NYPD Shooting Incident Analysis

Denzil James Greenwood

June 19 2024

```
# This section loads the required libraries for the analysis
knitr::opts_chunk$set(echo = TRUE, tidy.opts = list(width.cutoff = 80), tidy = TRUE)

# Function to check and install missing packages
check_and_install <- function(package) {
  if (!requireNamespace(package, quietly = TRUE)) {
    install.packages(package, dependencies = TRUE)
  }
  library(package, character.only = TRUE)
}

# List of required packages
required_packages <- c(
  "caret", "cluster", "dplyr", "forcats", "ggplot2","ggthemes",
  "glmnet", "gridExtra", "htmltools", "kableExtra", "knitr",
  "leaflet", "lubridate", "purrr", "pROC", "RColorBrewer",
  "readr", "rmarkdown", "ROCR", "sf", "stringr", "tidyverse",
  "tinytex", "viridisLite", "webshot2"
  )


# Check and install missing packages
invisible(lapply(required_packages, check_and_install))
```

Introduction and Background

Analyzing NYPD shooting incidents is crucial for understanding crime patterns and informing public safety strategies. This analysis can provide actionable insights to law enforcement and policymakers to address crime hot spots, allocate resources effectively, and develop targeted interventions to reduce gun violence.

Objective of the Analysis

Our analysis aims to evaluate the usefulness of the NYPD shooting incident data since 2006 and to provide actionable insights and recommendations for improving this process in the future.

NYPD Shooting Incident Data (Historic)

Metadata Updated: April 26, 2024 This dataset contains records of every shooting incident that occurred in NYC from 2006 through the end of the previous calendar year. The data is manually extracted

quarterly and reviewed by the Office of Management Analysis and Planning before being posted on the NYPD website. Each record includes information about the event, location, time of occurrence, and suspect and victim demographics. This dataset can be used by the public to explore the nature of shooting/criminal activity.

Source: NYPD Shooting Incident Data (Historic) https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

```
# Load the NYPD Shooting Incident data from the URL
# creates global variables for the data and column names

url_in = "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
NYPD_data = read_csv(url_in)


# Add global variables
# List of columns to analyze
COLUMNS_TO_ANALYZE <- c("LOC_CLASSFCTN_DESC", "LOC_OF_OCCUR_DESC", "LOCATION_DESC",
↪  "PERP_AGE_GROUP",
                        "PERP_SEX", "PERP_RACE", "VIC_AGE_GROUP", "VIC_SEX", "VIC_RACE")

# list of column names to be displayed in the title
COLUMN_NAMES <- c("Classification Description", "Location of occurrence description",
↪  "Location Description", "Perpetrator by Age Group",
                  "Perpetrator by Sex", "Perpetrator by Racial Group", "Victim by Age
                  ↪  Group", "Victim by Sex", "Victim by Racial Group")
```

NYPD Shooting Incident Data Sample The following is the first 10 lines of the NYPD Shooting Incident dataset.

- We will download the information from the website here: https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8/about_data.

```
# This section displays the first 10 lines of the NYPD Shooting Incident data set

# Create the HTML table
html_table <- kable(head(NYPD_data, 10), format = "html", table.attr = 'class="table
↪  table-stripped table-bordered table-hover"') %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive")) %>%
  scroll_box(height = "450px", width = "100%") %>%
  add_header_above(c(" " = 1, "NYPD Data" = ncol(NYPD_data) - 1))

# Create the collapsible and scrollable div with sticky header
html_output <- HTML(
  '<details>
    <summary>Click to display the first ten lines of the dataset.</summary>',
    html_table,
  '</details>'
)

HTML(html_output)
```

Column Names, Descriptions, Types of data and other related information:

NYPD Shooting Incident Level Data Footnotes

1. Information is accurate as of the date it was queried from the system of record, but should be considered a close approximation of current records, due to revisions and updates.
2. Data is available as of the date that technological enhancements to information systems allowed for data capture. Null values appearing frequently in certain fields may be attributed to changes on official department forms where data was previously not collected. Null values may also appear in instances where information was not available or unknown at the time of the report and should be considered as either "Unknown/Not Available/Not Reported."
3. A shooting incident can have multiple victims involved and as a result duplicate INCIDENT_KEY's are produced. Each INCIDENT_KEY represents a victim but similar duplicate keys are counted as one incident.
4. Shooting incidents occurring near an intersection are represented by the X coordinate and Y coordinates of the intersection Shooting incidents occurring anywhere other than at an intersection are geo-located to the middle of the nearest street segment where appropriate.
5. Any attempt to match the approximate location of the incident to an exact address or link to other datasets is not recommended.
6. Many other shooting incidents that were not able to be geo-coded (for example, due to an invalid address) have been located as occurring at the police station house within the precinct of occurrence.
7. Shooting incidents occurring in open areas such as parks or beaches may be geo-coded as occurring on streets or intersections bordering the area.
8. Shooting incidents occurring on a moving train on transit systems are geo-coded as occurring at the train's next stop.
9. All shooting incidents occurring within the jurisdiction of the Department of Correction have been geo-coded as occurring on Riker's Island.
10. X and Y Coordinates are in NAD 1983 State Plane New York Long Island Zone Feet (EPSG 2263).
11. Latitude and Longitude Coordinates are provided in Global Coordinate System WGS 1984 decimal degrees (EPSG 4326).
12. Errors in data transcription may result in nominal data inconsistencies.
13. The CSV file should be opened using an appropriate tool for data exploration, e.g. SPSS, SAS, Tableau, etc. If using MS Excel, be sure to use the tools for importing external data, otherwise inconsistencies may occur when viewing the data.
14. Only valid shooting incidents resulting in an injured victim are included in this release. Shooting incidents not resulting in an injured victim are classified according to the appropriate offense according to NYS Penal Law.

- Note: The previous information is provided from the following website. https://data.cityofnewyork.us/api/views/833y-fsy8/files/e4e3d86c-348f-4a16-a17f-19480c089429?download=true&filename=NYPD_Shootings_Incident_Level_Data_Footnotes.pdf

15. Field Names and Descriptions are as follows:

Column Name	Description	Type
INCIDENT_KEY	Randomly generated persistent ID for each arrest	Plain Text
OCCUR_DATE	Exact date of the shooting incident	Date & Time
OCCUR_TIME	Exact time of the shooting incident	Plain Text
BORO	Borough where the shooting incident occurred	Plain Text
LOC_OF_OCCUR_DESC	Description of the location of occurrence	Plain Text
PRECINCT	Precinct where the shooting incident occurred	Number
JURISDICTION_CODE	Jurisdiction where the shooting incident occurred. Jurisdiction codes 0(Patrol), 1(Transit), and 2(Housing) represent NYPDwhilst codes 3 and more represent non-NYPD jurisdictions	Number
LOC_CLASSFCTN_DESC	Location classification description	Plain Text
LOCATION_DESC	Location of the shooting incident	Plain Text
STATISTICAL_MURDER_FLAG	Shooting resulted in the victim's death which would be counted as a murder	Checkbox
PERP_AGE_GROUP	Perpetrator's age within a category	Plain Text
PERP_SEX	Perpetrator's sex description	Plain Text
PERP_RACE	Perpetrator's race description	Plain Text
VIC_AGE_GROUP	Victim's age within a category	Plain Text
VIC_SEX	Victim's sex description	Plain Text
VIC_RACE	Victim's race description	Plain Text
X_COORD_CD	Midblock X-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet(FIPS 3104)	Plain Text
Y_COORD_CD	Midblock Y-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet(FIPS 3104)	Plain Text
Latitude	Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees(EPSG 4326)	Number
Longitude	Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees(EPSG 4326)	Number
Lon_Lat	Longitude and Latitude Coordinates for mapping	Point

- Note: The table name, description and data types are provided at the following website. https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8/about_data.

Summary Information by Column

Each of the columns listed below has a large percentage of NA/null values. The main focus of this analysis is to identify causes and suggest improvements in the data collection process to reduce the number of missing values.

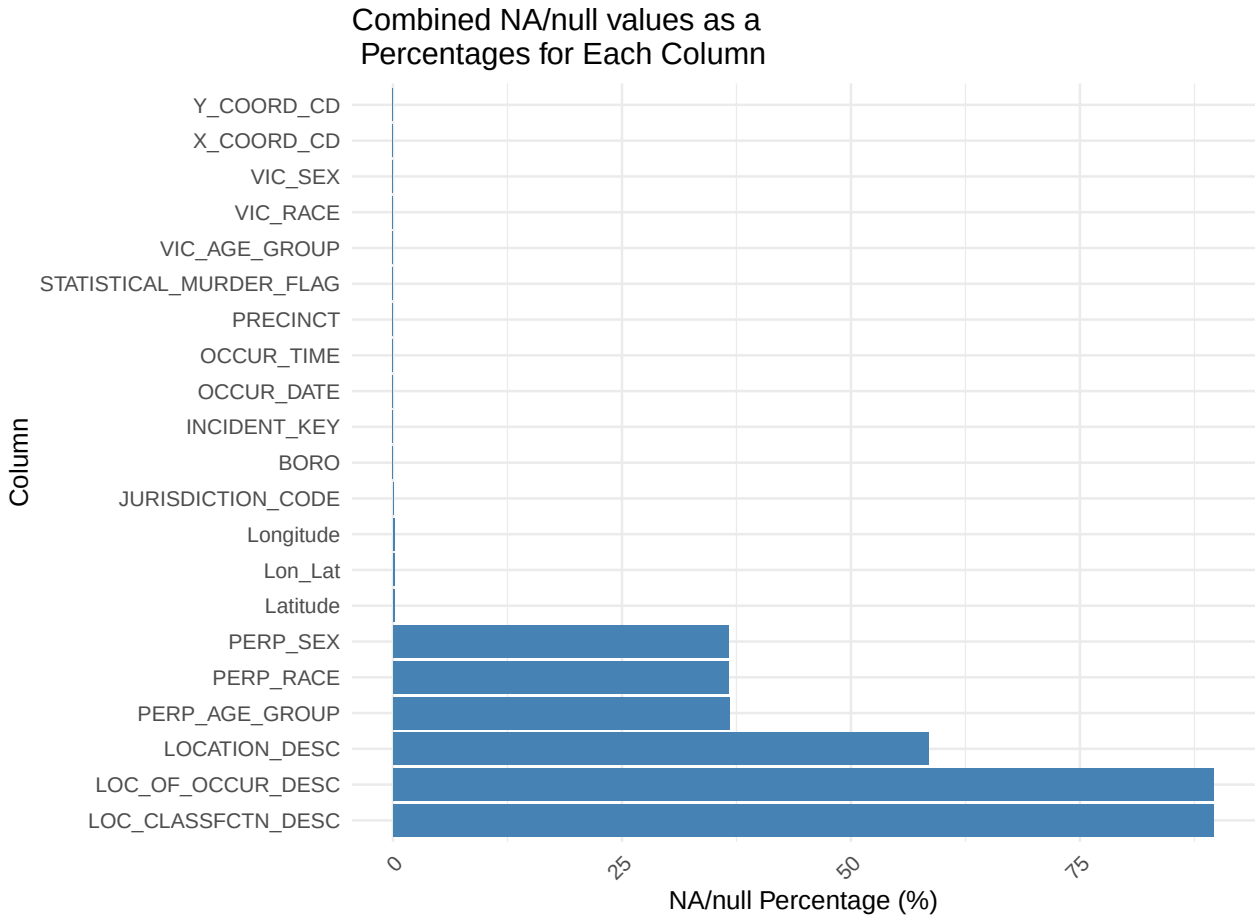
Columns are analyzed in this section for NA and null values. The column names are unchanged from the original dataset, but the descriptions are provided in the title section as follows:

- LOC_CLASSFCTN_DESC: Classification of the location of the incident.
- LOCATION_DESC: Description of the location of the incident.
- PERP_AGE_GROUP: Age group of the perpetrator.
- PERP_SEX: Sex of the perpetrator (M-male/F-female/U-unknown).
- PERP_RACE: Perpetrator by racial group.
- VIC_AGE_GROUP: Age group of the victim.
- VIC_SEX: Sex of the victim (M-male/F-female/U-unknown).
- VIC_RACE: Victim by racial group.

Percentages of NA/Null Values for each column. The following is a summary of the percentage of NA and null values for each of the columns listed above. By analyzing these percentages, we can identify potential issues with data collection and quality. Most of the missing data is from the perpetrators. A large number of NA and null values also show up in the Location Classification, Location Description, and the Location of Occurrence columns.

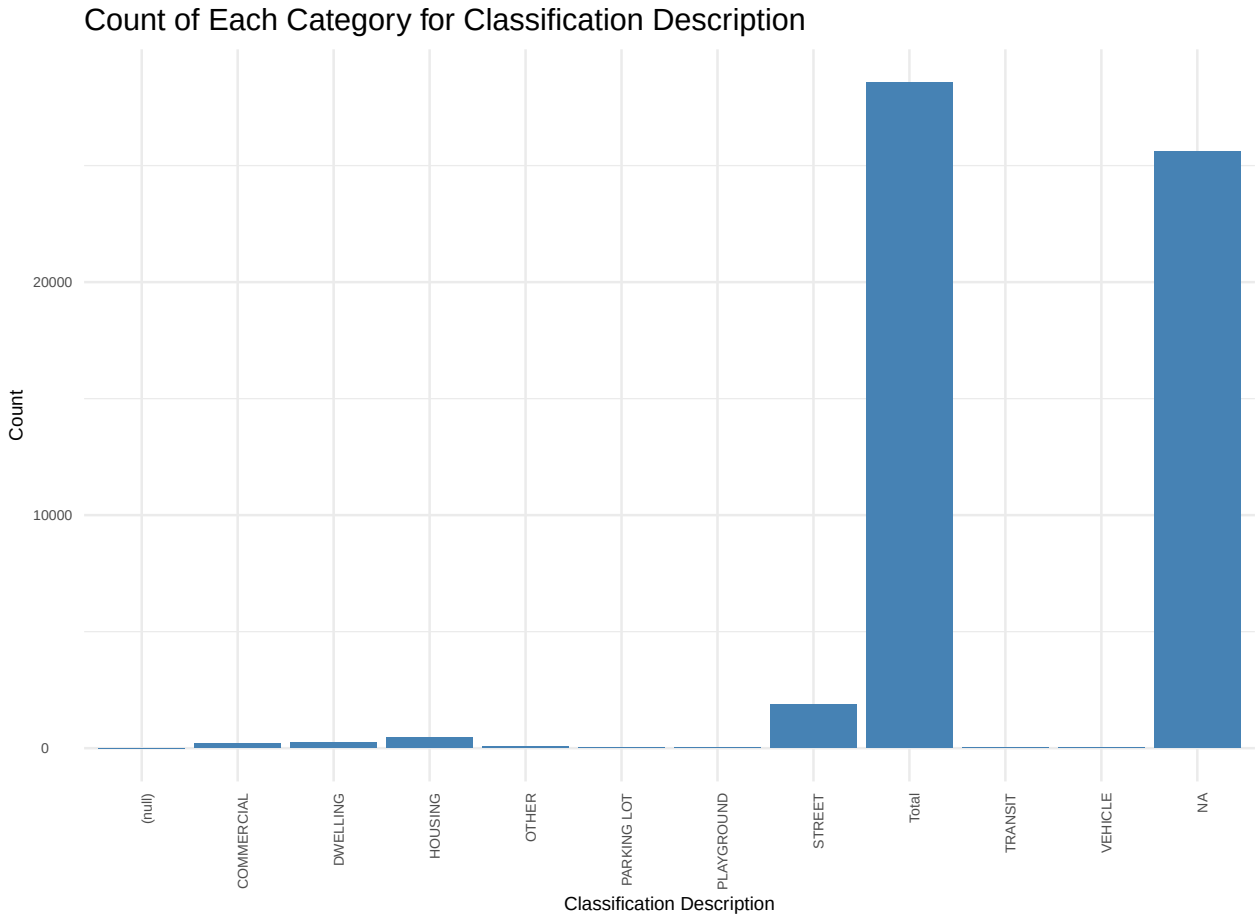
```
# Define a function to calculate category counts and percentages for a given column
calculate_category_percentages <- function(data, column_name) {
  category_counts <- data %>%
    group_by(across(all_of(column_name))) %>%
    summarize(count = n(), .groups = 'drop') %>%
    mutate(percentage = (count / nrow(data)) * 100)

  total_row <- data.frame(column_value = "Total", count = nrow(data), percentage = 100)
  colnames(total_row)[1] <- column_name

  bind_rows(category_counts, total_row)
}

# Define a function to calculate combined NA percentages for each column
calculate_combined_na_percentages <- function(data) {
  # Define a helper function to check for NA-like values
  is_na_like <- function(x) {
    is.na(x) | x %in% c("NA", "null", "NULL", "(null)", "")
  }

  # Calculate the percentage of combined NA-like values for each column
  na_percentages <- data %>%
    summarize(across(everything(), ~ mean(is_na_like(.)) * 100)) %>%
    pivot_longer(cols = everything(), names_to = "Column", values_to = "NA_Percentage")

  na_percentages <- na_percentages %>%
    mutate(NA_Percentage = round(NA_Percentage, 2))

  na_percentages
}

# Apply the function to the data frame to calculate combined NA percentages
na_percentages <- calculate_combined_na_percentages(NYPD_data)

# Plot the NA percentages
na_percentages_plot <- na_percentages %>%
  ggplot(aes(x = reorder(Column, -NA_Percentage), y = NA_Percentage)) +
  geom_bar(stat = "identity", fill = "steelblue") +
```

