NYPD Shooting Analysis

Jonathan Lampkin

2024-11-15

```
# Set Seed For Reproducibility
set.seed(42)
```

```
# Load Data, Drop Duplicates and Null Rows
nypd_data <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD") %>%
  distinct() %>%
  drop_na()
```

```
## 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.
```

```
# Display structure and summary statistics of the data
str(nypd_data)
```

```
## tibble [2,907 x 21] (S3: tbl_df/tbl/data.frame)
##  $ INCIDENT_KEY           : num [1:2907] 2.45e+08 2.48e+08 2.55e+08 2.50e+08 2.43e+08 ...
##  $ OCCUR_DATE             : chr [1:2907] "05/05/2022" "07/04/2022" "11/30/2022" "08/15/2022" ...
##  $ OCCUR_TIME             : 'hms' num [1:2907] 00:10:00 22:20:00 21:15:00 18:21:00 ...
##   ..- attr(*, "units")= chr "secs"
##  $ BORO                   : chr [1:2907] "MANHATTAN" "BRONX" "BRONX" "QUEENS" ...
##  $ LOC_OF_OCCUR_DESC      : chr [1:2907] "INSIDE" "OUTSIDE" "OUTSIDE" "OUTSIDE" ...
##  $ PRECINCT               : num [1:2907] 14 48 46 101 49 75 49 121 9 69 ...
##  $ JURISDICTION_CODE      : num [1:2907] 0 0 0 2 0 0 0 0 2 0 ...
##  $ LOC_CLASSFCTN_DESC     : chr [1:2907] "COMMERCIAL" "STREET" "STREET" "HOUSING" ...
##  $ LOCATION_DESC          : chr [1:2907] "VIDEO STORE" "(null)" "(null)" "MULTI DWELL - PUBLIC HOUS"
##  $ STATISTICAL_MURDER_FLAG: logi [1:2907] TRUE TRUE TRUE TRUE FALSE TRUE ...
##  $ PERP_AGE_GROUP         : chr [1:2907] "25-44" "(null)" "18-24" "(null)" ...
##  $ PERP_SEX               : chr [1:2907] "M" "(null)" "M" "(null)" ...
##  $ PERP_RACE              : chr [1:2907] "BLACK" "(null)" "BLACK" "(null)" ...
##  $ VIC_AGE_GROUP          : chr [1:2907] "25-44" "18-24" "<18" "18-24" ...
##  $ VIC_SEX                : chr [1:2907] "M" "M" "M" "M" ...
##  $ VIC_RACE               : chr [1:2907] "BLACK" "BLACK" "BLACK" "BLACK" ...
##  $ X_COORD_CD             : num [1:2907] 986050 1016802 1011263 1053494 1021686 ...
```

```
##  $ Y_COORD_CD             : num [1:2907] 214231 250581 251671 161531 251947 ...
##  $ Latitude               : num [1:2907] 40.8 40.9 40.9 40.6 40.9 ...
##  $ Longitude              : num [1:2907] -74 -73.9 -73.9 -73.8 -73.9 ...
##  $ Lon_Lat                : chr [1:2907] "POINT (-73.9935 40.754692)" "POINT (-73.88233 40.854402)" "
```

```
summary(nypd_data)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :238531159   Length:2907        Length:2907       Length:2907
##  1st Qu.:246192328   Class :character   Class1:hms        Class :character
##  Median :252647955   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :256854604                      Mode  :numeric
##  3rd Qu.:268973603
##  Max.   :279758069
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:2907        Min.   :  1.00   Min.   :0.0000    Length:2907
##  Class :character   1st Qu.: 43.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 60.00   Median :0.0000    Mode  :character
##                     Mean   : 62.22   Mean   :0.2425
##                     3rd Qu.: 79.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:2907        Mode :logical           Length:2907
##  Class :character   FALSE:2313              Class :character
##  Mode  :character   TRUE :594               Mode  :character
##
##
##
##    PERP_SEX           PERP_RACE          VIC_AGE_GROUP        VIC_SEX
##  Length:2907        Length:2907        Length:2907        Length:2907
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    VIC_RACE            X_COORD_CD        Y_COORD_CD        Latitude
##  Length:2907        Min.   : 929510   Min.   :127539   Min.   :40.52
##  Class :character   1st Qu.:1000459   1st Qu.:184337   1st Qu.:40.67
##  Mode  :character   Median :1008366   Median :212367   Median :40.75
##                     Mean   :1009286   Mean   :212612   Mean   :40.75
##                     3rd Qu.:1016743   3rd Qu.:242614   3rd Qu.:40.83
##                     Max.   :1059828   Max.   :269204   Max.   :40.91
##    Longitude        Lon_Lat
##  Min.   :-74.20   Length:2907
##  1st Qu.:-73.94   Class :character
##  Median :-73.91   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.73
```

