur_hour                       0.001628   0.001761   0.925   0.3550
## vic_age_group1022              -10.591006 324.743703  -0.033   0.9740
## vic_age_group18-24               0.261897   0.059435   4.406 1.05e-05 ***
## vic_age_group25-44               0.601137   0.057386  10.475  < 2e-16 ***
## vic_age_group45-64               0.759364   0.073527  10.328  < 2e-16 ***
## vic_age_group65+                 0.992271   0.151986   6.529 6.63e-11 ***
## vic_age_groupUNKNOWN             0.772756   0.313837   2.462   0.0138 *
## vic_sexM                        -0.030337   0.049629  -0.611   0.5410
## vic_sexU                        -0.549268   1.079340  -0.509   0.6108
## vic_raceASIAN / PACIFIC ISLANDER 11.326862  89.668125   0.126   0.8995
## vic_raceBLACK                   11.054359  89.668063   0.123   0.9019
## vic_raceBLACK HISPANIC          10.903067  89.668075   0.122   0.9032
## vic_raceUNKNOWN                 10.211109  89.669046   0.114   0.9093
## vic_raceWHITE                   11.372708  89.668102   0.127   0.8991
## vic_raceWHITE HISPANIC          11.169376  89.668069   0.125   0.9009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29251  on 29743  degrees of freedom
## Residual deviance: 28943  on 29724  degrees of freedom
## AIC: 28983
##
```

```
## Number of Fisher Scoring iterations: 11
```

0.6 Generalized Linear Model (GLM) Output Explanation

This logistic regression model predicts the likelihood of a shooting incident resulting in murder (`statistical_murder_flag`) based on several predictors: borough (`boro`), time of occurrence (`occur_hour`), victim's age group, sex, and race.

0.6.1 Key Findings

0.6.1.1 Statistically Significant Predictors

- `boroMANHATTAN`: Incidents in Manhattan are significantly less likely to result in murder compared to the baseline borough (likely the Bronx), with a small but statistically significant negative effect ($p = 0.0175$).
- `vic_age_group18-24, 25-44, 45-64, 65+, UNKNOWN`: All these age groups show statistically significant increased odds of the incident being a murder compared to the baseline (likely under 18), with the effect increasing with age.
- `vic_age_group18-24`: OR approximately $\exp(0.26) \rightarrow$ slight increase.
- `vic_age_group25-44 and 45-64`: Higher estimates suggest a stronger relationship.
- `vic_age_group65+`: The strongest effect among the age groups (Estimate approximately 0.99).
- `vic_age_groupUNKNOWN`: Also significant, though likely due to data irregularities or small counts.

0.6.1.2 Non-Significant Predictors

- **Other boroughs (Brooklyn, Queens, Staten Island)**: No significant difference from the baseline.
- `occur_hour`: No significant relationship between time of incident and likelihood of murder.
- `vic_sex`: Neither male (`M`) nor unknown (`U`) differs significantly from the baseline (`F`).
- `vic_race`: All race variables have extremely high standard errors and are not statistically significant. These inflated standard errors suggest **multicollinearity** or **complete/quasi-complete separation** in the data, leading to unreliable coefficient estimates.

0.6.2 Model Fit

- **Null deviance**: 29251 $\rightarrow$ deviance of model with no predictors.
- **Residual deviance**: 28943 $\rightarrow$ deviance after including predictors.
- **AIC**: 28983 $\rightarrow$ a model selection metric; lower is better.
- **Model improvement** is minimal (small drop in deviance), suggesting limited predictive power from the included variables.

0.6.3 Notes

- The very large standard errors and coefficients for race categories imply instability in the model, possibly due to sparse data or perfect prediction for certain groups.
- Only a few variables meaningfully contribute to predicting murder outcomes in the dataset, most notably certain age groups and location in Manhattan.

0.7 5 Personal Bias and Mitigation

My Personal Bias: As an analyst examining crime data, I may carry assumptions about the relationships between demographics, geography, and violence that could influence interpretation. Additionally, focusing on statistical patterns might overlook the human impact and community context of these incidents.

Mitigation Strategies: To address these biases, I have employed transparent methodology, avoided making causal claims about demographic relationships, focused on descriptive rather than prescriptive analysis, and explicitly acknowledged data limitations. The analysis emphasizes temporal and geographic patterns rather than making judgments about individual or group characteristics.

0.8 6 Conclusion

This analysis reveals several important patterns in NYC shooting incidents. The data shows a general decline in shooting incidents from 2011 to 2019, followed by a sharp increase in 2020, suggesting that broader social and economic factors significantly influence gun violence patterns. Geographic analysis indicates that Brooklyn and the Bronx experience the highest absolute numbers of incidents, while temporal analysis shows peak activity during evening and late-night hours.

The analysis is subject to limitations, primarily stemming from potential data biases as discussed. Future work could involve more advanced modeling techniques, deeper investigation into the "UNKNOWN" categories, incorporating external