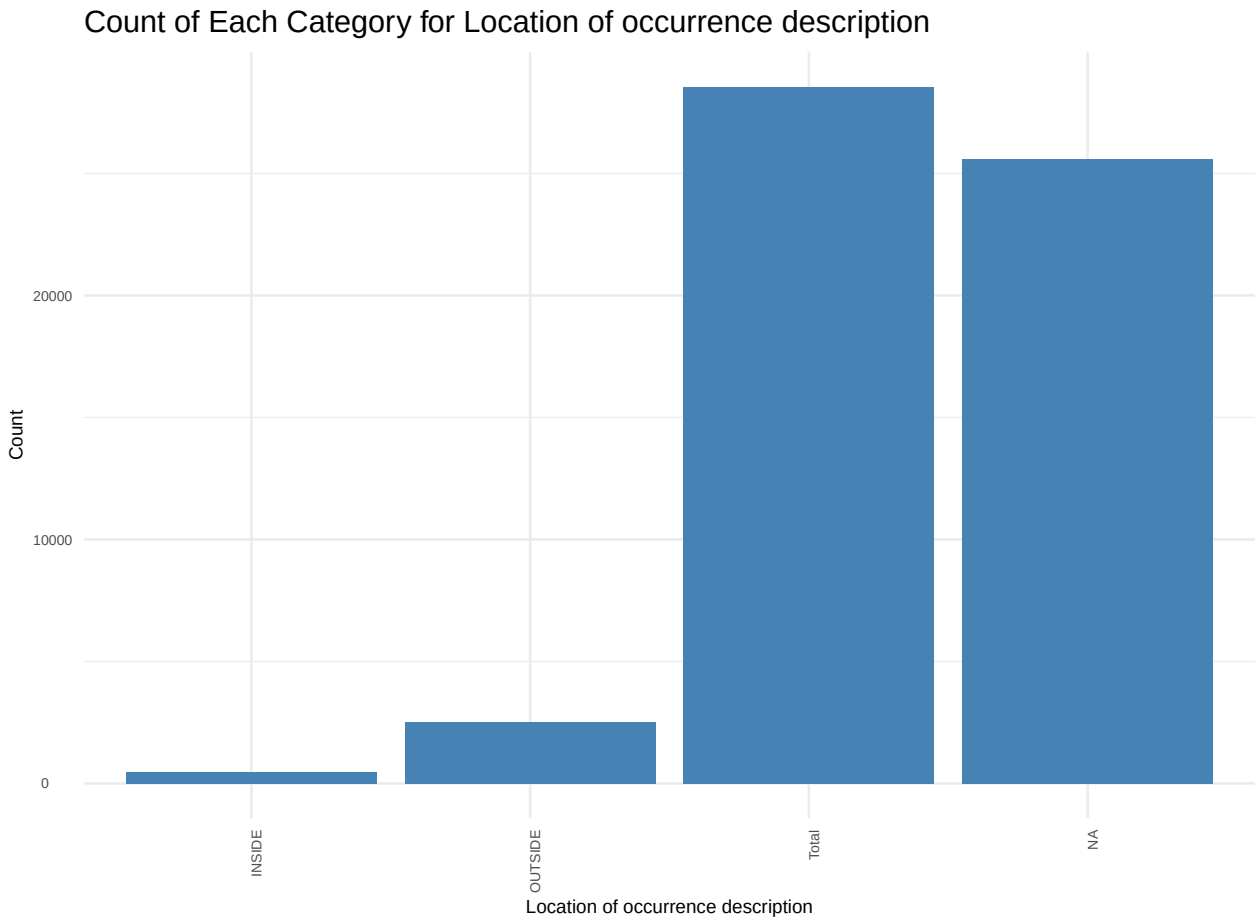
```
  coord_flip() +
  labs(title = "Combined NA/null values as a\n Percentages for Each Column", x =
  ↪  "Column", y = "NA/null Percentage (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Print the plot
print(na_percentages_plot)
```

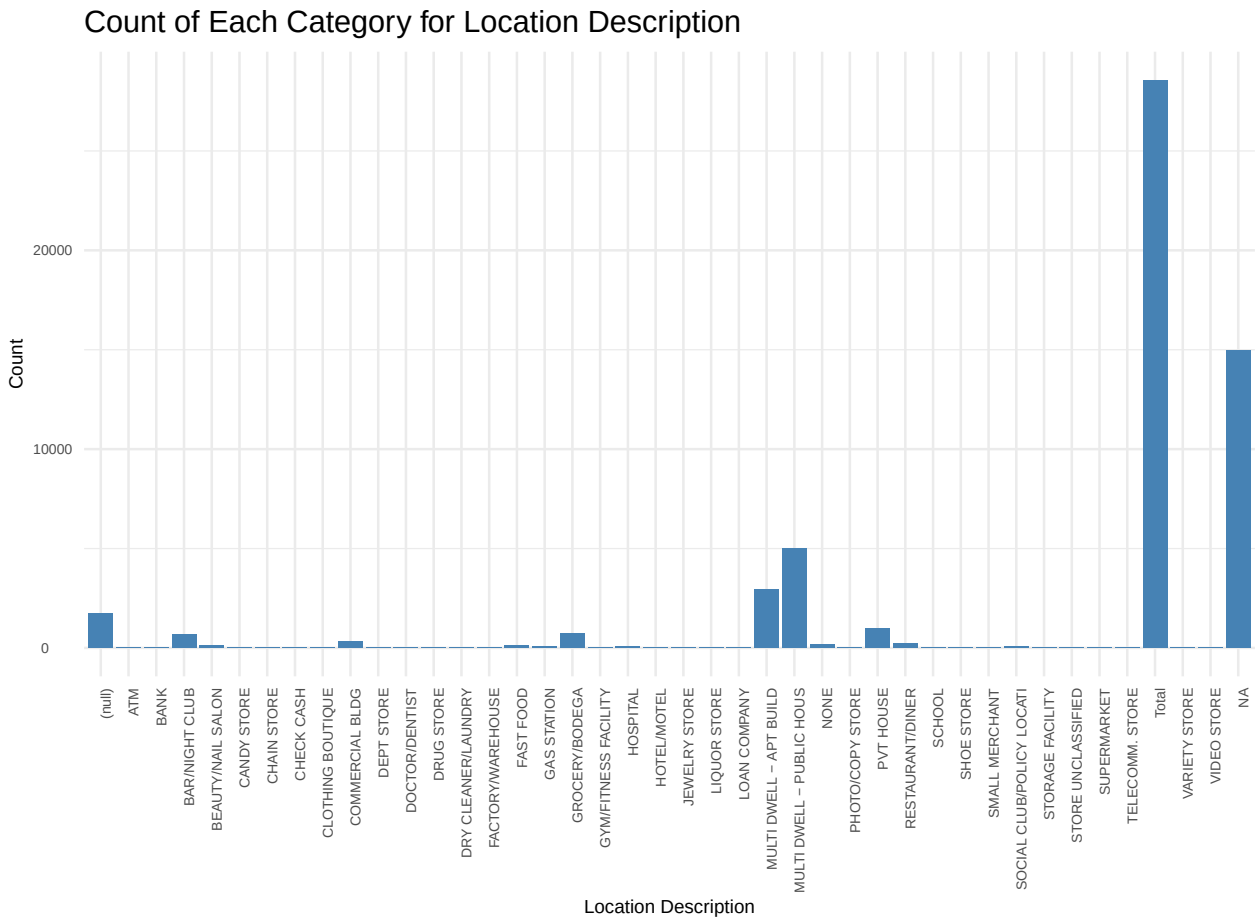


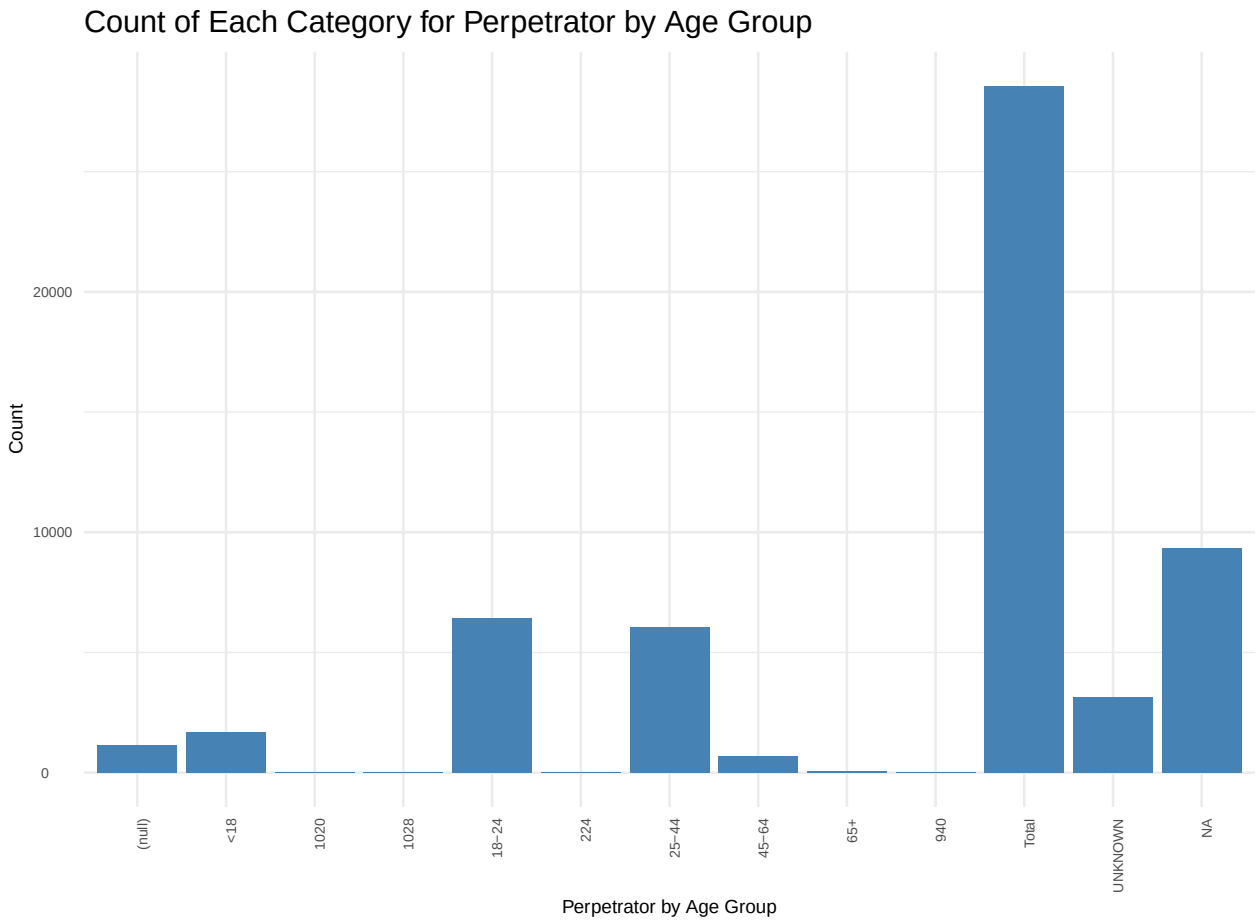

```
# Create bar charts for each column
plot_list <- lapply(seq_along(COLUMNS_TO_ANALYZE), function(i) {
  col <- COLUMNS_TO_ANALYZE[i]
  col_name <- COLUMN_NAMES[i]

  category_percentages <- calculate_category_percentages(NYPD_data, col)

  ggplot(category_percentages, aes_string(x = col, y = "count")) +
    geom_bar(stat = "identity", fill = "steelblue") +
    labs(title = paste("Count of Each Category for", col_name), x = col_name, y =
    ↪  "Count") +
    theme_minimal() +
    theme(axis.title = element_text(size = 8),
          axis.text.y = element_text(size = 6, angle = 0, hjust = 0.25),
```

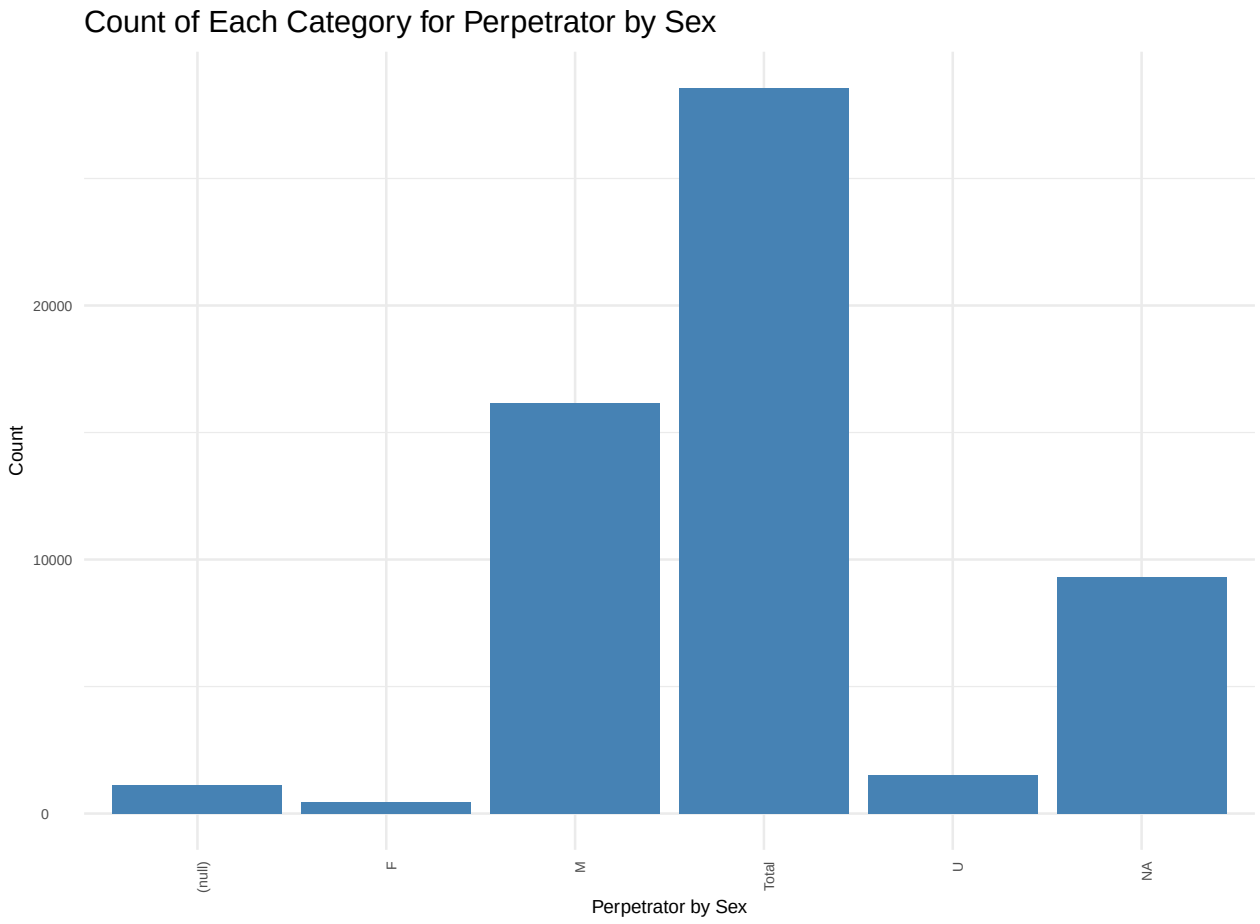
```
          axis.text.x = element_text(size = 6, angle = 90, hjust = 1))
    })

# Print plots
for (plot in plot_list) {
  print(plot)
}
```