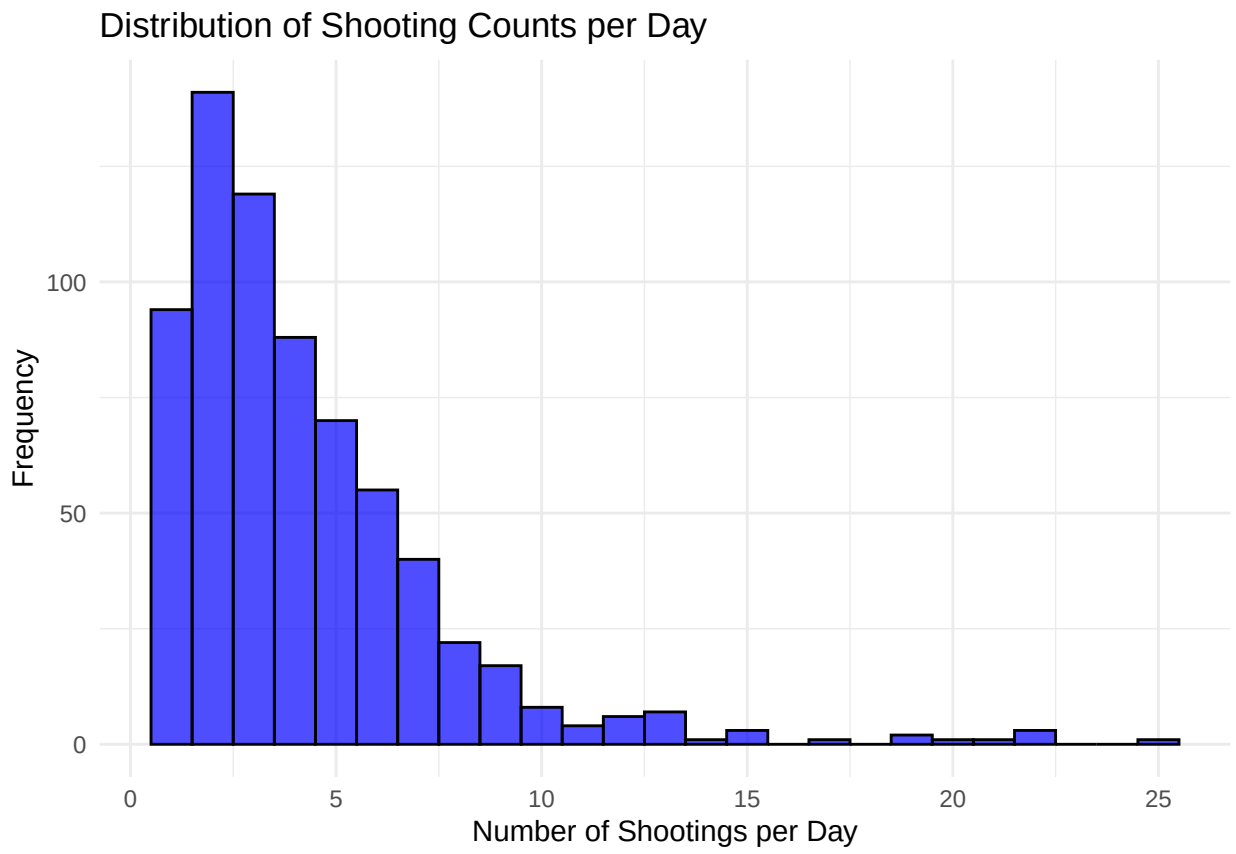
```
head(nypd_data)
```

```
## # A tibble: 6 x 21
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO      LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <chr>      <time>     <chr>     <chr>                <dbl>
## 1    244608249 05/05/2022 00:10      MANHATTAN INSIDE                  14
## 2    247542571 07/04/2022 22:20      BRONX     OUTSIDE                 48
## 3    254911480 11/30/2022 21:15      BRONX     OUTSIDE                 46
## 4    249623757 08/15/2022 18:21      QUEENS    OUTSIDE                101
## 5    243433246 04/10/2022 17:00      BRONX     OUTSIDE                 49
## 6    253757468 11/07/2022 11:35      BROOKLYN  OUTSIDE                 75
## # i 15 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>
```

Exploratory Data Analysis and Data Cleaning/Transformation

```r
# Histogram of Daily Shooting Counts
daily_counts <- nypd_data %>%
  group_by(OCCUR_DATE) %>%
  summarise(SHOOTING_COUNT = n(), .groups = "drop")

ggplot(daily_counts, aes(x = SHOOTING_COUNT)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Shooting Counts per Day",
       x = "Number of Shootings per Day",
       y = "Frequency") +
  theme_minimal()
```




```
nypd_data <- nypd_data %>%
    mutate(
      OCCUR_DATE = mdy(OCCUR_DATE),
      Year = year(OCCUR_DATE),
      Month = factor(month(OCCUR_DATE, label = TRUE, abbr = TRUE), levels = month.abb),
      DayOfWeek = factor(wday(OCCUR_DATE, label = TRUE, abbr = TRUE), levels = c("Sun", "Mon", "Tue", "
      TimeOfDay = case_when(
        hour(OCCUR_TIME) >= 6 & hour(OCCUR_TIME) < 12 ~ "Morning",
        hour(OCCUR_TIME) >= 12 & hour(OCCUR_TIME) < 18 ~ "Afternoon",
        hour(OCCUR_TIME) >= 18 & hour(OCCUR_TIME) < 24 ~ "Evening",
        TRUE ~ "Night"
      ))
```

```
# Display structure and summary statistics of the data
str(nypd_data)
```

```
## tibble [2,907 x 25] (S3: tbl_df/tbl/data.frame)
##  $ INCIDENT_KEY           : num [1:2907] 2.45e+08 2.48e+08 2.55e+08 2.50e+08 2.43e+08 ...
##  $ OCCUR_DATE             : Date[1:2907], format: "2022-05-05" "2022-07-04" ...
##  $ OCCUR_TIME             : 'hms' num [1:2907] 00:10:00 22:20:00 21:15:00 18:21:00 ...
##   ..- attr(*, "units")= chr "secs"
##  $ BORO                   : chr [1:2907] "MANHATTAN" "BRONX" "BRONX" "QUEENS" ...
##  $ LOC_OF_OCCUR_DESC      : chr [1:2907] "INSIDE" "OUTSIDE" "OUTSIDE" "OUTSIDE" ...
##  $ PRECINCT               : num [1:2907] 14 48 46 101 49 75 49 121 9 69 ...
##  $ JURISDICTION_CODE      : num [1:2907] 0 0 0 2 0 0 0 0 2 0 ...
```

```
##  $ LOC_CLASSFCTN_DESC     : chr [1:2907] "COMMERCIAL" "STREET" "STREET" "HOUSING" ...
##  $ LOCATION_DESC          : chr [1:2907] "VIDEO STORE" "(null)" "(null)" "MULTI DWELL - PUBLIC HOUS"
##  $ STATISTICAL_MURDER_FLAG: logi [1:2907] TRUE TRUE TRUE TRUE FALSE TRUE ...
##  $ PERP_AGE_GROUP         : chr [1:2907] "25-44" "(null)" "18-24" "(null)" ...
##  $ PERP_SEX               : chr [1:2907] "M" "(null)" "M" "(null)" ...
##  $ PERP_RACE              : chr [1:2907] "BLACK" "(null)" "BLACK" "(null)" ...
##  $ VIC_AGE_GROUP          : chr [1:2907] "25-44" "18-24" "<18" "18-24" ...
##  $ VIC_SEX                : chr [1:2907] "M" "M" "M" "M" ...
##  $ VIC_RACE               : chr [1:2907] "BLACK" "BLACK" "BLACK" "BLACK" ...
##  $ X_COORD_CD             : num [1:2907] 986050 1016802 1011263 1053494 1021686 ...
##  $ Y_COORD_CD             : num [1:2907] 214231 250581 251671 161531 251947 ...
##  $ Latitude               : num [1:2907] 40.8 40.9 40.9 40.6 40.9 ...
##  $ Longitude              : num [1:2907] -74 -73.9 -73.9 -73.8 -73.9 ...
##  $ Lon_Lat                : chr [1:2907] "POINT (-73.9935 40.754692)" "POINT (-73.88233 40.854402)" "
##  $ Year                   : num [1:2907] 2022 2022 2022 2022 2022 ...
##  $ Month                  : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 5 7 11 8 4 11 12 6 10 2 ..
##  $ DayOfWeek              : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 5 2 4 2 1 2 7 1 5 3 ...
##  $ TimeOfDay              : chr [1:2907] "Night" "Evening" "Evening" "Evening" ...
```

```
summary(nypd_data)
```

```
##   INCIDENT_KEY         OCCUR_DATE          OCCUR_TIME           BORO
##  Min.   :238531159   Min.   :2022-01-01   Length:2907       Length:2907
##  1st Qu.:246192328   1st Qu.:2022-06-06   Class1:hms        Class :character
##  Median :252647955   Median :2022-10-15   Class2:difftime   Mode  :character
##  Mean   :256854604   Mean   :2022-11-24   Mode  :numeric
##  3rd Qu.:268973603   3rd Qu.:2023-05-28
##  Max.   :279758069   Max.   :2023-12-29
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:2907        Min.   :  1.00   Min.   :0.0000    Length:2907
##  Class :character   1st Qu.: 43.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 60.00   Median :0.0000    Mode  :character
##                     Mean   : 62.22   Mean   :0.2425
##                     3rd Qu.: 79.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:2907        Mode :logical           Length:2907
##  Class :character   FALSE:2313              Class :character
##  Mode  :character   TRUE :594               Mode  :character
##
##
##
##
##    PERP_SEX           PERP_RACE         VIC_AGE_GROUP        VIC_SEX
##  Length:2907        Length:2907        Length:2907        Length:2907
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
```

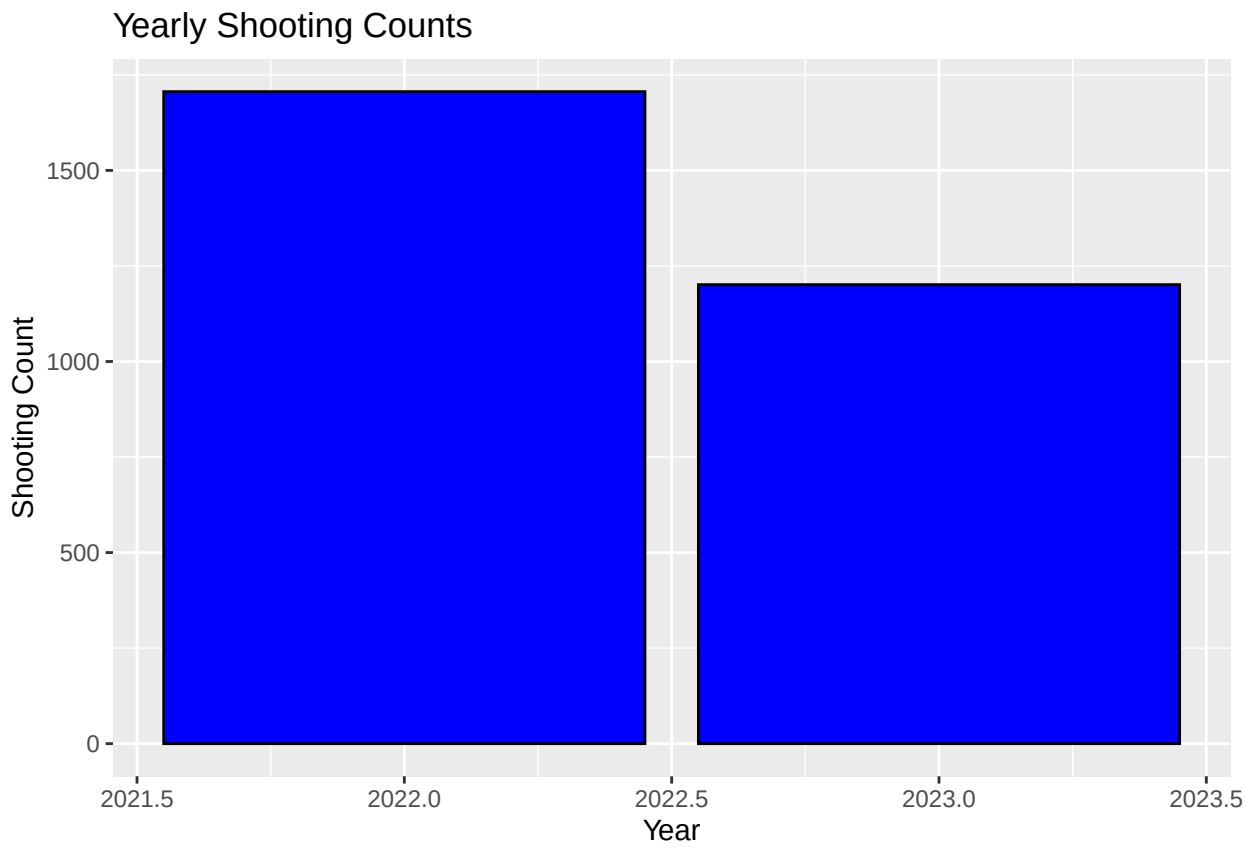
```
##  Length:2907        Min.   : 929510   Min.   :127539   Min.   :40.52
##  Class :character   1st Qu.:1000459   1st Qu.:184337   1st Qu.:40.67
##  Mode  :character   Median :1008366   Median :212367   Median :40.75
##                     Mean   :1009286   Mean   :212612   Mean   :40.75
##                     3rd Qu.:1016743   3rd Qu.:242614   3rd Qu.:40.83
##                     Max.   :1059828   Max.   :269204   Max.   :40.91
##
##    Longitude        Lon_Lat               Year          Month      DayOfWeek
##  Min.   :-74.20   Length:2907        Min.   :2022   Jul    : 375   Sun:502
##  1st Qu.:-73.94   Class :character   1st Qu.:2022   Jun    : 290   Mon:451
##  Median :-73.91   Mode  :character   Median :2022   May    : 277   Tue:370
##  Mean   :-73.91                      Mean   :2022   Mar    : 262   Wed:323
##  3rd Qu.:-73.88                      3rd Qu.:2023   Aug    : 259   Thu:358
##  Max.   :-73.73                      Max.   :2023   Sep    : 254   Fri:372
##                                                     (Other):1190   Sat:531
##   TimeOfDay
##  Length:2907
##  Class :character
##  Mode  :character
##
##
##
##
```

```
head(nypd_data)
```