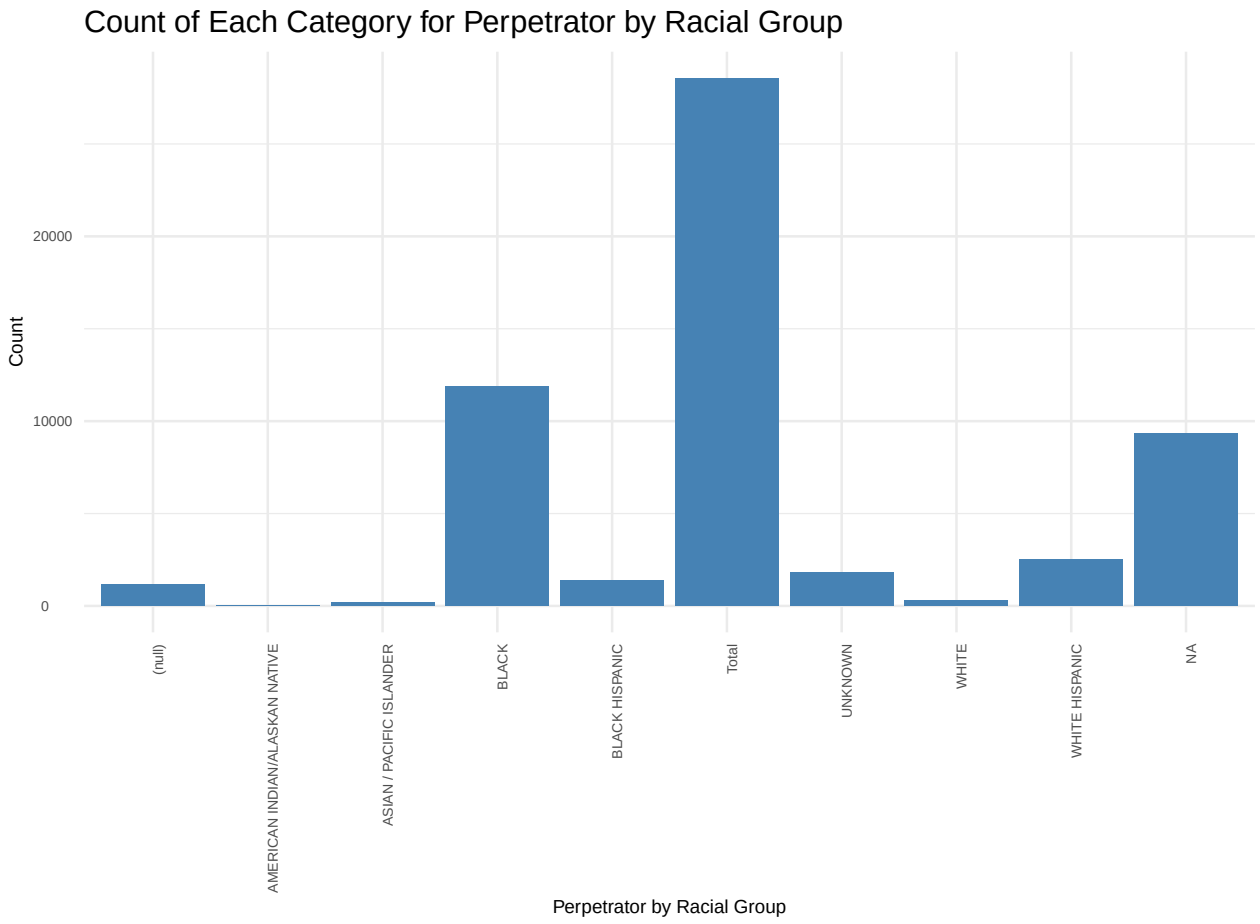


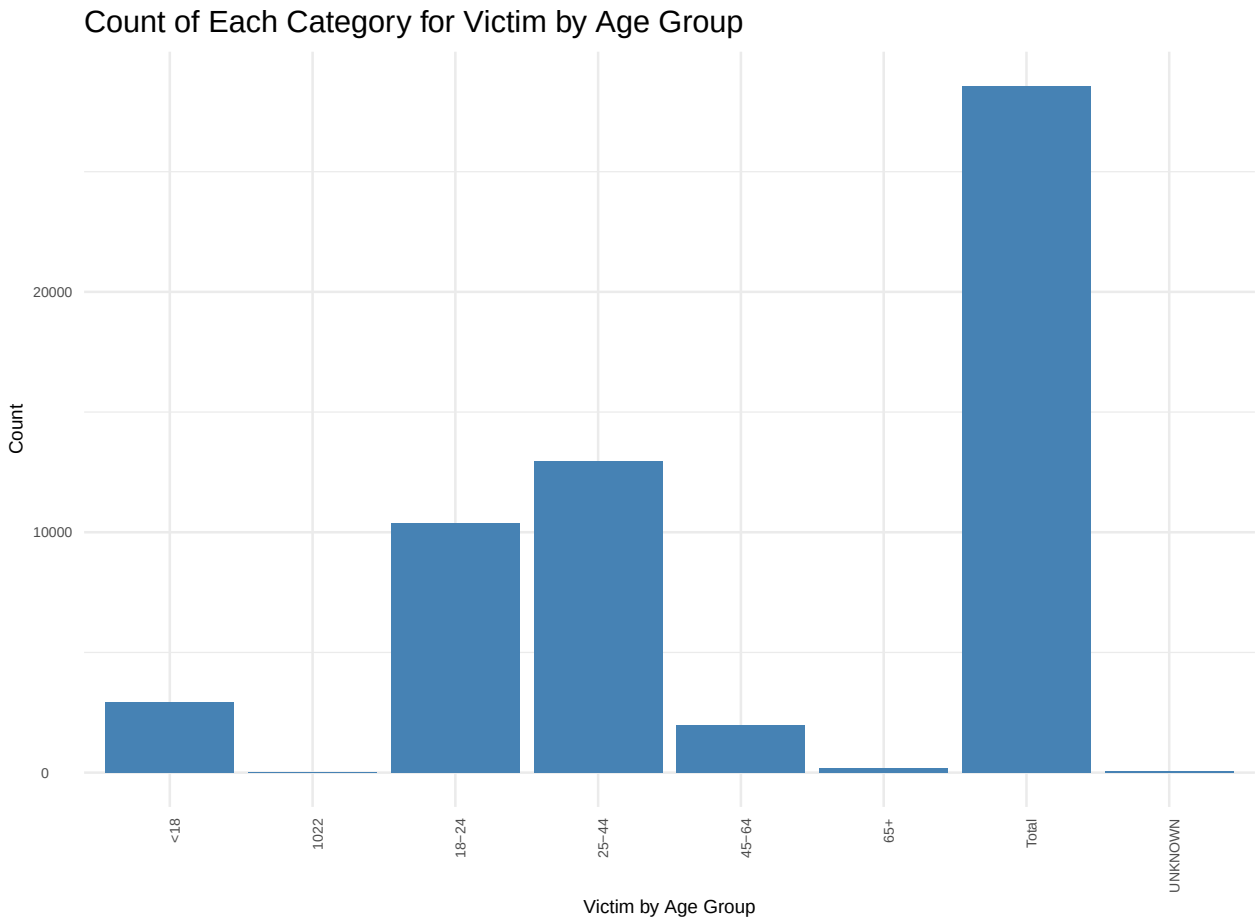

Count of Each Category for Location of occurrence description
Count
20000
10000
0
INSIDE
OUTSIDE
Total
NA
Location of occurrence description

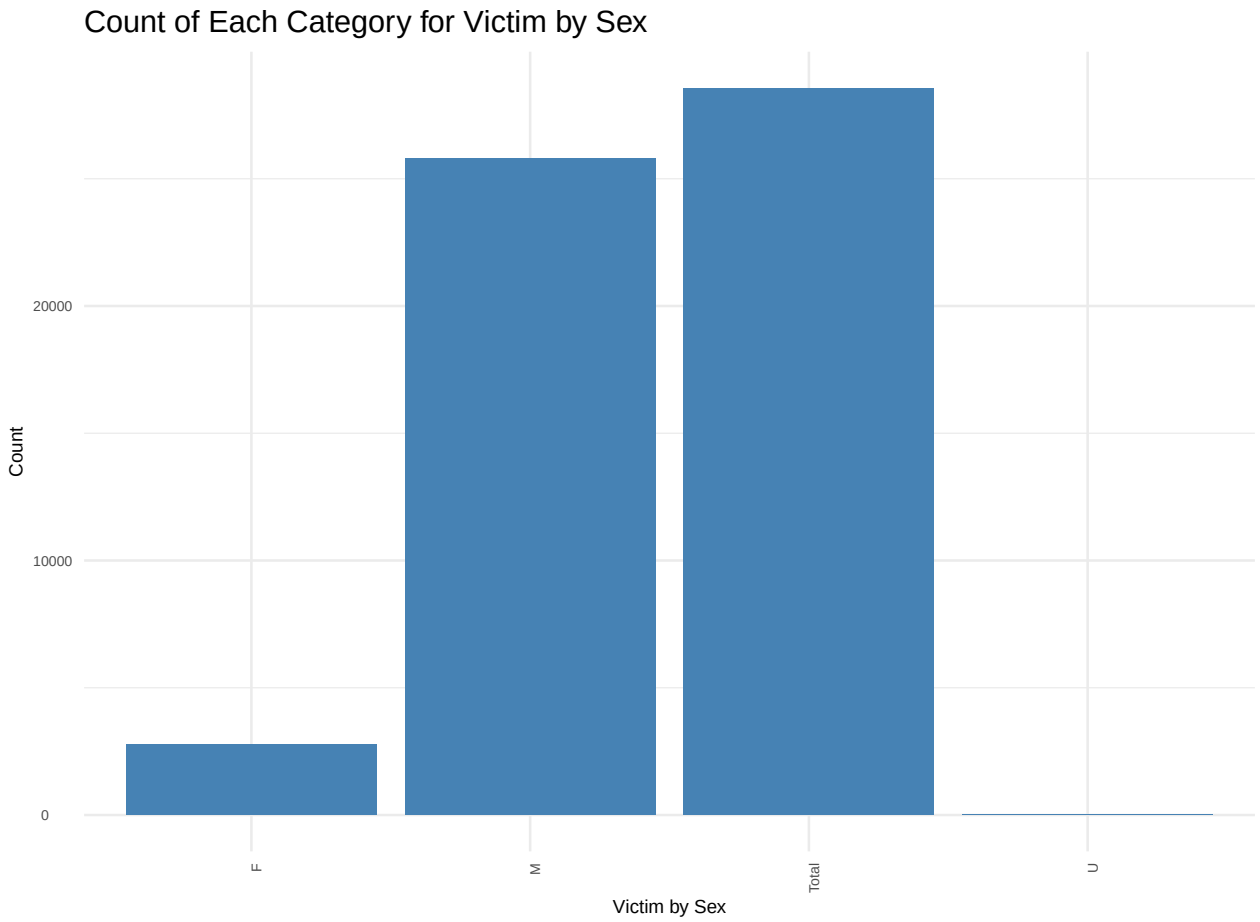

Count of Each Category for Location Description
Count
20000
10000
0
(null)
ATM
BANK
BAR/NIGHT CLUB
BEAUTY/NAIL SALON
CANDY STORE
CHAIN STORE
CHECK CASH
CLOTHING BOUTIQUE
COMMERCIAL BLDG
DEPT STORE
DOCTOR/DENTIST
DRUG STORE
DRY CLEANER/LAUNDRY
FACTORY/WAREHOUSE
FAST FOOD
GAS STATION
GROCERY/BODEGA
GYM/FITNESS FACILITY
HOSPITAL
HOTEL/MOTEL
JEWELRY STORE
LIQUOR STORE
LOAN COMPANY
MULTI DWELL – APT BUILD
MULTI DWELL – PUBLIC HOUS
NONE
PHOTO/COPY STORE
PVT HOUSE
RESTAURANT/DINER
SCHOOL
SHOE STORE
SMALL MERCHANT
SOCIAL CLUB/POLICY LOCATI
STORAGE FACILITY
STORE UNCLASSIFIED
SUPERMARKET
TELECOMM. STORE
Total
VARIETY STORE
VIDEO STORE
NA
Location Description

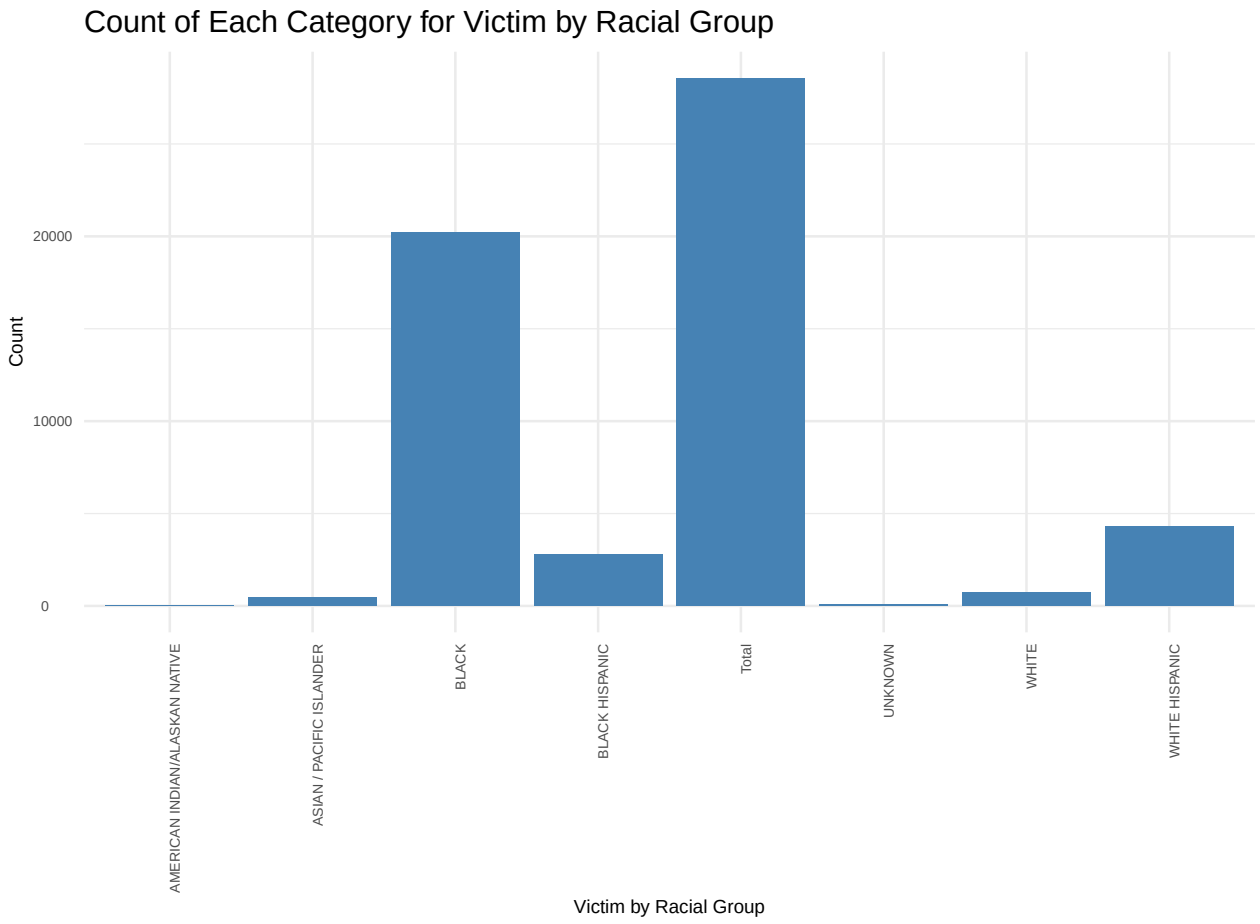

Count of Each Category for Perpetrator by Age Group
Count
20000
10000
0
(null)
<18
1020
1028
18–24
224
25–44
45–64
65+
940
Total
UNKNOWN
NA
Perpetrator by Age Group


Count of Each Category for Perpetrator by Sex
Count
20000
10000
0
(null)
F
M
Total
U
NA
Perpetrator by Sex


Count of Each Category for Perpetrator by Racial Group
Count
20000
10000
0
(null)
AMERICAN INDIAN/ALASKAN NATIVE
ASIAN / PACIFIC ISLANDER
BLACK
BLACK HISPANIC
Total
UNKNOWN
WHITE
WHITE HISPANIC
NA
Perpetrator by Racial Group


Count of Each Category for Victim by Age Group
Count
20000
10000
0
<18
1022
18–24
25–44
45–64
65+
Total
UNKNOWN
Victim by Age Group


Count of Each Category for Victim by Sex
Count
20000
10000
0
F
M
Total
U
Victim by Sex



Data Cleaning and Transformation

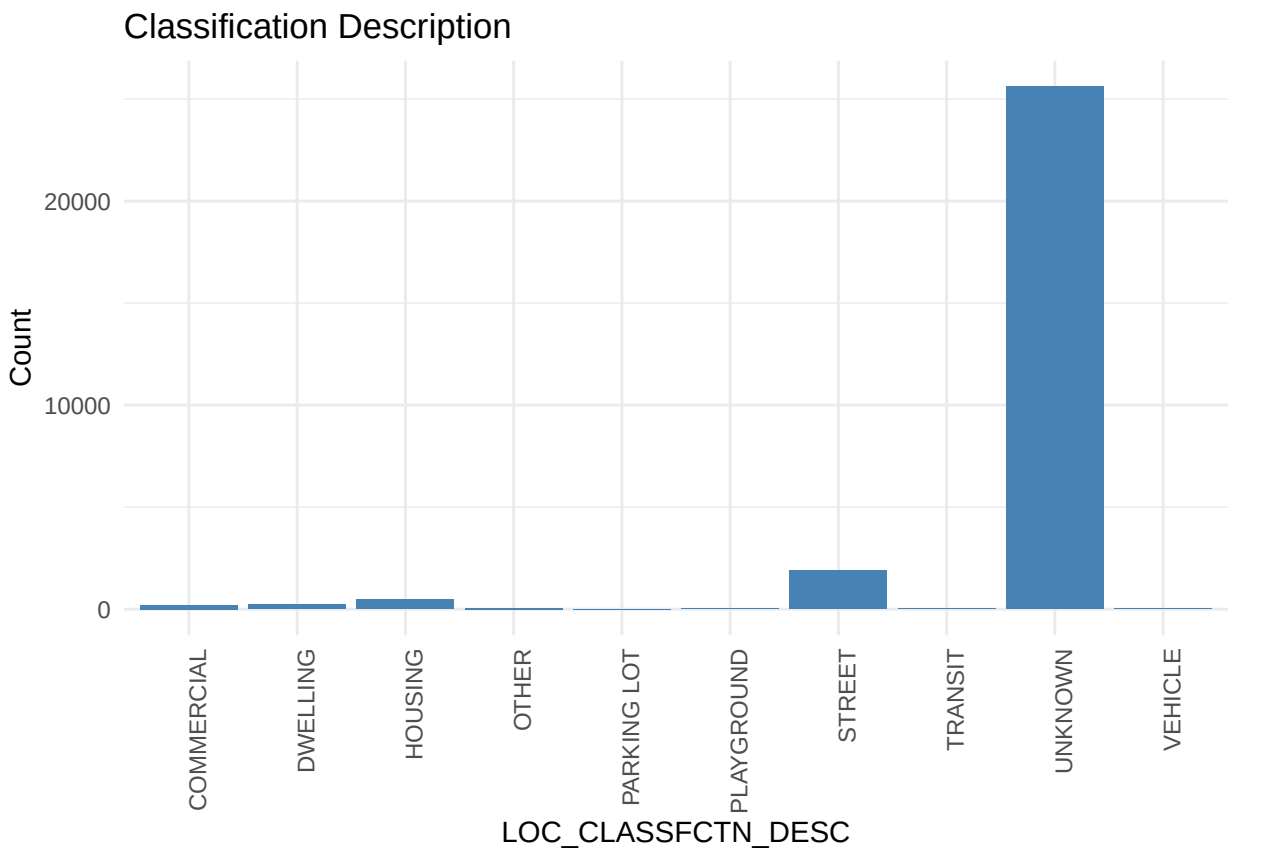
The following steps outline how the data is cleaned and transformed for analysis.