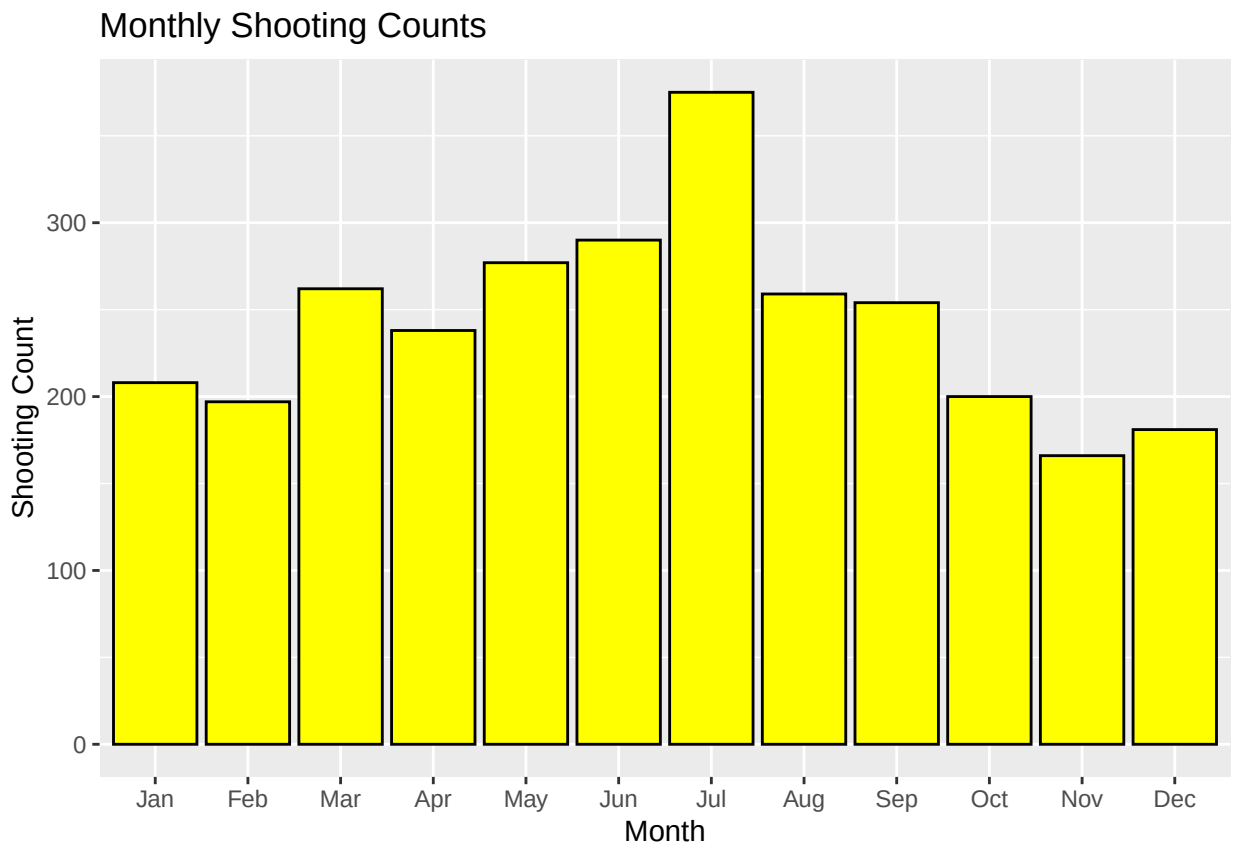
```
## # A tibble: 6 x 25
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO      LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <date>     <time>     <chr>     <chr>                <dbl>
## 1    244608249 2022-05-05 00:10      MANHATTAN INSIDE                  14
## 2    247542571 2022-07-04 22:20      BRONX     OUTSIDE                 48
## 3    254911480 2022-11-30 21:15      BRONX     OUTSIDE                 46
## 4    249623757 2022-08-15 18:21      QUEENS    OUTSIDE                101
## 5    243433246 2022-04-10 17:00      BRONX     OUTSIDE                 49
## 6    253757468 2022-11-07 11:35      BROOKLYN  OUTSIDE                 75
## # i 19 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>, Year <dbl>, Month <ord>, DayOfWeek <ord>,
## #   TimeOfDay <chr>
```

```
# Count of Shootings Per Year
yearly_counts <- nypd_data %>%
  group_by(Year) %>%
  summarise(SHOOTING_COUNT = n())

# Count of Shootings Per Month
monthly_counts <- nypd_data %>%
  group_by(Month) %>%
  summarise(SHOOTING_COUNT = n())

# Count of Shootings Per Day Of Week
```

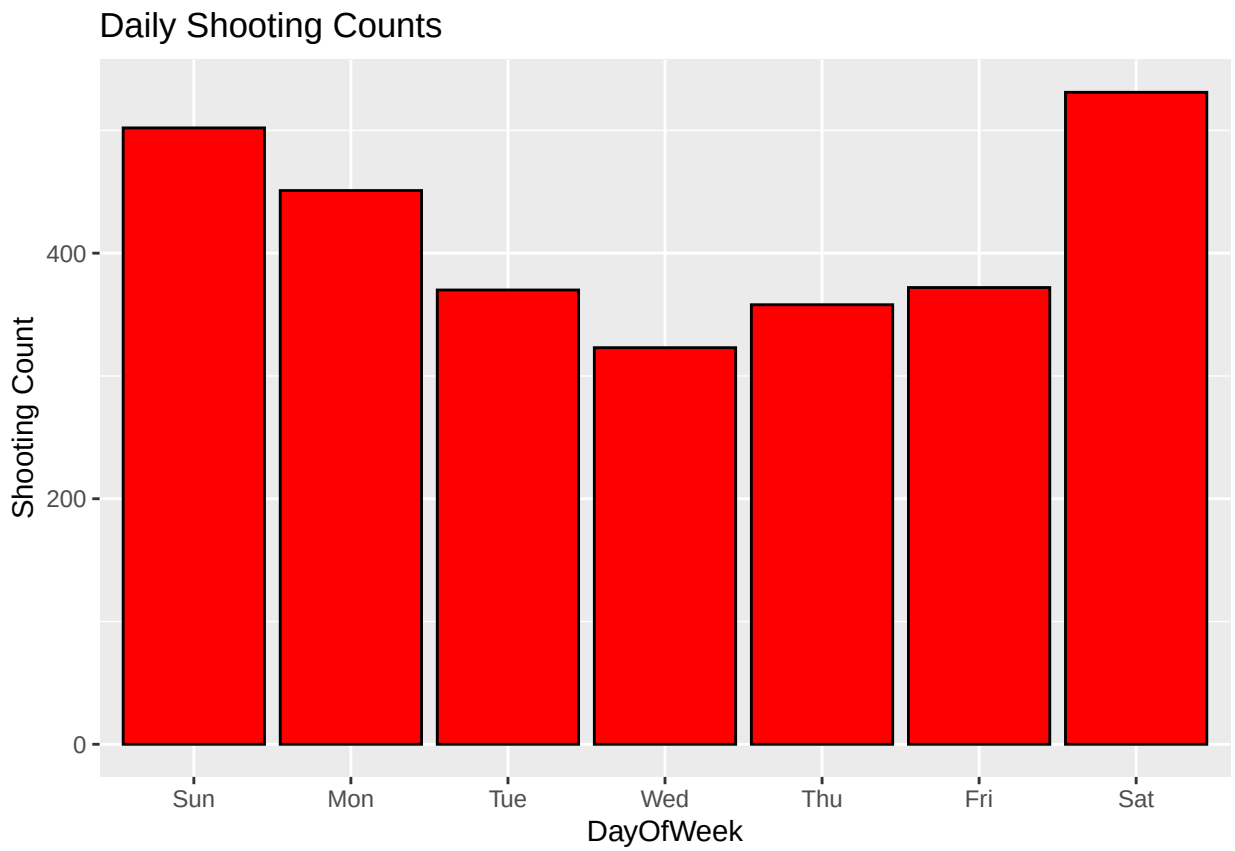
```r
daily_counts <- nypd_data %>%
  group_by(DayOfWeek) %>%
  summarise(SHOOTING_COUNT = n())

# Count of Shootings Per Time Of Day
time_of_day_counts <- nypd_data %>%
  group_by(TimeOfDay) %>%
  summarise(SHOOTING_COUNT = n())

# Count of Shootings Per Borough
borough_counts <- nypd_data %>%
  group_by(BORO) %>%
  summarise(SHOOTING_COUNT = n())


# Yearly Shooting Counts Plot
ggplot(yearly_counts, aes(x = Year, y = SHOOTING_COUNT)) +
  geom_bar(stat = "identity", fill = "blue", color = "black") +
  labs(title = "Yearly Shooting Counts", x = "Year", y = "Shooting Count")
```