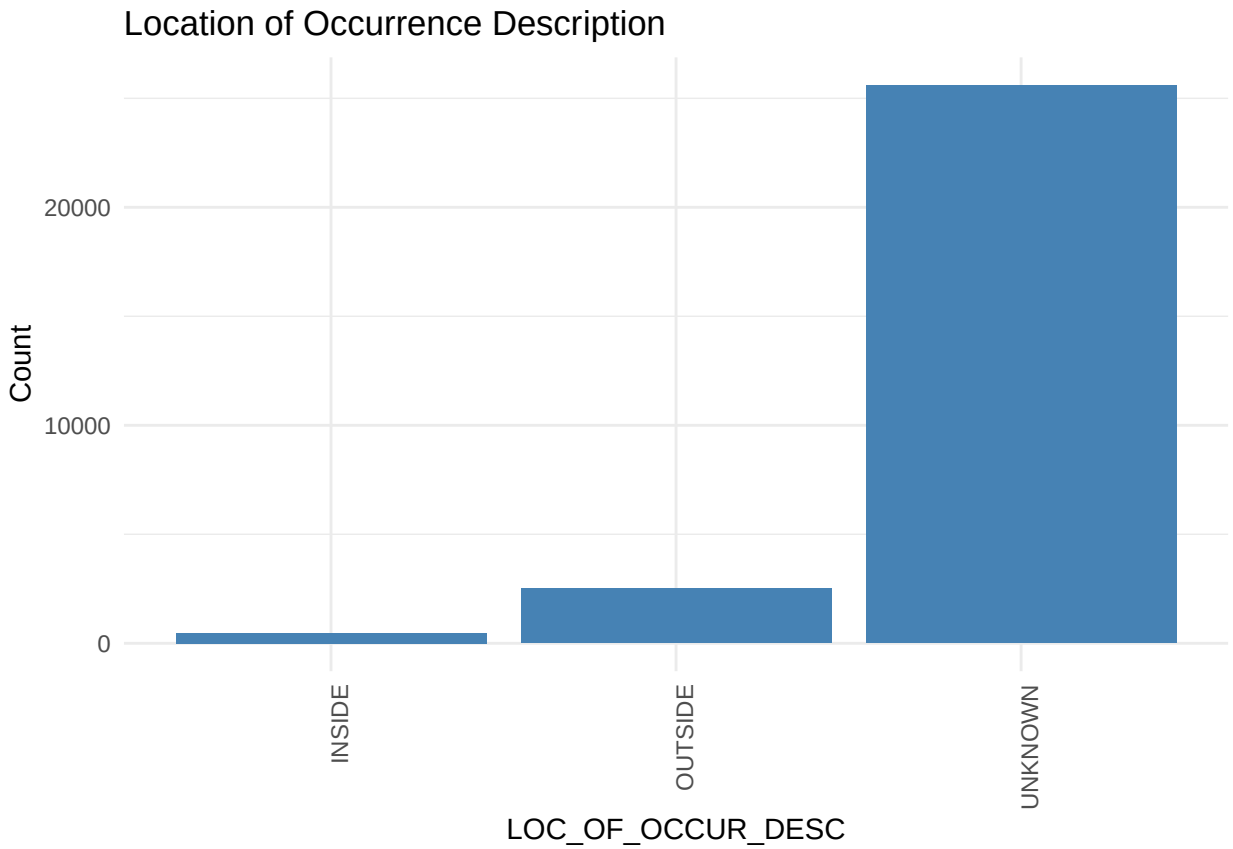
```
# Clean the data by filling in missing values in LOC_CLASSFCTN_DESC, LOC_OF_OCCUR_DESC,
↪  and LOCATION_DESC
NYPD_data_cleaned <- NYPD_data

# Convert OCCUR_TIME to total seconds past midnight
NYPD_data_cleaned$OCCUR_TIME_SECONDS <- as.numeric(substr(NYPD_data_cleaned$OCCUR_TIME,
↪  1, 2)) * 3600 +
                                        as.numeric(substr(NYPD_data_cleaned$OCCUR_TIME,
                                        ↪  4, 5)) * 60 +
                                        as.numeric(substr(NYPD_data_cleaned$OCCUR_TIME,
                                        ↪  7, 8))

# Calculate the mode of the LOC_OF_OCCUR_DESC column
calculate_mode <- function(x) {
  ux <- unique(na.omit(x))
  ux[which.max(tabulate(match(x, ux)))]
}

# Define a helper function to check for NA-like values
```

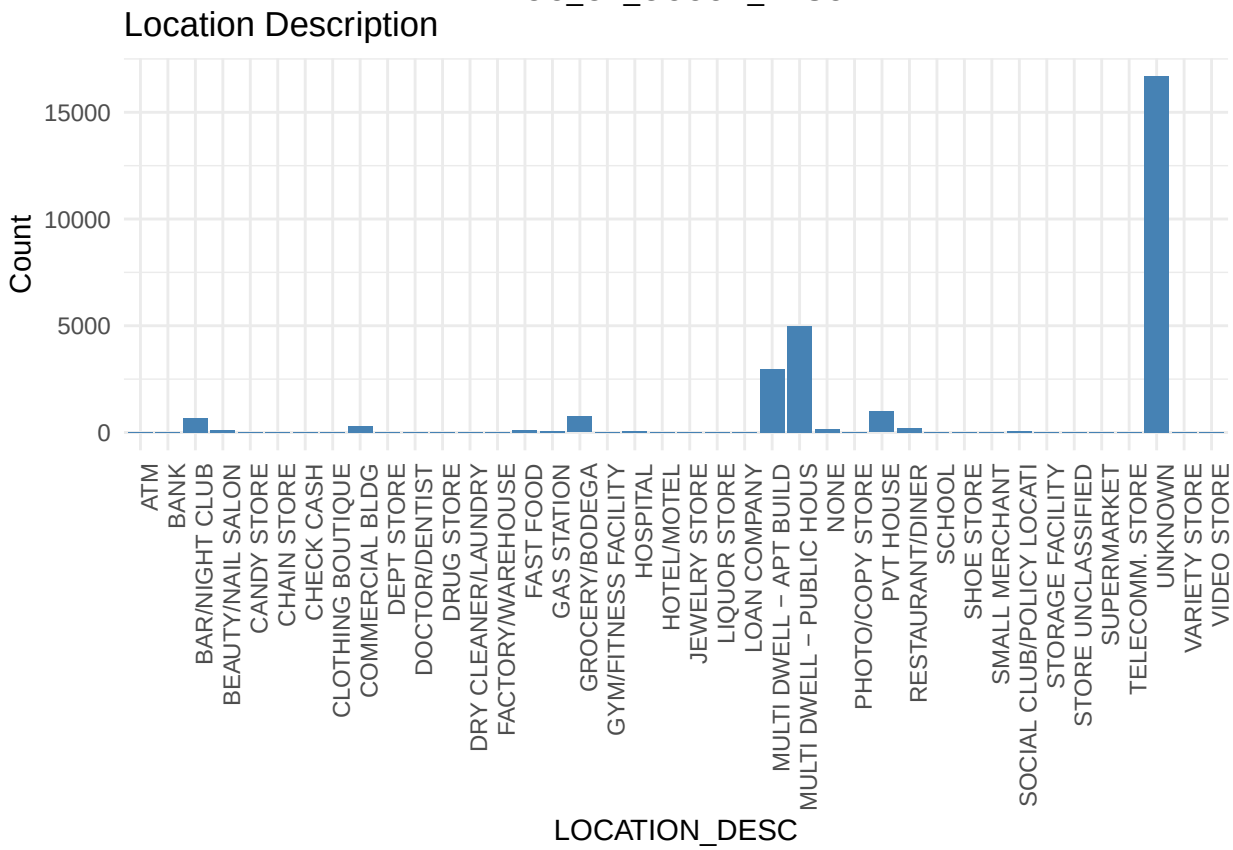
```r
is_na_like <- function(x) {
  is.na(x) | x %in% c("na", "NA", "(null)", "null", "")
}

# Fill missing values in LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, and LOCATION_DESC
NYPD_data_cleaned <- NYPD_data %>%
  mutate(
    LOC_CLASSFCTN_DESC = if_else(is_na_like(LOC_CLASSFCTN_DESC), 'UNKNOWN',
    ↪  LOC_CLASSFCTN_DESC),
    LOC_OF_OCCUR_DESC = if_else(is_na_like(LOC_OF_OCCUR_DESC), 'UNKNOWN',
    ↪  LOC_OF_OCCUR_DESC),
    LOCATION_DESC = if_else(is_na_like(LOCATION_DESC), 'UNKNOWN', LOCATION_DESC)
  )

# Convert PRECINCT and JURISDICTION_CODE to numeric
NYPD_data_cleaned <- NYPD_data_cleaned %>%
  mutate(
    PRECINCT = as.numeric(PRECINCT),
    JURISDICTION_CODE = as.numeric(JURISDICTION_CODE)
  )

# Replace NA, "na", and "NULL" values with "UNKNOWN" in PERP_SEX, VIC_SEX column
NYPD_data_cleaned <- NYPD_data_cleaned %>%
  mutate(
      PERP_SEX = case_when(
      is.na(PERP_SEX) ~ "UNKNOWN",
      PERP_SEX %in% c("na", "NA", "NULL", "(null)", "null", "U") ~ "UNKNOWN",
      TRUE ~ PERP_SEX
    ),
    VIC_SEX = case_when(
      is.na(VIC_SEX) ~ "UNKNOWN",
      VIC_SEX %in% c("na", "NA", "NULL", "(null)", "null", "U") ~ "UNKNOWN",
      TRUE ~ VIC_SEX
    )
  )

# Replace NA, "na", and "NULL" values with "UNKNOWN" in PERP_AGE_GROUP, VIC_AGE_GROUP
↪  columns
NYPD_data_cleaned <- NYPD_data_cleaned %>%
  mutate(
    PERP_AGE_GROUP = case_when(
      is.na(PERP_AGE_GROUP) ~ "UNKNOWN",
      PERP_AGE_GROUP %in% c("na", "NA", "NULL", "(null)", "null") ~ "UNKNOWN",
      TRUE ~ PERP_AGE_GROUP
    ),
    VIC_AGE_GROUP = case_when(
      is.na(VIC_AGE_GROUP) ~ "UNKNOWN",
      VIC_AGE_GROUP %in% c("na", "NA", "NULL", "(null)", "null") ~ "UNKNOWN",
      TRUE ~ VIC_AGE_GROUP
    )
  )

# Replace NA, "na", and "NULL" values with "UNKNOWN" in PERP_RACE, VIC_RACE columns
```

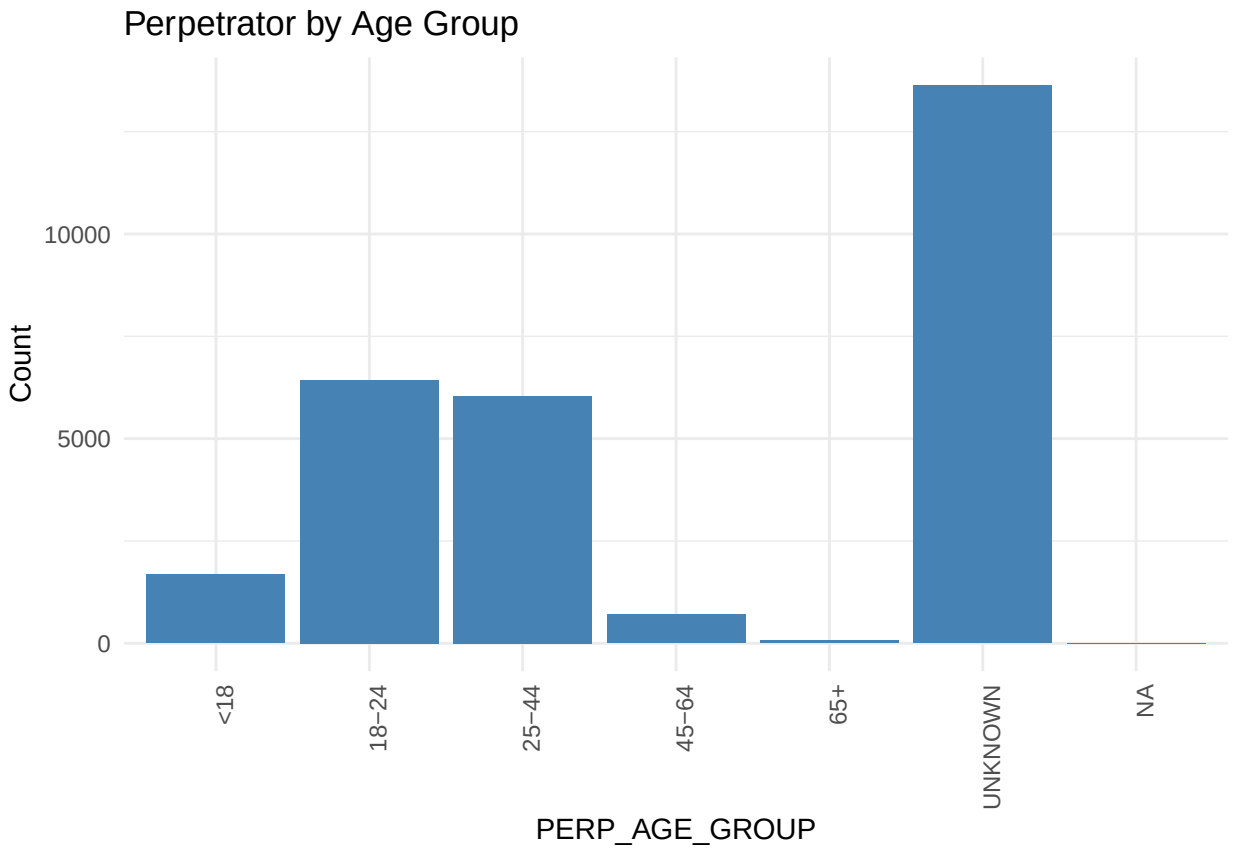
```r
NYPD_data_cleaned <- NYPD_data_cleaned %>%
  mutate(
    PERP_RACE = case_when(
      is.na(PERP_RACE) ~ "UNKNOWN",
      PERP_RACE %in% c("na", "NA", "NULL", "(null)", "null") ~ "UNKNOWN",
      TRUE ~ PERP_RACE
    ),
    VIC_RACE = case_when(
      is.na(VIC_RACE) ~ "UNKNOWN",
      VIC_RACE %in% c("na", "NA", "NULL", "(null)", "null") ~ "UNKNOWN",
      TRUE ~ VIC_RACE
    )
  )

# Set the factor levels for PERP_AGE_GROUP
NYPD_data_cleaned$PERP_AGE_GROUP <- factor(NYPD_data_cleaned$PERP_AGE_GROUP,
                                           levels = c("<18", "18-24", "25-44", "45-64",
                                           ↪  "65+", "UNKNOWN", "NA"))

# Set the factor levels for VIC_AGE_GROUP
NYPD_data_cleaned$VIC_AGE_GROUP <- factor(NYPD_data_cleaned$VIC_AGE_GROUP,
                                          levels = c("<18", "18-24", "25-44", "45-64",
                                          ↪  "65+", "UNKNOWN", "NA"))

# Define the titles for each column
titles <- c(
  "Classification Description",
  "Location of Occurrence Description",
  "Location Description",
  "Perpetrator by Age Group",
  "Perpetrator by Sex",
  "Perpetrator by Racial Group",
  "Victim by Age Group",
  "Victim by Sex",
  "Victim by Racial Group"
)

# Function to plot bar charts
plot_bar_chart <- function(data, column, title) {
  ggplot(data, aes_string(x = column)) +
    geom_bar(fill = "steelblue") +
    labs(title = title, x = column, y = "Count") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
}

plots <- lapply(seq_along(COLUMNS_TO_ANALYZE), function(i) {
  plot_bar_chart(NYPD_data_cleaned, COLUMNS_TO_ANALYZE[i], titles[i])
})

# Print plots
for (plot in plots) {
  print(plot)
```