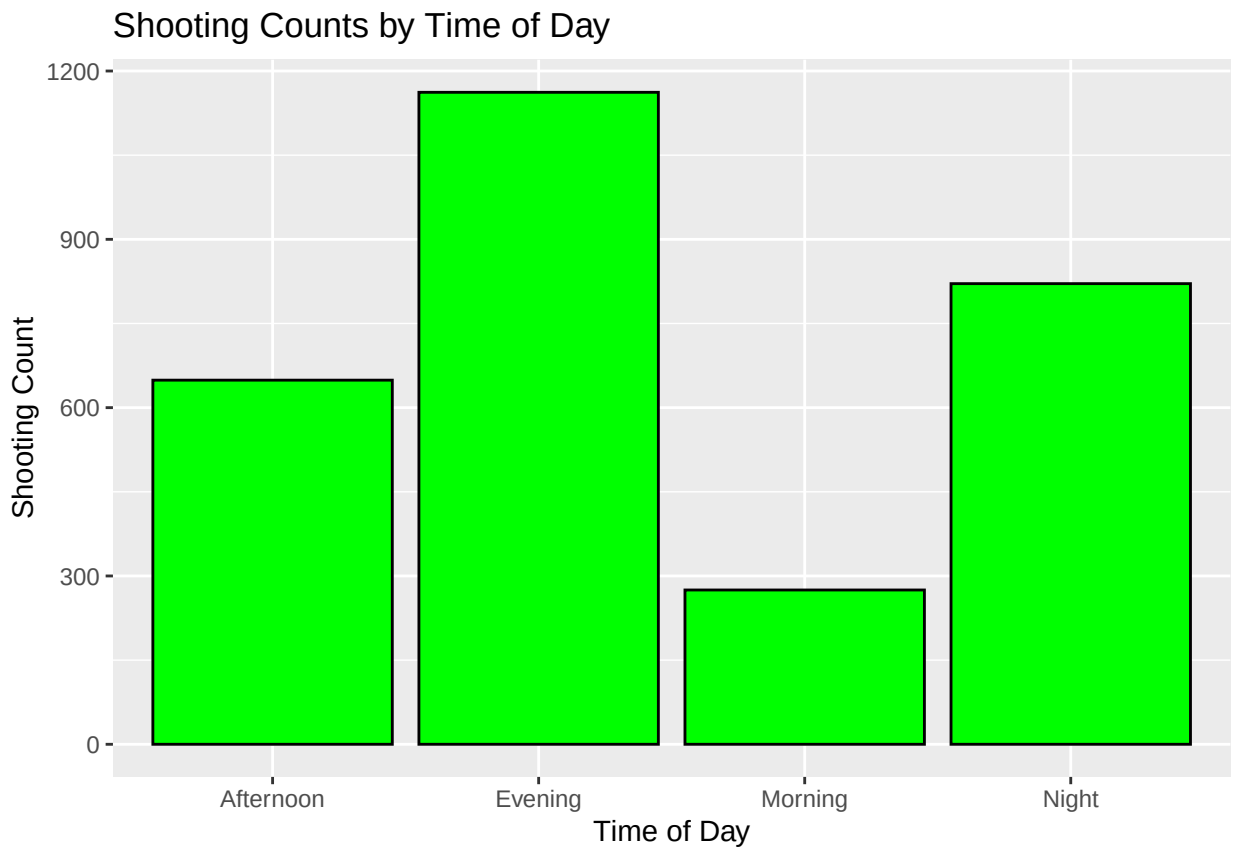

```r
# Monthly Shooting Counts Plot
ggplot(monthly_counts, aes(x = Month, y = SHOOTING_COUNT)) +
  geom_bar(stat = "identity", fill = "yellow", color = "black") +
  labs(title = "Monthly Shooting Counts", x = "Month", y = "Shooting Count")
```



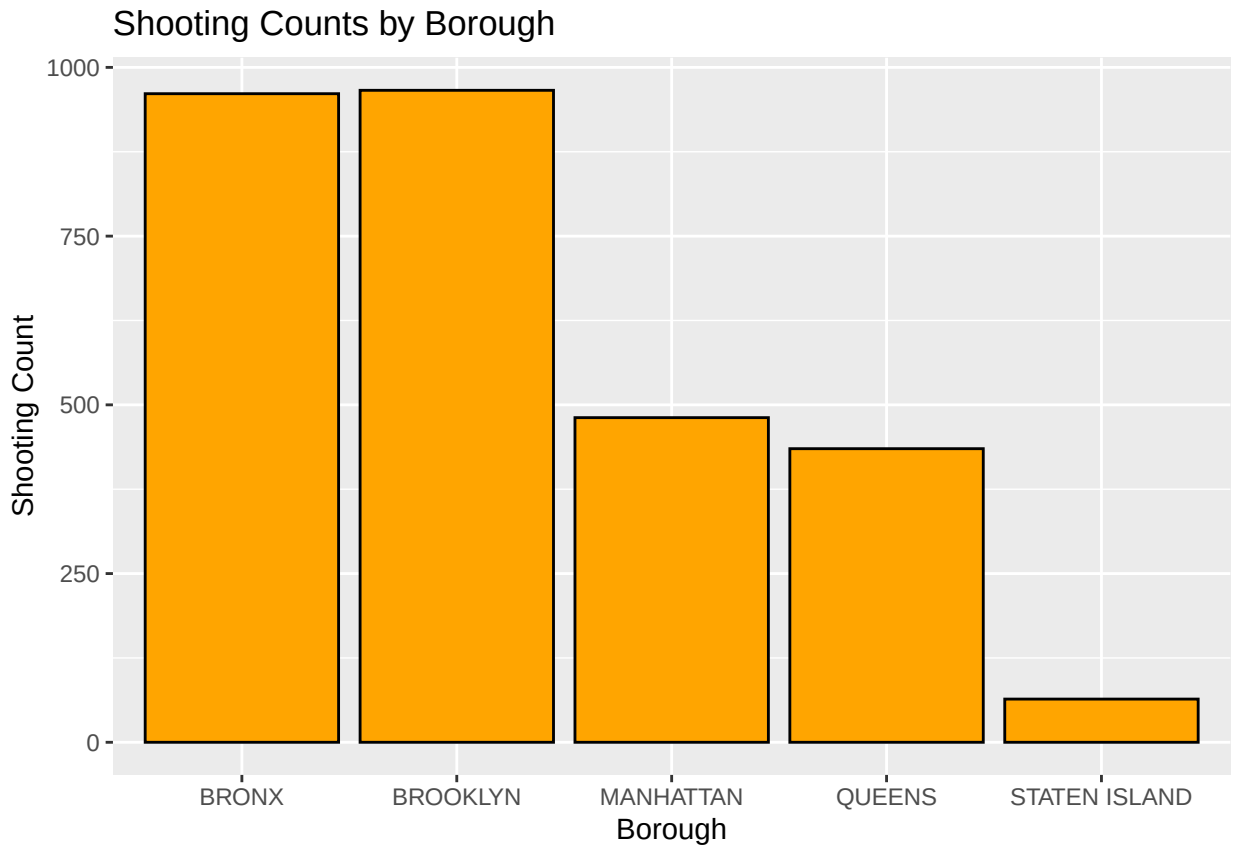

```
# Daily Shooting Counts Plot
ggplot(daily_counts, aes(x = DayOfWeek, y = SHOOTING_COUNT)) +
  geom_bar(stat = "identity", fill = "red", color = "black") +
  labs(title = "Daily Shooting Counts", x = "DayOfWeek", y = "Shooting Count")
```



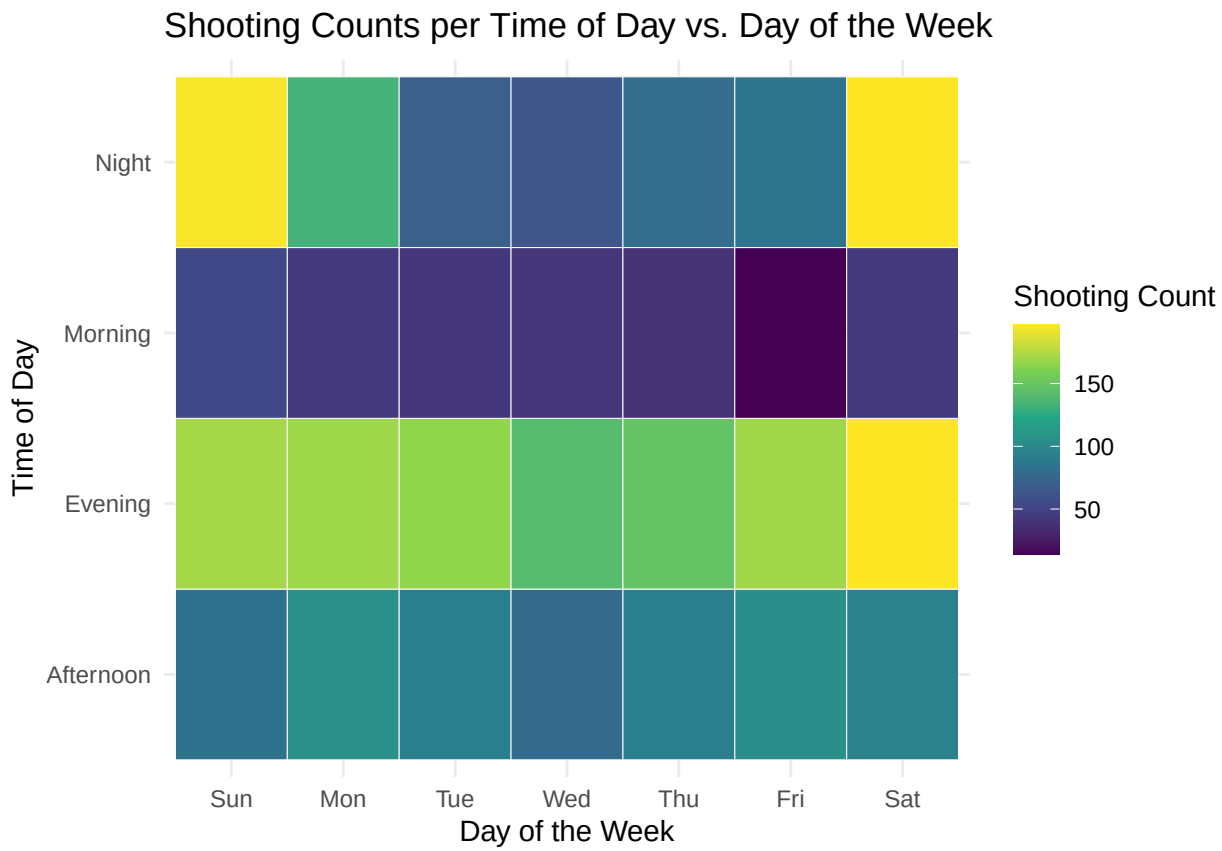

```
# Shooting Counts by Time of Day Plot
ggplot(time_of_day_counts, aes(x = TimeOfDay, y = SHOOTING_COUNT)) +
  geom_bar(stat = "identity", fill = "green", color = "black") +
  labs(title = "Shooting Counts by Time of Day", x = "Time of Day", y = "Shooting Count")
```