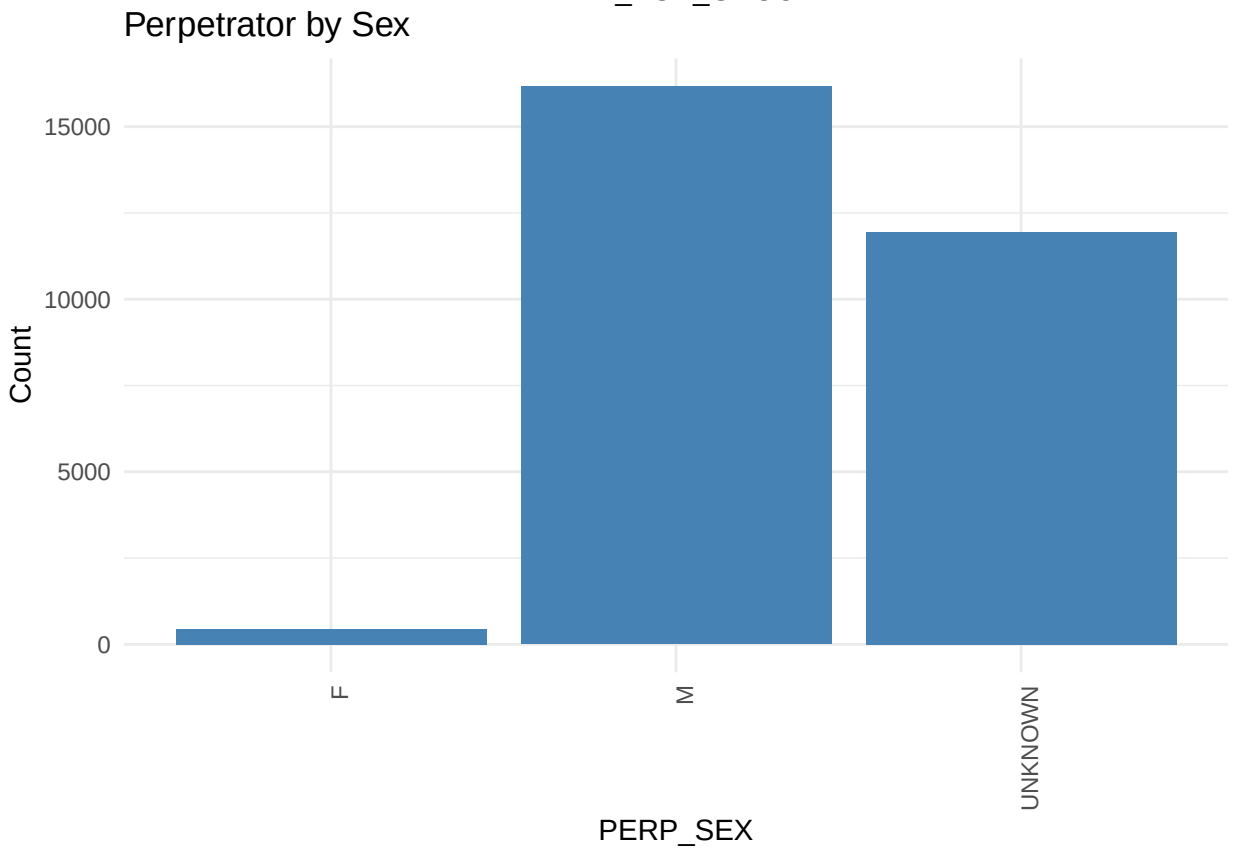
```
}
```

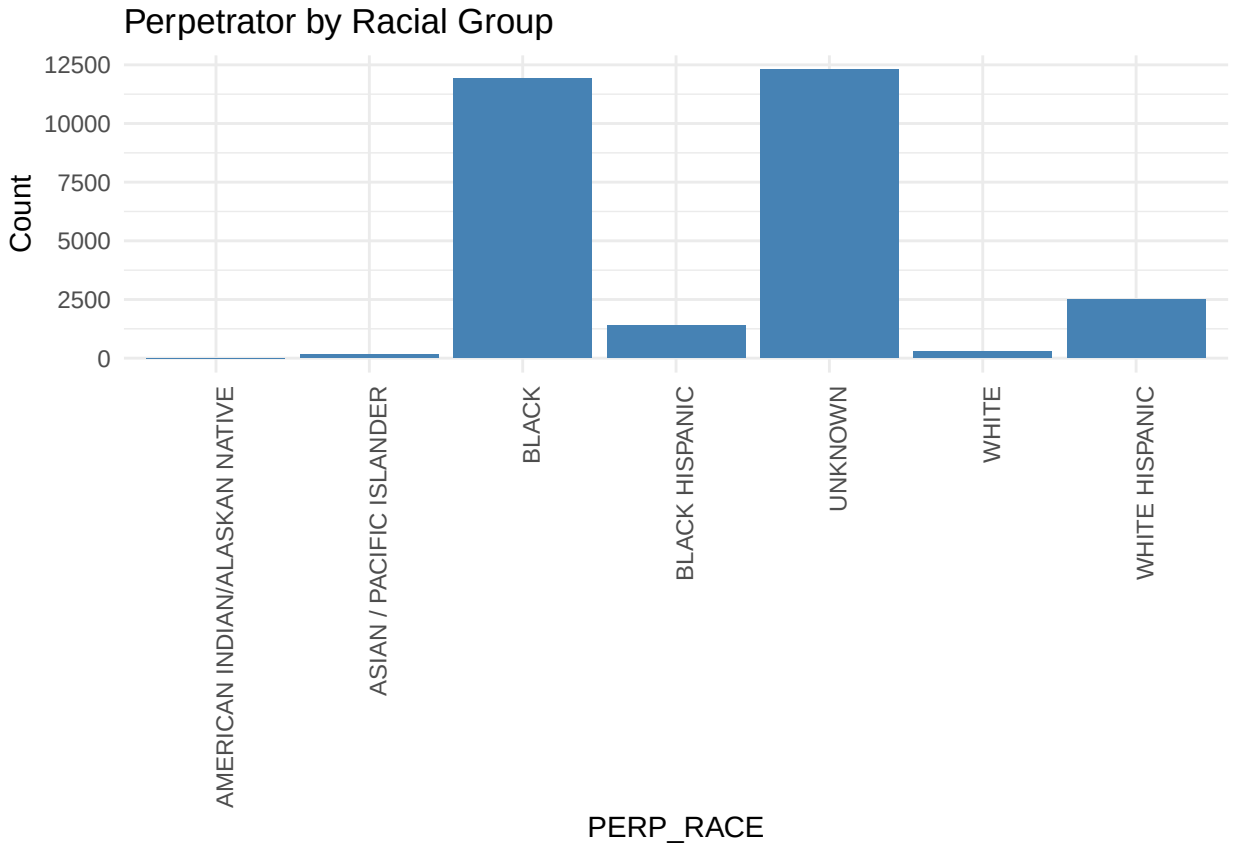

Classification Description
Count
20000
10000
0
COMMERCIAL
DWELLING
HOUSING
OTHER
PARKING LOT
PLAYGROUND
STREET
TRANSIT
UNKNOWN
VEHICLE
LOC_CLASSFCTN_DESC

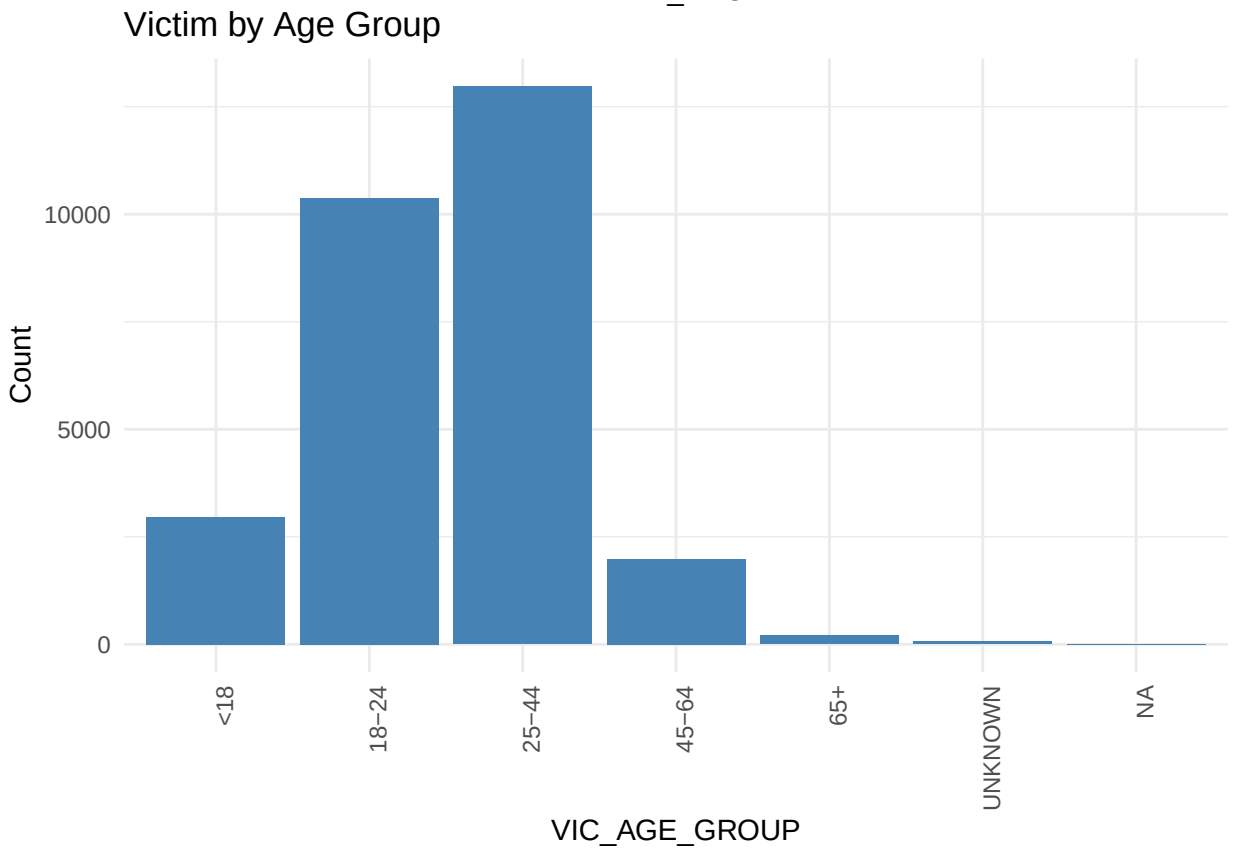

Location of Occurrence Description
Count
20000
10000
0
INSIDE
OUTSIDE
UNKNOWN
LOC_OF_OCCUR_DESC

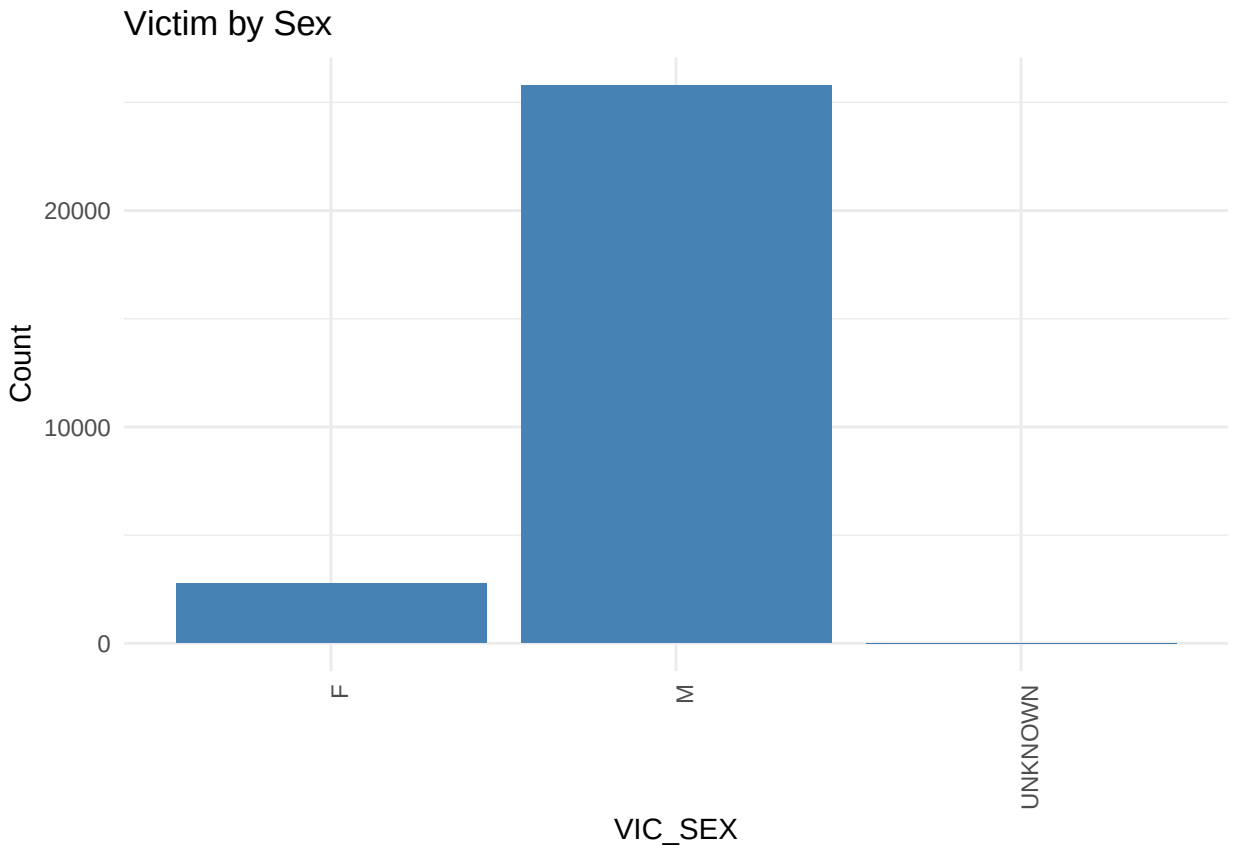

Location Description
Count
15000
10000
5000
0
ATM
BANK
BAR/NIGHT CLUB
BEAUTY/NAIL SALON
CANDY STORE
CHAIN STORE
CHECK CASH
CLOTHING BOUTIQUE
COMMERCIAL BLDG
DEPT STORE
DOCTOR/DENTIST
DRUG STORE
DRY CLEANER/LAUNDRY
FACTORY/WAREHOUSE
FAST FOOD
GAS STATION
GROCERY/BODEGA
GYM/FITNESS FACILITY
HOSPITAL
HOTEL/MOTEL
JEWELRY STORE
LIQUOR STORE
LOAN COMPANY
MULTI DWELL - APT BUILD
MULTI DWELL - PUBLIC HOUS
NONE
PHOTO/COPY STORE
PVT HOUSE
RESTAURANT/DINER
SCHOOL
SHOE STORE
SMALL MERCHANT
SOCIAL CLUB/POLICY LOCATI
STORAGE FACILITY
STORE UNCLASSIFIED
SUPERMARKET
TELECOMM. STORE
UNKNOWN
VARIETY STORE
VIDEO STORE
LOCATION_DESC

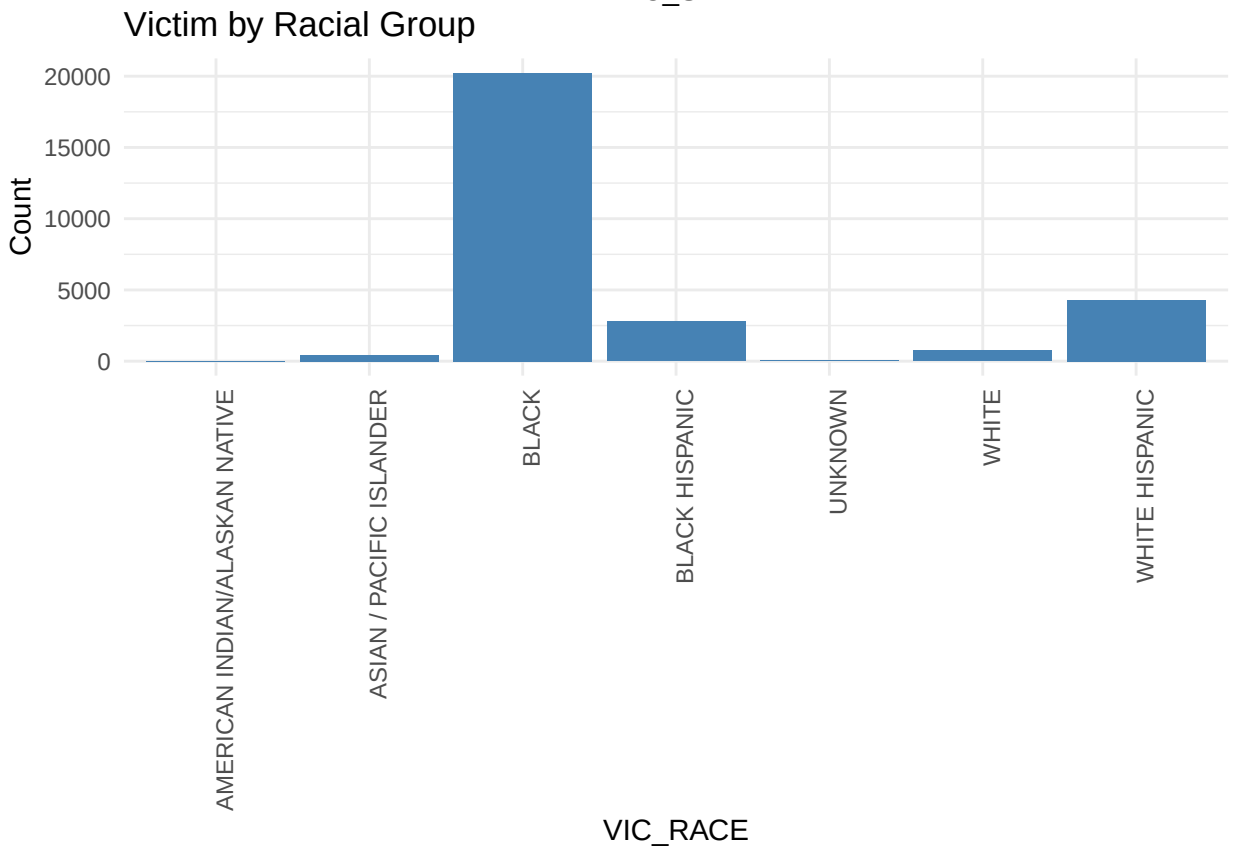

Perpetrator by Age Group
Count
10000
5000
0
<18
18–24
25–44
45–64
65+
UNKNOWN
NA
PERP_AGE_GROUP


Perpetrator by Sex
Count
15000
10000
5000
0
F
M
UNKNOWN
PERP_SEX


Perpetrator by Racial Group
Count
12500
10000
7500
5000
2500
0
AMERICAN INDIAN/ALASKAN NATIVE
ASIAN / PACIFIC ISLANDER
BLACK
BLACK HISPANIC
UNKNOWN
WHITE
WHITE HISPANIC
PERP_RACE


Victim by Age Group
Count
10000
5000
0
<18
18–24
25–44
45–64
65+
UNKNOWN
NA
VIC_AGE_GROUP

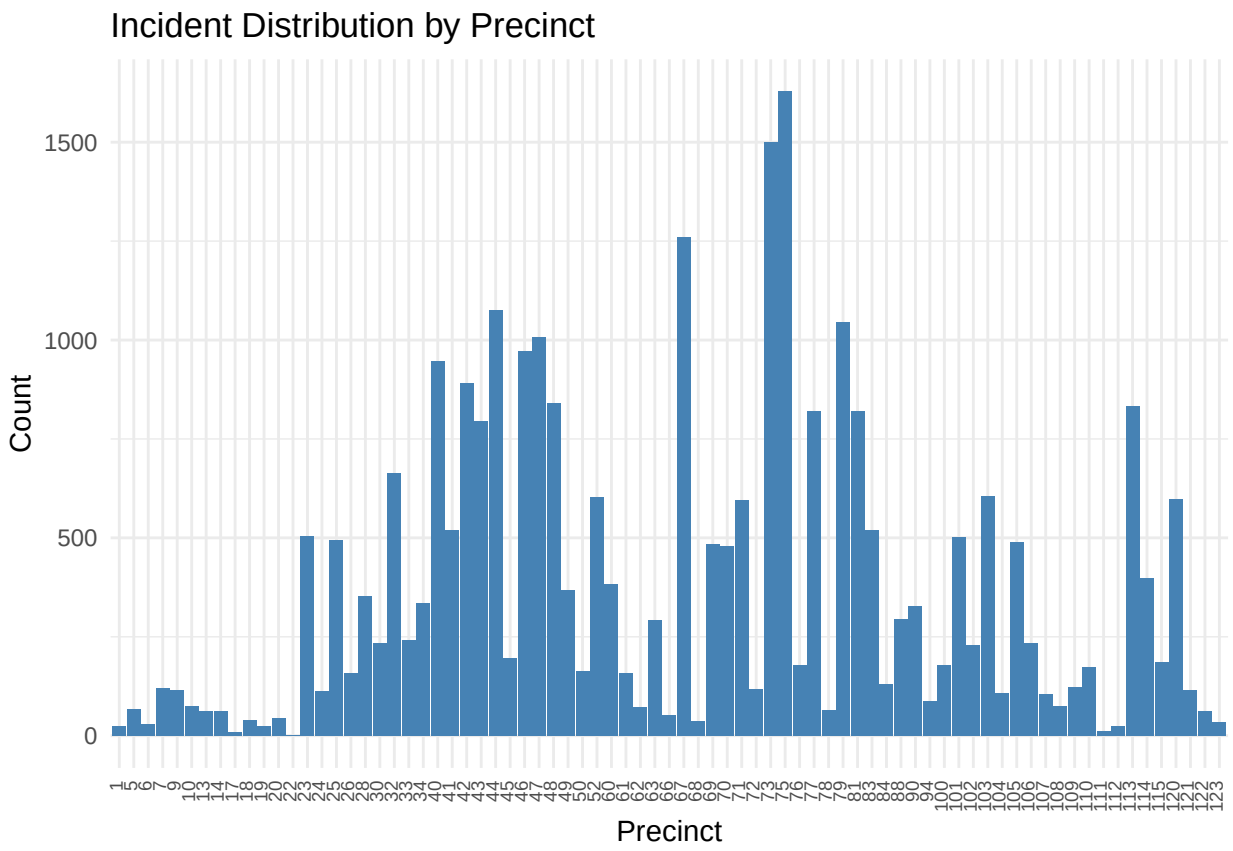

Victim by Sex
Count
20000
10000
0
F
M
UNKNOWN
VIC_SEX


Victim by Racial Group
Count
20000
15000
10000
5000
0
AMERICAN INDIAN/ALASKAN NATIVE
ASIAN / PACIFIC ISLANDER
BLACK
BLACK HISPANIC
UNKNOWN
WHITE
WHITE HISPANIC
VIC_RACE