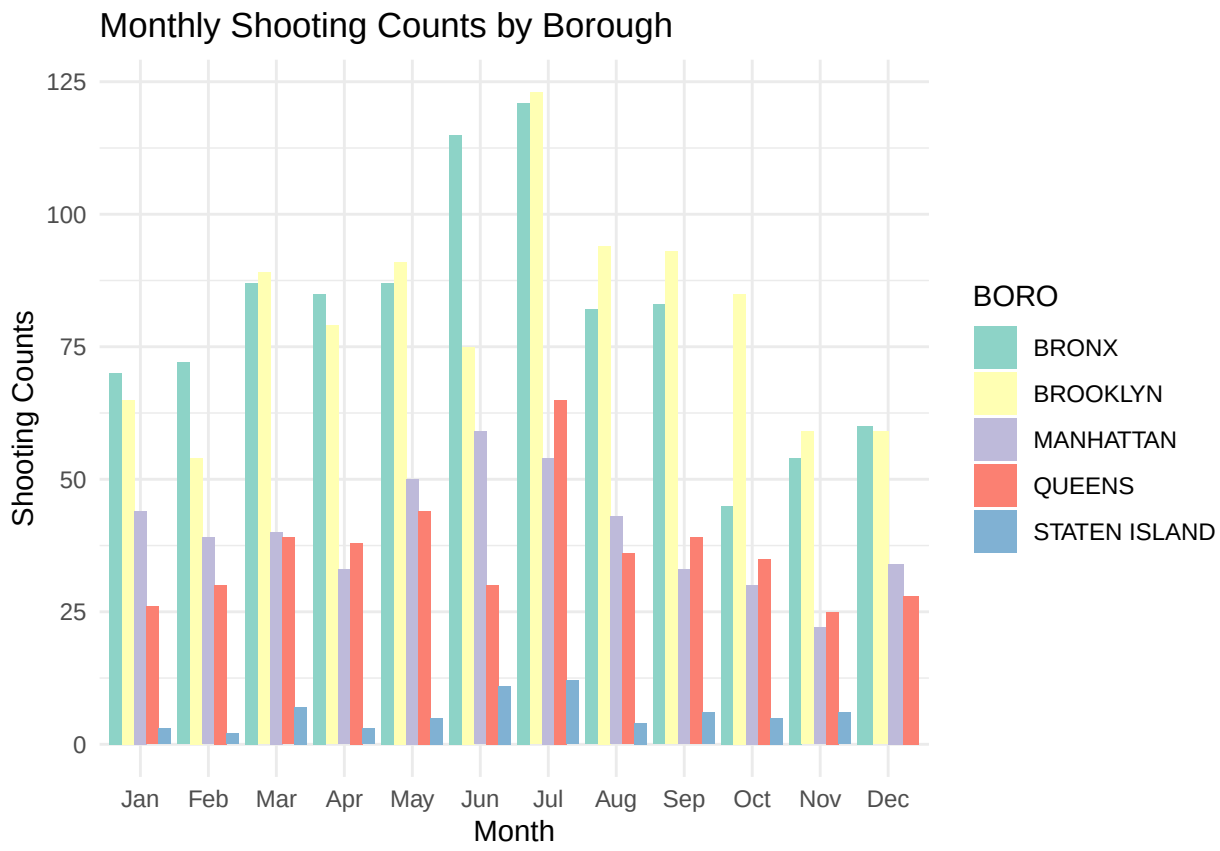

```
# Shooting Counts by Borough Plot
ggplot(borough_counts, aes(x = BORO, y = SHOOTING_COUNT)) +
  geom_bar(stat = "identity", fill = "orange", color = "black") +
  labs(title = "Shooting Counts by Borough", x = "Borough", y = "Shooting Count")
```




```r
# Shooting counts for each time of day and day of the week
shooting_summary <- nypd_data %>%
  group_by(DayOfWeek, TimeOfDay) %>%
  summarise(ShootingCount = n(), .groups = 'drop')

# Create a heatmap of shootings per time of day and day of week
ggplot(shooting_summary, aes(x = factor(DayOfWeek), y = factor(TimeOfDay), fill = ShootingCount)) +
  geom_tile(color = "white") +
  scale_fill_viridis_c() +
  labs(
    title = "Shooting Counts per Time of Day vs. Day of the Week",
    x = "Day of the Week",
    y = "Time of Day",
    fill = "Shooting Count"
  ) +
  scale_x_discrete(labels = c("1" = "Sunday", "2" = "Monday", "3" = "Tuesday", "4" = "Wednesday",
                              "5" = "Thursday", "6" = "Friday", "7" = "Saturday")) +
  scale_y_discrete(labels = c("1" = "Morning", "2" = "Afternoon", "3" = "Evening", "4" = "Night")) +
  theme_minimal()
```



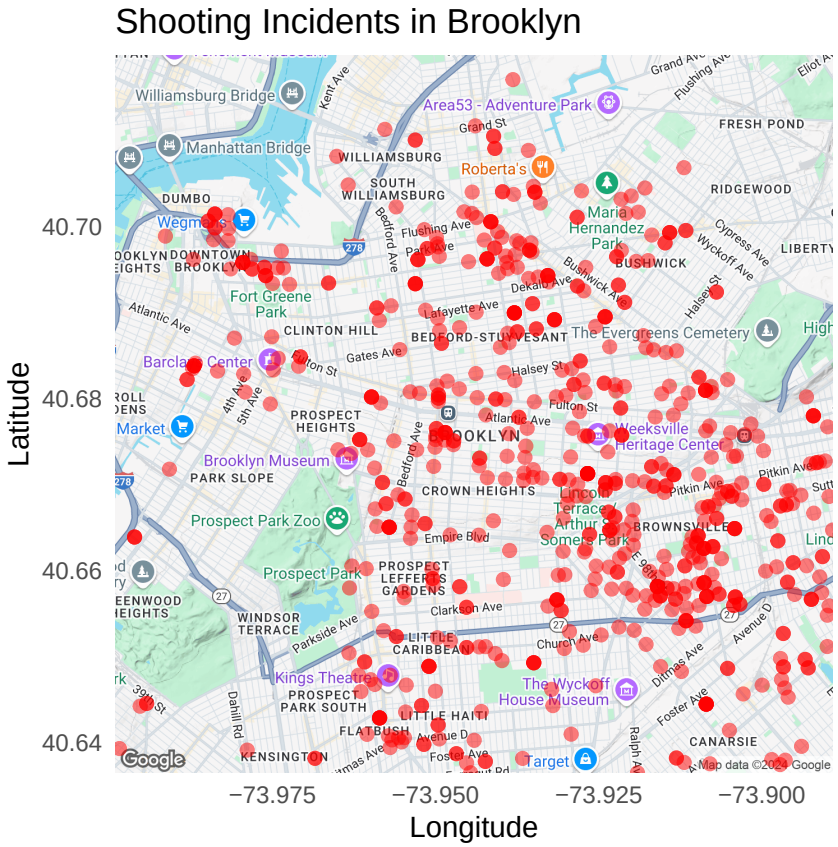

```r
# Count of Monthly Shootings by Borough
monthly_borough_trends <- nypd_data %>%
  group_by(Month, BORO) %>%
  summarise(SHOOTING_COUNT = n(), .groups = "drop") # Explicitly drop grouping after summarise

# Plotting monthly shooting counts by borough
ggplot(monthly_borough_trends, aes(x = Month, y = SHOOTING_COUNT, fill = BORO)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Monthly Shooting Counts by Borough", x = "Month", y = "Shooting Counts") +
  scale_fill_brewer(palette = "Set3") +
  theme_minimal()
```