Incident Distribution by Precinct

The following table shows the distribution of shooting incidents by precinct.

```
# Create the precinct table
precinct_table <- table(NYPD_data_cleaned$PRECINCT)

# Convert the table to a data frame
precinct_df <- as.data.frame(precinct_table)
colnames(precinct_df) <- c("Precinct", "Count")

# Create the bar chart using ggplot2
ggplot(precinct_df, aes(x = as.factor(Precinct), y = Count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Incident Distribution by Precinct", x = "Precinct", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 7, angle = 90, hjust = 1, vjust = 0.25))
```

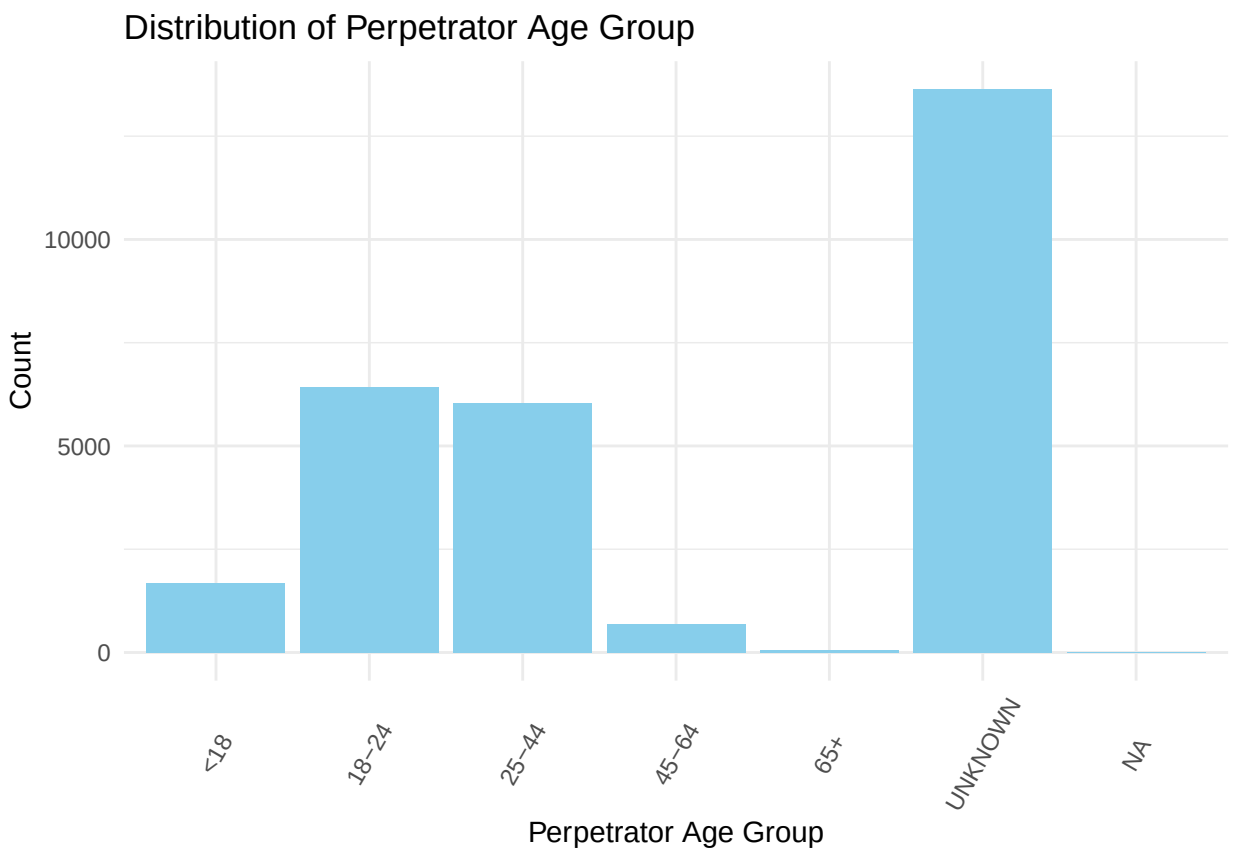


Distribution by Perpetrator Age Group, Sex, and Race

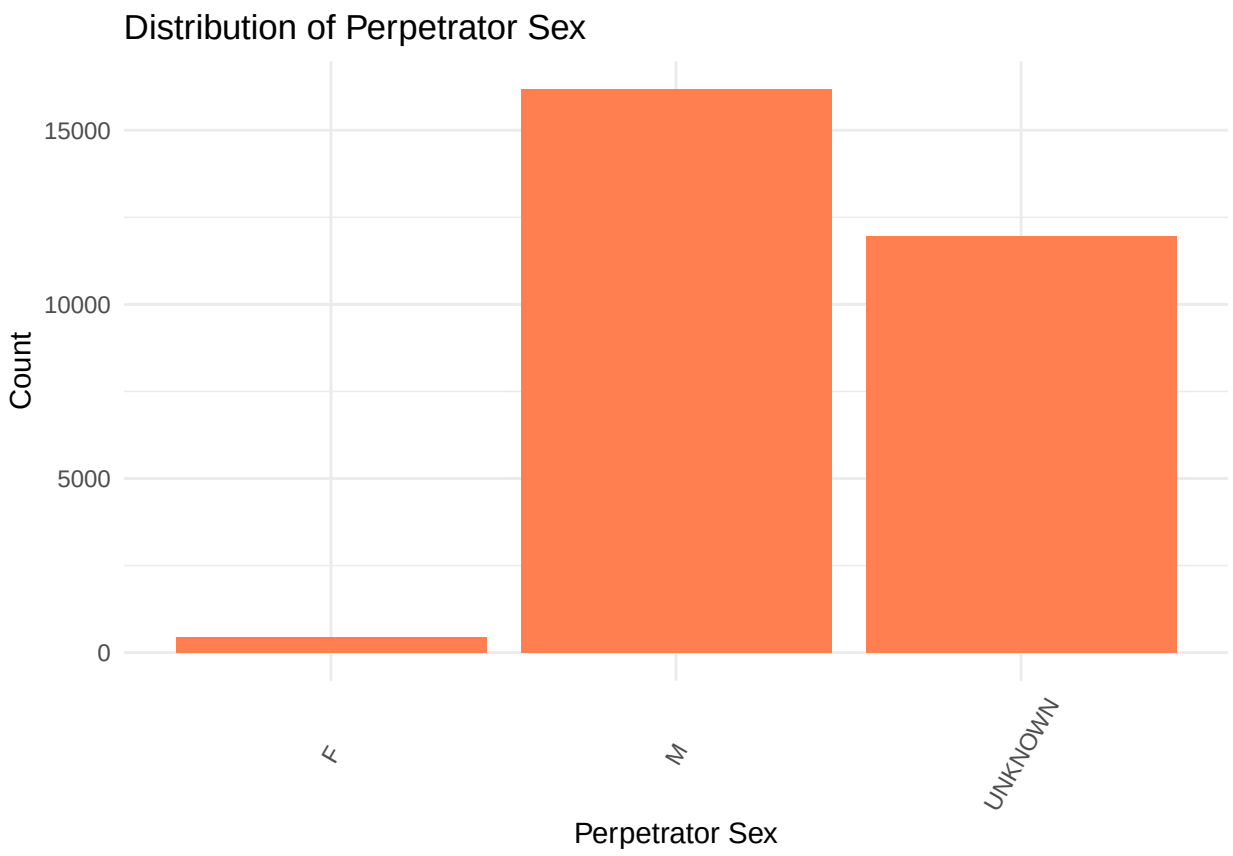
This section analyzes the distribution of perpetrators by age group, sex, and race.

```
# Plot for PERP_AGE_GROUP
ggplot(NYPD_data_cleaned, aes(x = PERP_AGE_GROUP)) +
  geom_bar(fill = "skyblue") +
```

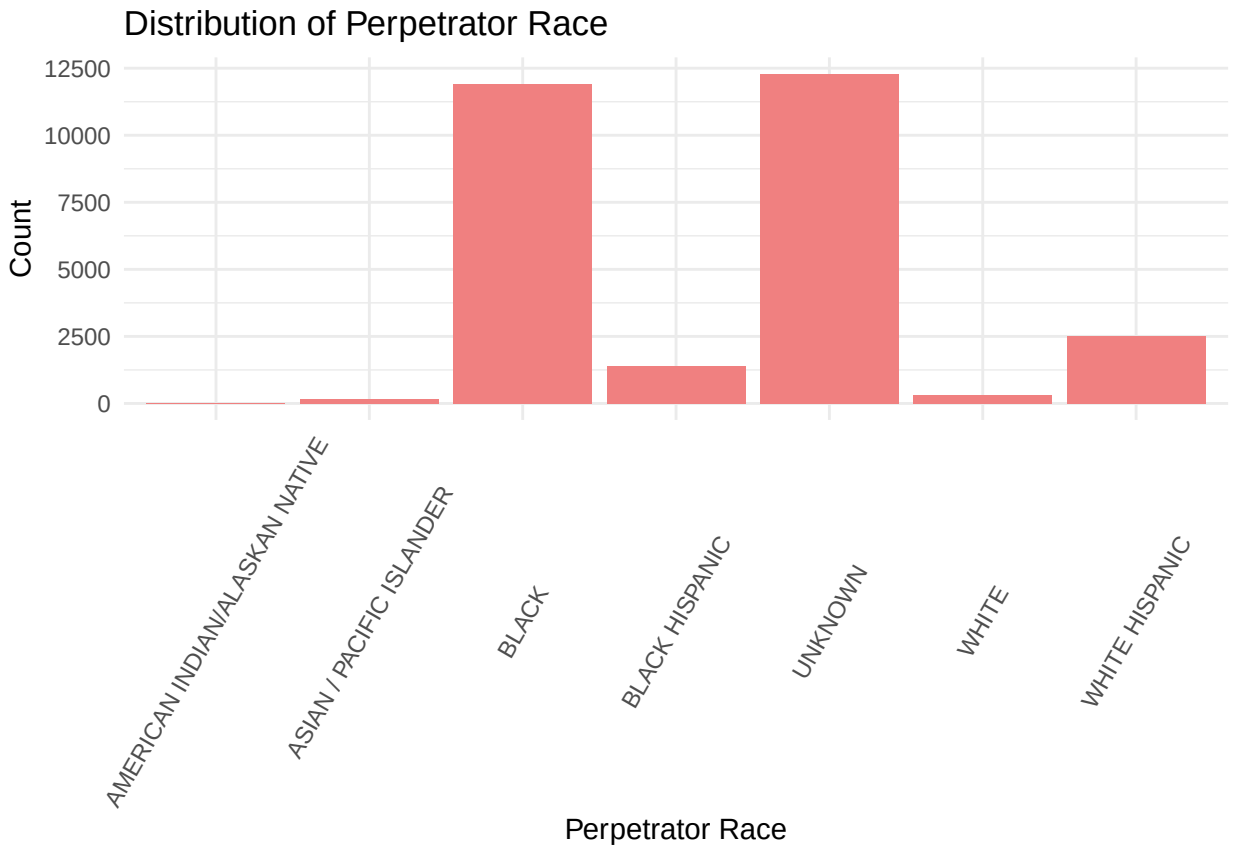
```
  labs(title = "Distribution of Perpetrator Age Group",
       x = "Perpetrator Age Group",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 0.5, vjust = 0.5))
```




```
# Plot for PERP_SEX
ggplot(NYPD_data_cleaned, aes(x = PERP_SEX)) +
  geom_bar(fill = "coral") +
  labs(title = "Distribution of Perpetrator Sex",
       x = "Perpetrator Sex",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 0.5, vjust = 0.5))
```




```
# Plot for PERP_RACE
ggplot(NYPD_data_cleaned, aes(x = PERP_RACE)) +
  geom_bar(fill = "lightcoral") +
  labs(title = "Distribution of Perpetrator Race",
       x = "Perpetrator Race",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 0.5, vjust = 0.5))
```

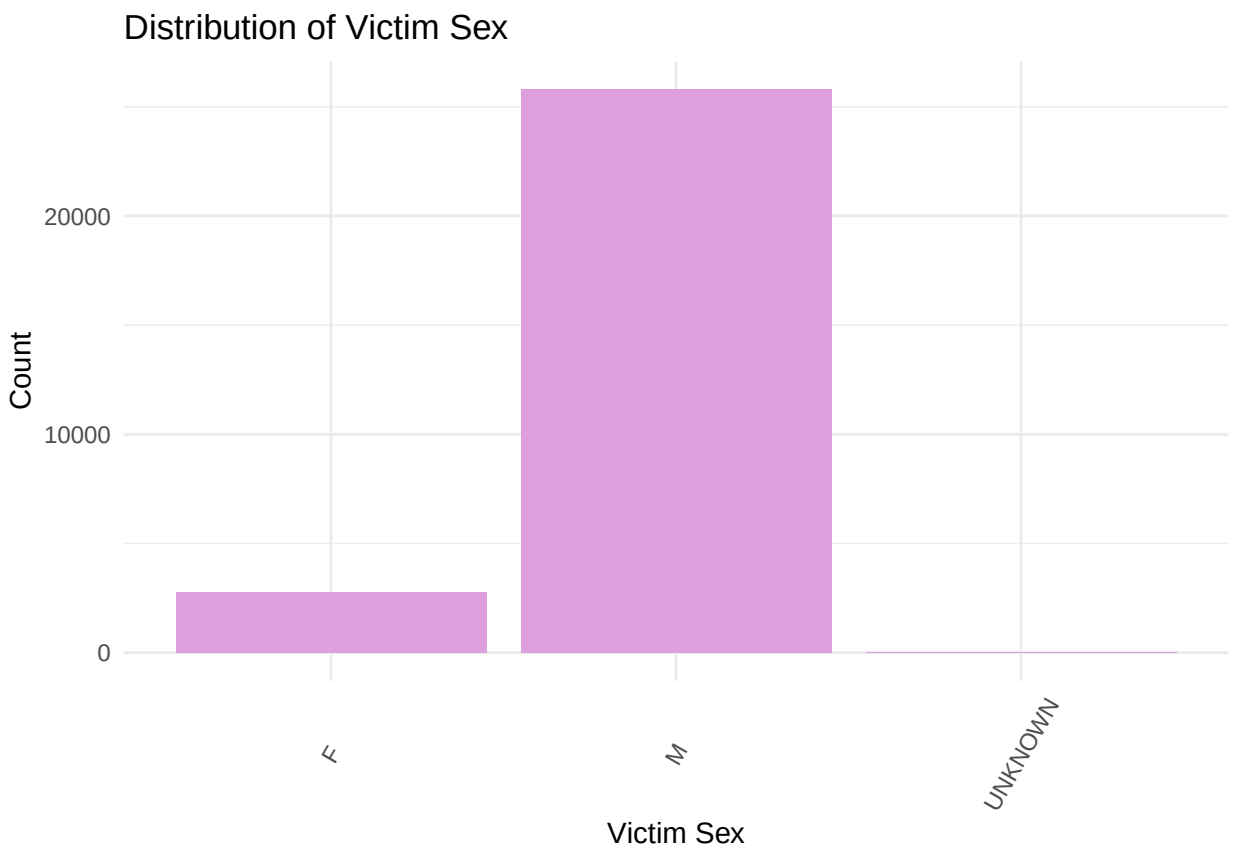


Distribution by Victim Age Group, Sex, and Race

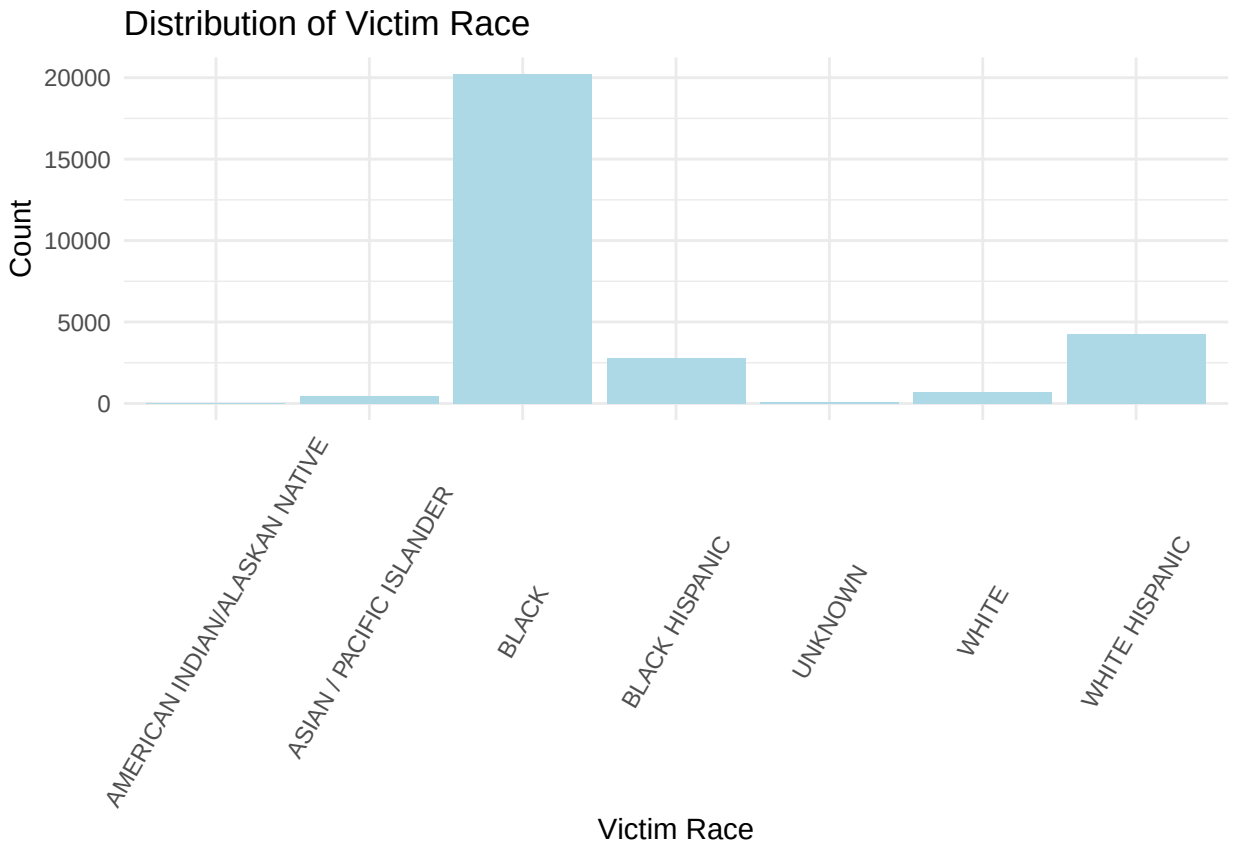
This section analyzes the distribution of victims by age group, sex, and race.

```
# Plot for VIC_AGE_GROUP
ggplot(NYPD_data_cleaned, aes(x = VIC_AGE_GROUP)) +
  geom_bar(fill = "lightgreen") +
  labs(title = "Distribution of Victim Age Group",
       x = "Victim Age Group",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 0.5, vjust = 0.5))
```


```
# Plot for VIC_SEX
ggplot(NYPD_data_cleaned, aes(x = VIC_SEX)) +
  geom_bar(fill = "plum") +
  labs(title = "Distribution of Victim Sex",
       x = "Victim Sex",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 0.5, vjust = 0.5))
```




```
# Plot for VIC_RACE
ggplot(NYPD_data_cleaned, aes(x = VIC_RACE)) +
  geom_bar(fill = "lightblue") +
  labs(title = "Distribution of Victim Race",
       x = "Victim Race",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 0.5, vjust = 0.5))
```



Contingency Table: Precinct vs. Statistical Murder Flag

This section presents a contingency table showing the relationship between precincts and whether an incident resulted in a murder.

```
# Convert STATISTICAL_MURDER_FLAG to 0 and 1
NYPD_data_cleaned <- NYPD_data_cleaned %>%
  mutate(STATISTICAL_MURDER_FLAG = ifelse(STATISTICAL_MURDER_FLAG == TRUE, 1, 0))

# Create the contingency table
contingency_table <- table(NYPD_data_cleaned$PRECINCT,
↪  NYPD_data_cleaned$STATISTICAL_MURDER_FLAG)

# Convert the table to a data frame
contingency_df <- as.data.frame.matrix(contingency_table)

# Rename the column names
colnames(contingency_df) <- c("Non-Murder", "Murder")
contingency_df <- tibble::rownames_to_column(contingency_df, "Precinct")

# Display the contingency table
kable(contingency_df, col.names = c("Precinct", "Non-Murder", "Murder"), caption =
↪  "Precinct vs. Statistical Murder Flag: Shooting resulted in the victim's death which
↪  would be counted as a murder.") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive")) %>%
  column_spec(3) %>%
```

```
scroll_box(height = "450px", width = "100%") %>%
kable_classic("striped", full_width = F)
```

Table 2: Precinct vs. Statistical Murder Flag: Shooting resulted in the victim's death which would be counted as a murder.

Precinct	Non-Murder	Murder
1	18	7
5	50	17
6	22	6
7	110	10
9	89	25
10	57	17
13	45	16
14	45	16
17	8	2
18	32	6
19	18	6
20	37	6
22	1	0
23	423	82
24	86	27
25	409	85
26	138	19
28	294	59
30	189	45
32	540	123
33	198	44
34	281	54
40	783	164
41	420	99
42	715	175
43	653	143
44	876	200
45	158	37
46	772	200
47	785	221
48	676	165
49	284	84
50	130	32
52	490	114
60	303	80
61	114	43
62	53	19
63	236	56
66	45	8
67	1009	250
68	30	6
69	393	91

70	371	108
71	498	97
72	92	25
73	1223	277
75	1299	329
76	146	33
77	645	176
78	57	8
79	849	196
81	665	156
83	424	96
84	111	20
88	232	62
90	266	62
94	75	12
100	161	17
101	414	88
102	176	53
103	505	100
104	87	21
105	383	105
106	161	72
107	82	23
108	66	9
109	91	32
110	135	39
111	12	0
112	15	8
113	673	161
114	326	71
115	144	41
120	481	116
121	93	21
122	38	25
123	25	8

Chi-Squared Test: Precinct vs. Statistical Murder Flag

The following code performs a chi-squared test to evaluate the relationship between precincts and whether an incident resulted in a murder.

```
chisq_result <- chisq.test(contingency_table)
chisq_summary <- data.frame(
  Statistic = chisq_result$statistic,
  Parameter = chisq_result$parameter,
  P.Value = chisq_result$p.value
)
kable(chisq_summary, caption = "Chi-Squared Test Result: Precinct vs. Statistical Murder
↪  Flag") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

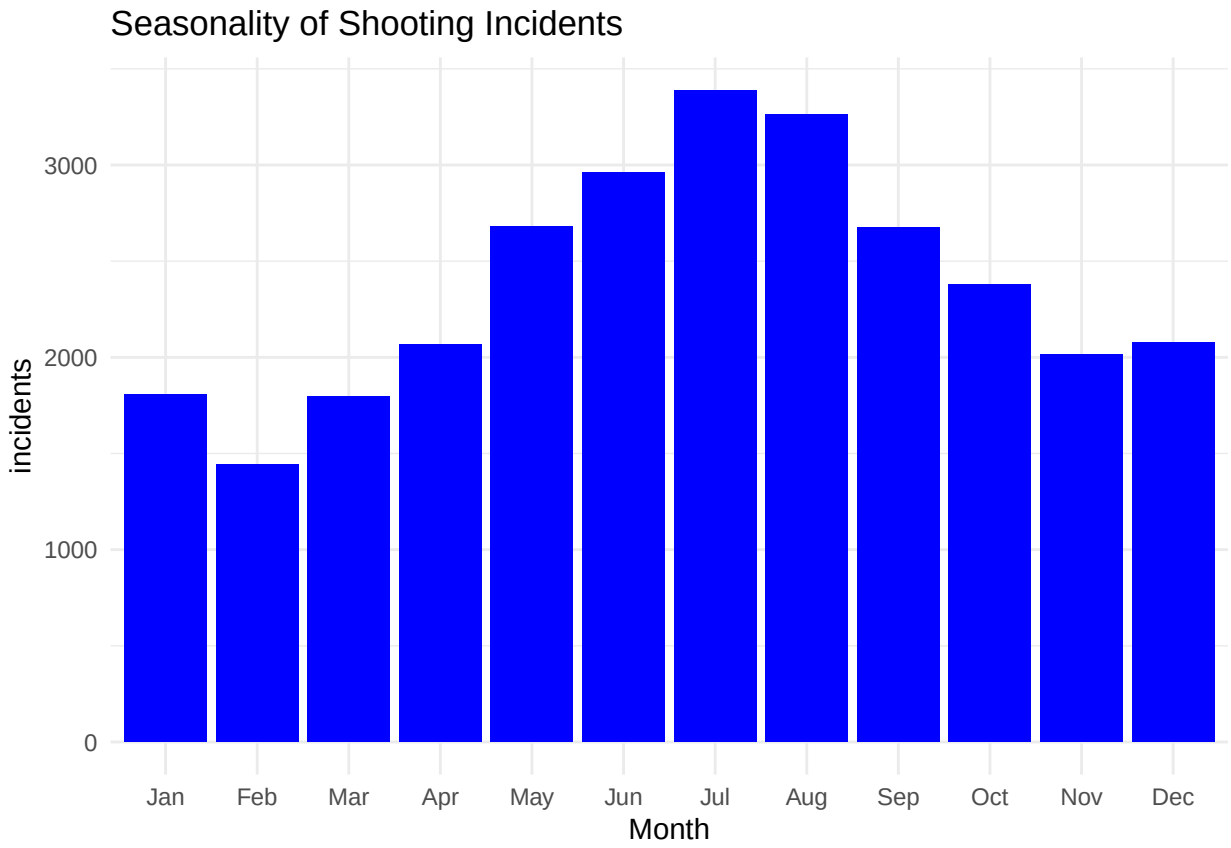
Table 3: Chi-Squared Test Result: Precinct vs. Statistical Murder Flag

	Statistic	Parameter	P.Value
X-squared	145.8652	76	2.6e-06

Seasonality of Shooting Incidents

This section examines the seasonality of shooting incidents by plotting the number of incidents per month.

```
# Convert OCCUR_DATE to Date type if it is not already
NYPD_data_cleaned$OCCUR_DATE <- as.Date(NYPD_data_cleaned$OCCUR_DATE, format =
↪  "%m/%d/%Y")

# Extract Month, Day, and Year
NYPD_data_cleaned <- NYPD_data_cleaned %>%
  mutate(
    Month = month(OCCUR_DATE, label = TRUE), # Extract month as labeled factor
    Day = day(OCCUR_DATE),                   # Extract day
    Year = year(OCCUR_DATE)                  # Extract year
  )

# Seasonality
NYPD_data_cleaned %>%
  group_by(Month) %>%
  summarise(incidents = n()) %>%
  ggplot(aes(x = Month, y = incidents)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Seasonality of Shooting Incidents") +
  theme_minimal()
```

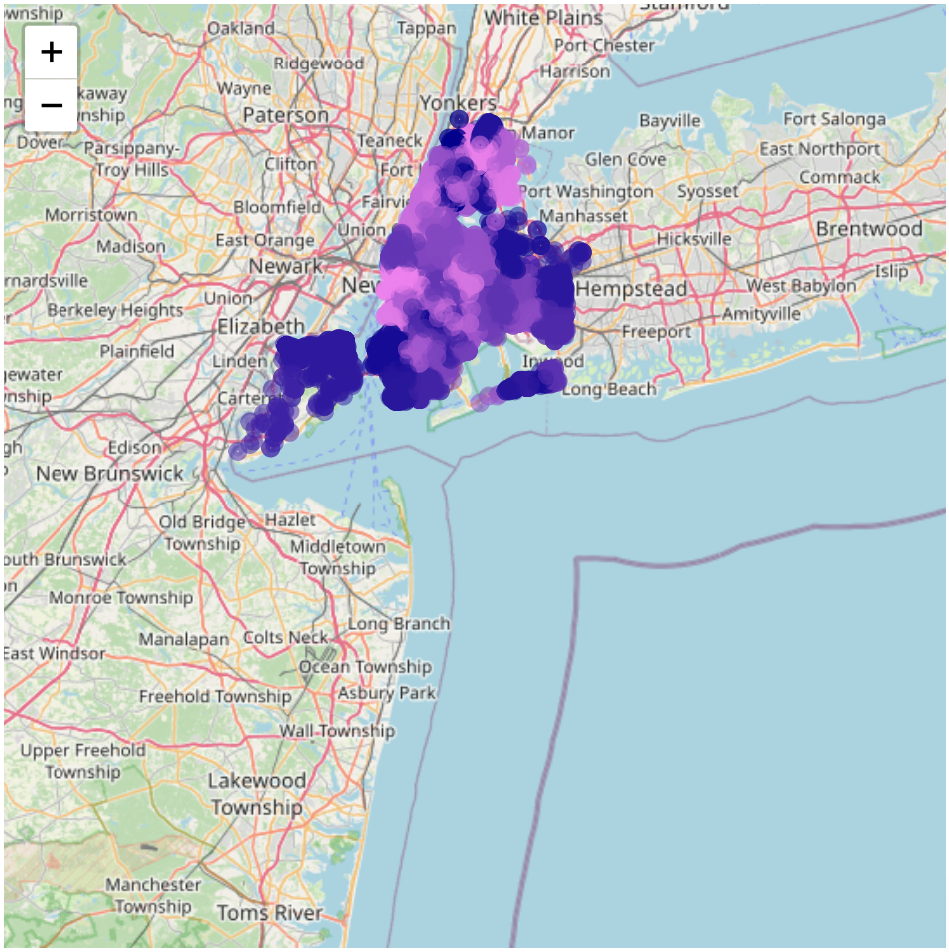


Clustering Analysis: NYPD Shooting Clusters

This section performs k-means clustering on the latitude and longitude coordinates of the shooting incidents to identify clusters.

```
# Remove rows with NA in Latitude or Longitude for clustering
data_no_na <- NYPD_data_cleaned %>%
  filter(!is.na(Latitude) & !is.na(Longitude))

# Precinct count
total_precincts <- NYPD_data_cleaned %>%
  summarise(total_precincts = n_distinct(PRECINCT)) %>%
  pull(total_precincts)

# Perform k-means clustering
data_for_clustering <- data_no_na %>%
  select(Latitude, Longitude)
clusters <- kmeans(data_for_clustering, centers = total_precincts)

# Add cluster results back to the original data frame
data_no_na$cluster <- clusters$cluster

# Calculate the count of each cluster
cluster_counts <- data_no_na %>%
  group_by(cluster) %>%
  summarise(count = n()) %>%
```

```r
  arrange(cluster)  # Ensure numeric order

# Create custom labels for the legend from the sorted cluster_counts
custom_labels <- paste("Cluster", cluster_counts$cluster, ":", cluster_counts$count)

# Define a custom color palette using viridis
custom_colors <- colorRampPalette(c("violet", "darkblue"))(total_precincts)
color_palette <- colorFactor(palette = custom_colors, domain = data_no_na$cluster)

# Create the interactive map with a custom HTML legend
leaflet(data_no_na) %>%
  addTiles() %>%
  addCircleMarkers(
    ~Longitude, ~Latitude,
    color = ~color_palette(cluster),
    radius = 3,
    popup = ~paste("Cluster:", cluster)
  ) %>%
  addControl(
    html = htmltools::tags$div(
      style = "max-height: 400px; overflow-y: auto; padding: 10px; background:
      ↪  linear-gradient(to bottom, white, lightblue); opacity: 1;",
      htmltools::tags$h4("Cluster Counts"),
      htmltools::tags$ul(
        style = "list-style: none; padding: 0;",
        lapply(seq_along(custom_labels), function(i) {
          htmltools::tags$li(
            style = paste("color:", custom_colors[i], ";"),
            custom_labels[i]
          )
        })
      )
    ),
    position = "bottomright"
  )
```



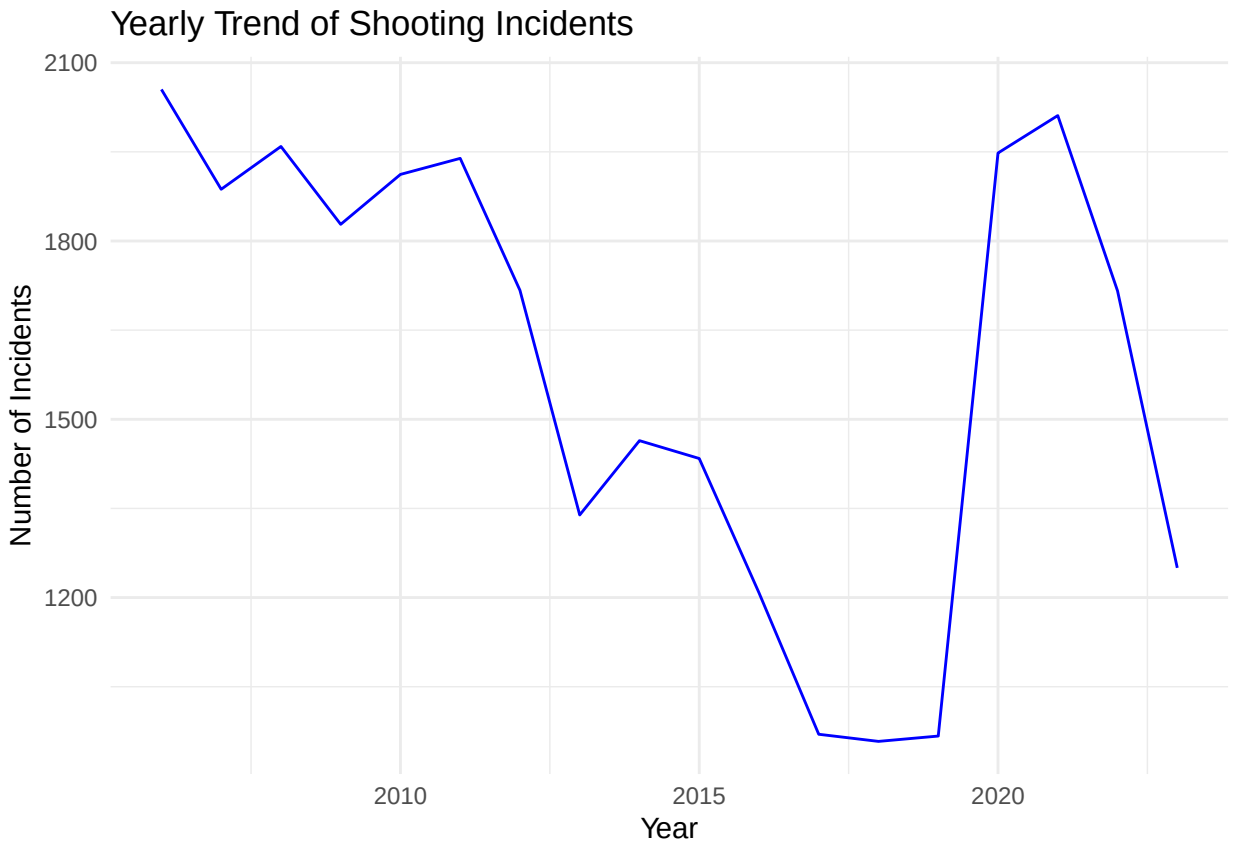
Cluster Counts

Cluster 1 : 357
Cluster 2 : 499
Cluster 3 : 498
Cluster 4 : 308
Cluster 5 : 150
Cluster 6 : 171
Cluster 7 : 186
Cluster 8 : 557
Cluster 9 : 60
Cluster 10 : 331
Cluster 11 : 479
Cluster 12 : 141
Cluster 13 : 338
Cluster 14 : 503
Cluster 15 : 248
Cluster 16 : 548
Cluster 17 : 442
Cluster 18 : 430
Cluster 19 : 516
Cluster 20 : 309

Temporal Analysis by Year

Analyze trends over multiple years to understand long-term changes in shooting incidents.

```
# Group by Year and plot the yearly trend of shooting incidents
NYPD_data_cleaned %>%
  group_by(Year) %>%
  summarise(incidents = n()) %>%
  ggplot(aes(x = Year, y = incidents)) +
  geom_line(color = "blue") +
  labs(title = "Yearly Trend of Shooting Incidents", x = "Year", y = "Number of
  ↪  Incidents") +
  theme_minimal()
```



Predictive Modeling with Logistic Regression

Let's add an example of how to set up a logistic regression model to predict whether an incident results in a murder.

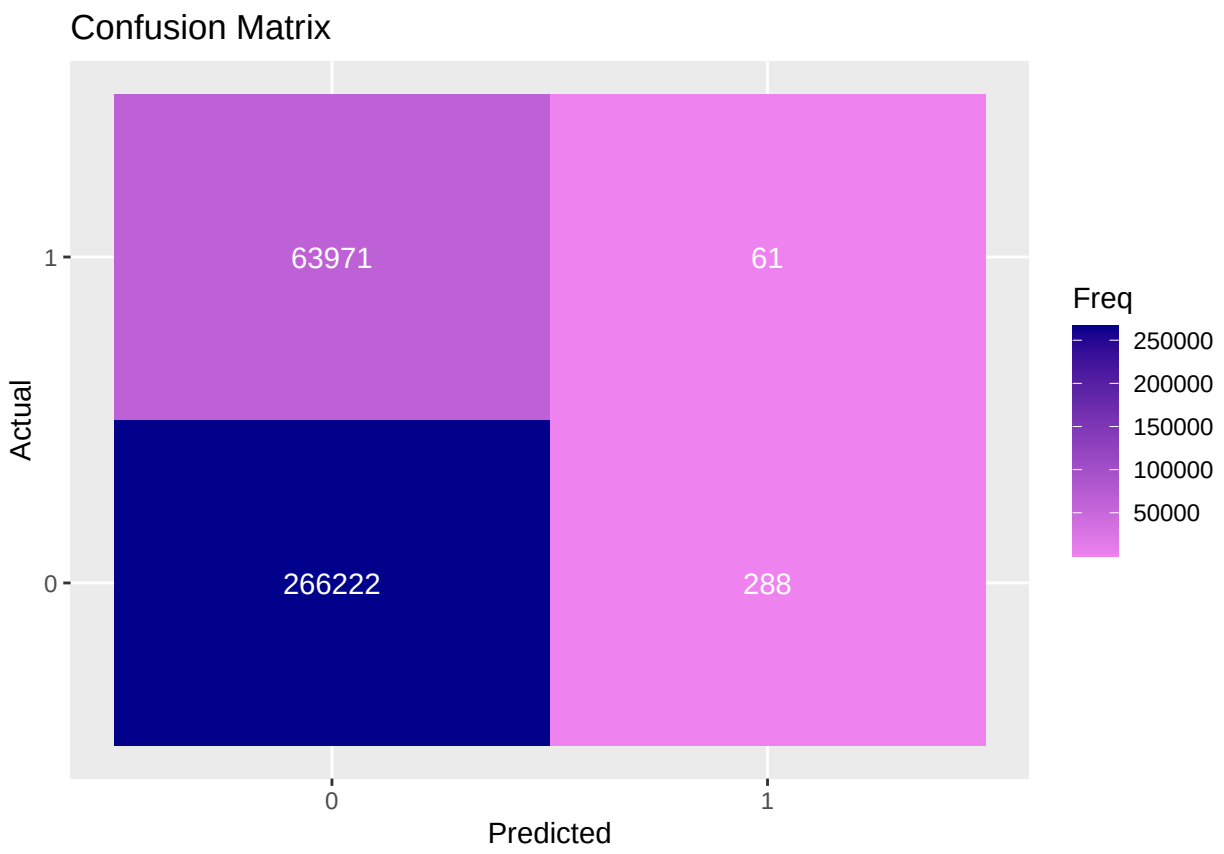
```
# Prepare the data
NYPD_data_cleaned_filtered <- NYPD_data_cleaned %>%
  filter(!is.na(STATISTICAL_MURDER_FLAG)) %>%
  mutate(STATISTICAL_MURDER_FLAG = as.factor(STATISTICAL_MURDER_FLAG))

# Select relevant features and remove rows with NA values
```

```r
features <- c("PRECINCT", "PERP_AGE_GROUP", "PERP_SEX", "PERP_RACE", "VIC_AGE_GROUP",
↪  "VIC_SEX", "VIC_RACE", "Latitude", "Longitude", "Month", "Day")
data_model <- NYPD_data_cleaned_filtered %>%
  select(all_of(features), STATISTICAL_MURDER_FLAG) %>%
  drop_na(all_of(features), STATISTICAL_MURDER_FLAG)

# Convert categorical variables to dummy variables
data_matrix <- model.matrix(STATISTICAL_MURDER_FLAG ~ ., data = data_model)[,-1]
target <- data_model$STATISTICAL_MURDER_FLAG

# Split the data into training and test sets
set.seed(123)  # For reproducibility
trainIndex <- createDataPartition(target, p = 0.8, list = FALSE)

# Prepare training and testing datasets
data_train <- data_matrix[trainIndex, ]
data_test <- data_matrix[-trainIndex, ]
target_train <- target[trainIndex]
target_test <- target[-trainIndex]

# Ensure there are no missing values in the training and testing sets
if(any(is.na(data_train)) | any(is.na(data_test))) {
  stop("There are still missing values in the data.")
}

# Train the logistic regression model
model <- glmnet(data_train, target_train, family = "binomial")

# Make predictions
predictions <- predict(model, newx = data_test, type = "response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)

# Upsample target_test to match the length of predicted_classes if necessary
if (length(predicted_classes) != length(target_test)) {
  target_test_upsampled <- rep(target_test, length.out = length(predicted_classes))
} else {
  target_test_upsampled <- target_test
}

# Ensure both predicted and actual target are factors with the same levels
predicted_classes <- factor(predicted_classes, levels = c(0, 1))
target_test_upsampled <- factor(target_test_upsampled, levels = c(0, 1))

# Create the confusion matrix
conf_matrix <- confusionMatrix(predicted_classes, target_test_upsampled)

# Print the confusion matrix
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      0      1
```

```
##          0 266222  63971
##          1    288     61
##
##                Accuracy : 0.8056
##                  95% CI : (0.8042, 0.8069)
##     No Information Rate : 0.8063
##     P-Value [Acc > NIR] : 0.8416
##
##                   Kappa : -2e-04
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9989194
##             Specificity : 0.0009526
##          Pos Pred Value : 0.8062618
##          Neg Pred Value : 0.1747851
##              Prevalence : 0.8062818
##          Detection Rate : 0.8054105
##    Detection Prevalence : 0.9989442
##       Balanced Accuracy : 0.4999360
##
##        'Positive' Class : 0
##
```

```r
# Convert the confusion matrix to a data frame
conf_matrix_df <- as.data.frame(conf_matrix$table)

# Plot the confusion matrix
ggplot(conf_matrix_df, aes(x = Prediction, y = Reference, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white") +
  scale_fill_gradient(low = "violet", high = "darkblue") +
  labs(title = "Confusion Matrix", x = "Predicted", y = "Actual")
```



Confusion Matrix

The confusion matrix shows the performance of the classification model by comparing the predicted labels to the actual labels. The rows represent the predicted classes, while the columns represent the actual classes.

Model Performance Metrics

Additional performance metrics for the logistic regression model, such as ROC-AUC, precision, recall, and F1-score, to provide a more comprehensive evaluation.

```
# Calculate additional performance metrics
# Ensure the target_test and predictions are of the same length
if (length(predictions) != length(target_test_upsampled)) {
  stop("The lengths of predictions and target_test do not match.")
}

# Convert predictions to numeric
numeric_predictions <- as.numeric(predictions)

# Calculate additional performance metrics
roc_auc <- pROC::roc(response = target_test_upsampled, predictor = numeric_predictions)
roc_auc_value <- pROC::auc(roc_auc)

precision <- caret::posPredValue(predicted_classes, target_test_upsampled)
recall <- caret::sensitivity(predicted_classes, target_test_upsampled)
```

```r
f1_score <- 2 * ((precision * recall) / (precision + recall))

performance_metrics <- data.frame(
  Metric = c("ROC-AUC", "Precision", "Recall", "F1-Score"),
  Value = c(roc_auc_value, precision, recall, f1_score)
)

# Display the performance metrics
kable(performance_metrics, caption = "Logistic Regression Performance Metrics") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

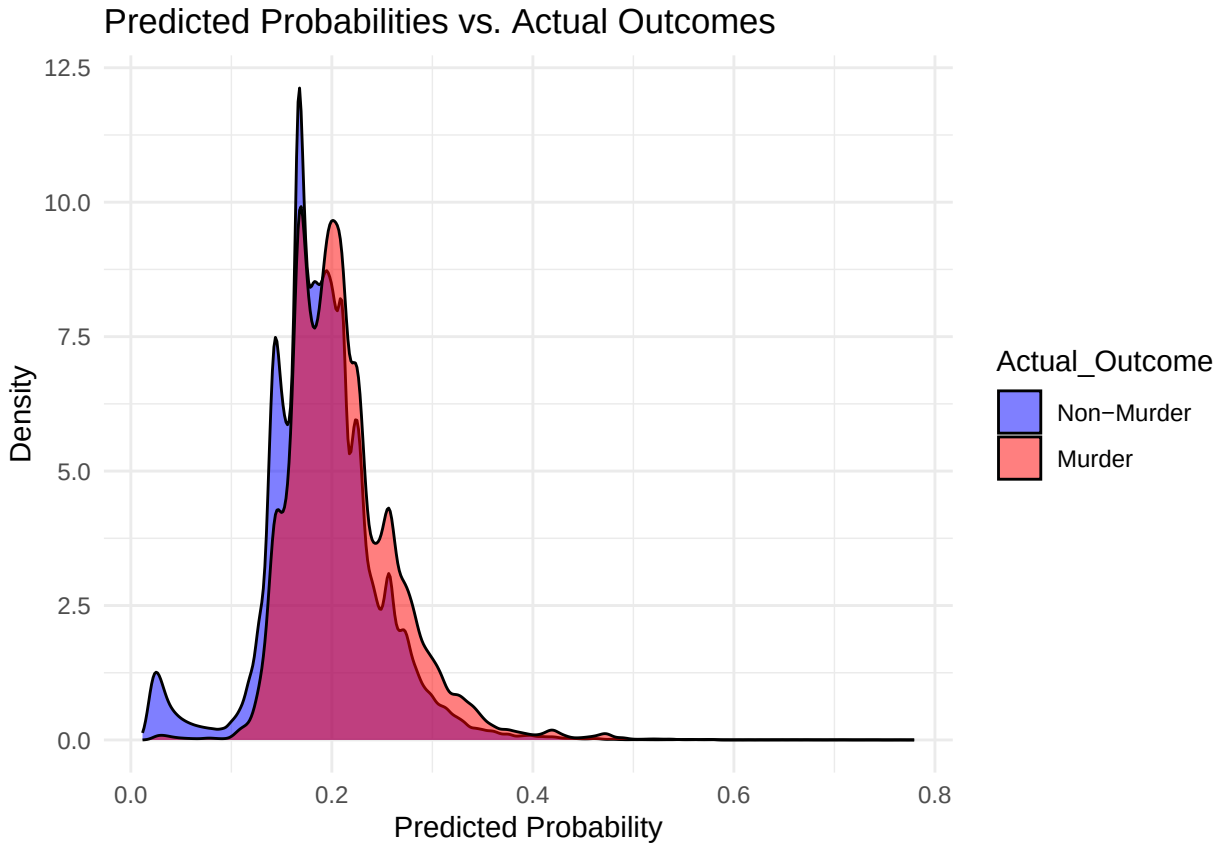
Table 4: Logistic Regression Performance Metrics

Metric	Value
ROC-AUC	0.6069635
Precision	0.8062618
Recall	0.9989194
F1-Score	0.8923099

Predicted Probability Graph Explanation

The following graph shows the predicted probability of an incident resulting in a murder based on the logistic regression model. The x-axis represents the predicted probability, while the y-axis shows the number of incidents with that probability.

```r
# Create a data frame for plotting predicted probabilities vs actual outcomes
plot_data <- data.frame(
  Predicted_Probability = as.numeric(predictions),
  Actual_Outcome = factor(target_test, levels = c(0, 1), labels = c("Non-Murder",
  ↪  "Murder"))
)

# Plot predicted probabilities vs actual outcomes
ggplot(plot_data, aes(x = Predicted_Probability, fill = Actual_Outcome)) +
  geom_density(alpha = 0.5) +
  labs(title = "Predicted Probabilities vs. Actual Outcomes", x = "Predicted
  ↪  Probability", y = "Density") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal()
```



Bias Analysis in NYPD Shooting Incident Data Analysis

When analyzing sensitive data such as the NYPD Shooting Incident Data, it's crucial to recognize potential biases that may influence the results and interpretations. Here are several key areas where bias could be present in the provided analysis:

Selection Bias The data only includes shooting incidents that were reported and recorded by the NYPD. Incidents that were not reported or recorded accurately are missing, leading to an incomplete picture. Filtering out categories with exactly one occurrence may remove significant but rare events, introducing bias by focusing only on more frequent incidents.

Survivorship Bias Analysis may disproportionately focus on certain precincts or demographics that have higher reporting rates, thereby neglecting areas or groups with lower reporting rates.

Reporting Bias There could be discrepancies in how different precincts or officers report incidents, which may affect the data consistency. For instance, some precincts might be more diligent or have different criteria for reporting.

Measurement Bias The way data is recorded might introduce errors. For example, inaccuracies in recording the exact location, time, or demographics can lead to skewed results. The "STATISTICAL_MURDER_FLAG" relies on accurate determination of whether a shooting resulted in a murder, which might be influenced by subjective judgment or administrative decisions.

Sample Size Bias Removing rows with NA values might lead to a reduced dataset that does not fully represent the original population, especially if NA values are not randomly distributed.

Imbalance in Data The dataset might have an imbalance in the number of incidents across different categories (e.g., more incidents in certain boroughs or involving certain demographics), which can lead to biased model predictions. The confusion matrix indicates a high imbalance between non-murder and murder cases, which affects the model's ability to accurately predict rare events.

Model Bias Logistic regression and other machine learning models may inherit and amplify existing biases in the data. If the training data is biased, the model predictions will also be biased. The choice of features and how they are encoded (e.g., converting categorical variables to dummy variables) can influence the model's performance and fairness.

Suggestions to Address Bias

Data Augmentation and Cleaning Insure thorough cleaning and preprocessing of data to minimize errors and inconsistencies. Consider imputation methods for handling NA values rather than simply removing them. Augment the dataset with additional relevant features that might help reduce bias, such as socioeconomic indicators or historical crime rates.

Balanced Sampling Use techniques like oversampling, undersampling, or synthetic data generation (e.g., SMOTE) to address class imbalances in the dataset.

Bias Detection and Mitigation Implement fairness-aware algorithms and bias detection techniques to identify and mitigate biases in the model. Regularly evaluate the model's performance across different demographic groups to ensure fairness and equity.

Transparency and Documentation Clearly document the data sources, preprocessing steps, and any assumptions made during the analysis to enhance transparency and reproducibility. Provide contextual information about potential biases and limitations of the analysis to inform stakeholders and decision-makers.

Engage with Stakeholders Collaborate with community stakeholders, law enforcement, and subject matter experts to validate findings and ensure that the analysis considers diverse perspectives and real-world implications.

Conclusions from the NYPD Shooting Incident Analysis

The NYPD Shooting Incident Analysis provides valuable insights into the distribution, temporal trends, and demographic patterns of shooting incidents in New York City. While the analysis highlights significant patterns and potential areas of intervention, it also underscores the need for continuous improvement in data quality, model performance, and bias mitigation. By addressing these recommendations, the analysis can provide more accurate, fair, and actionable insights to help reduce shooting incidents and improve public safety in New York City.

Recommendations

- Improve Data Quality: Enhance the accuracy and completeness of the data collection process. Addressing missing values through advanced imputation techniques and ensuring consistent reporting across precincts can improve the reliability of the analysis.
- Enhance Model Performance: Consider using more advanced machine learning models and techniques to improve specificity and overall model performance. Techniques such as ensemble methods or incorporating additional features could help in better distinguishing between non-murder and murder incidents.
- Address Biases: Implement strategies to detect and mitigate biases in the data and model. Regularly evaluate the model's fairness across different demographic groups and ensure transparency in reporting the analysis.
- Stakeholder Collaboration: Engage with community stakeholders, law enforcement, and experts to validate findings and incorporate diverse perspectives. Collaborative efforts can lead to more effective and equitable interventions.
- Regular Updates and Monitoring: Update the analysis regularly to reflect the most recent data and monitor trends over time. This will help in identifying emerging patterns and making timely adjustments to strategies and policies.