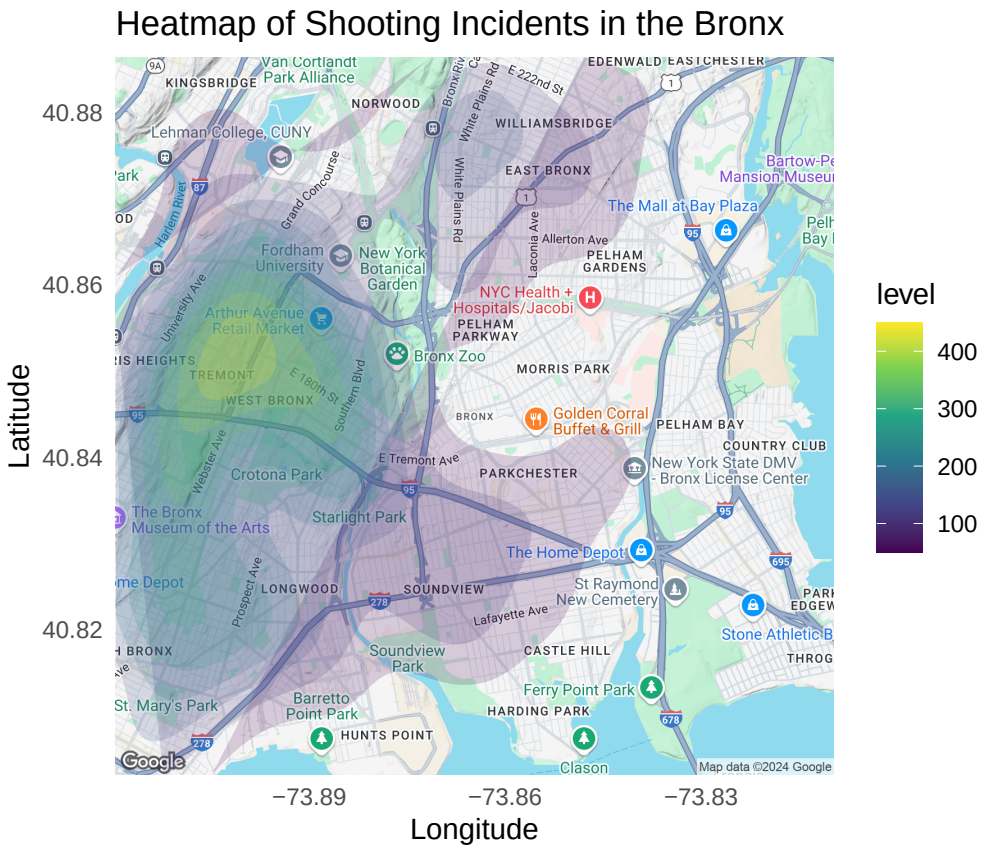

```
## i <https://maps.googleapis.com/maps/api/staticmap?center=New%20York%20City&zoom=11&size=640x640&scal

## i <https://maps.googleapis.com/maps/api/geocode/json?address=New+York+City&key=xxx>

## i <https://maps.googleapis.com/maps/api/staticmap?center=40.8448,-73.8648&zoom=13&size=640x640&scale

## i <https://maps.googleapis.com/maps/api/staticmap?center=40.6782,-73.9442&zoom=13&size=640x640&scale

## Warning: Removed 228 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


```
## Warning: Removed 142 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

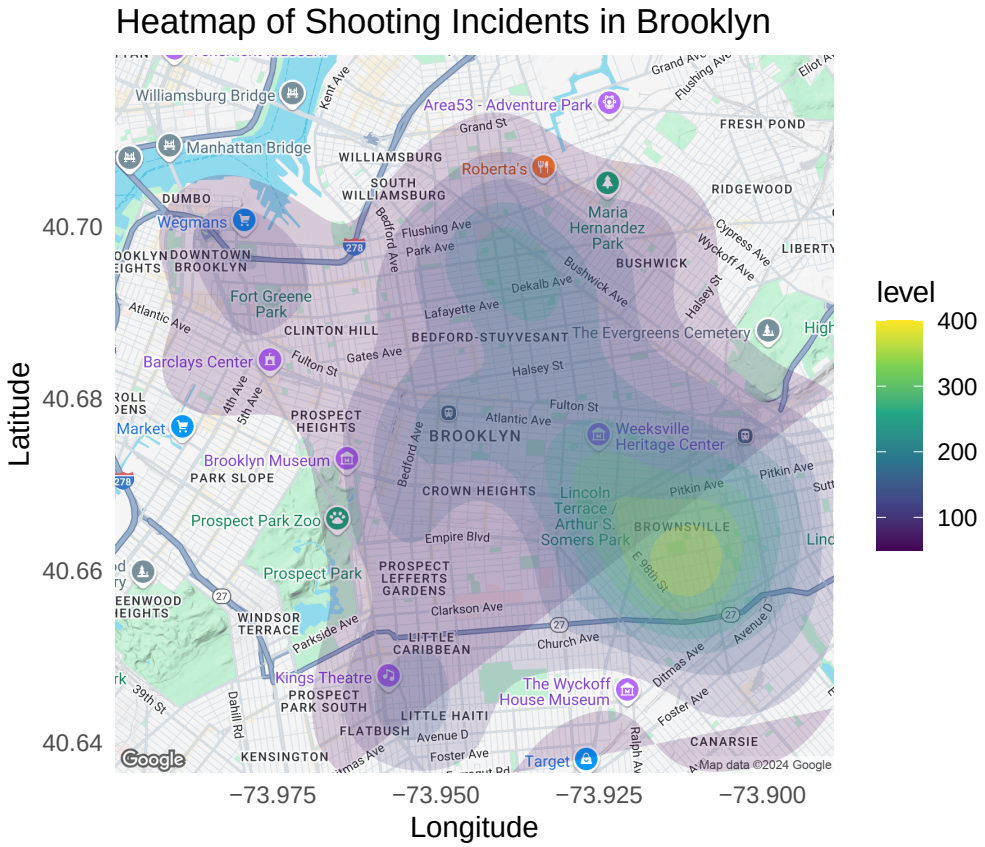

```
## Warning: Removed 278 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
## Warning: The dot-dot notation (‘..level..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(level)‘ instead.
## i The deprecated feature was likely used in the ggmap package.
##   Please report the issue at <https://github.com/dkahle/ggmap/issues>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning: Removed 142 rows containing non-finite outside the scale range
## (‘stat_density2d()‘).
```




```
## Warning: Removed 278 rows containing non-finite outside the scale range
## (‘stat_density2d()‘).
```

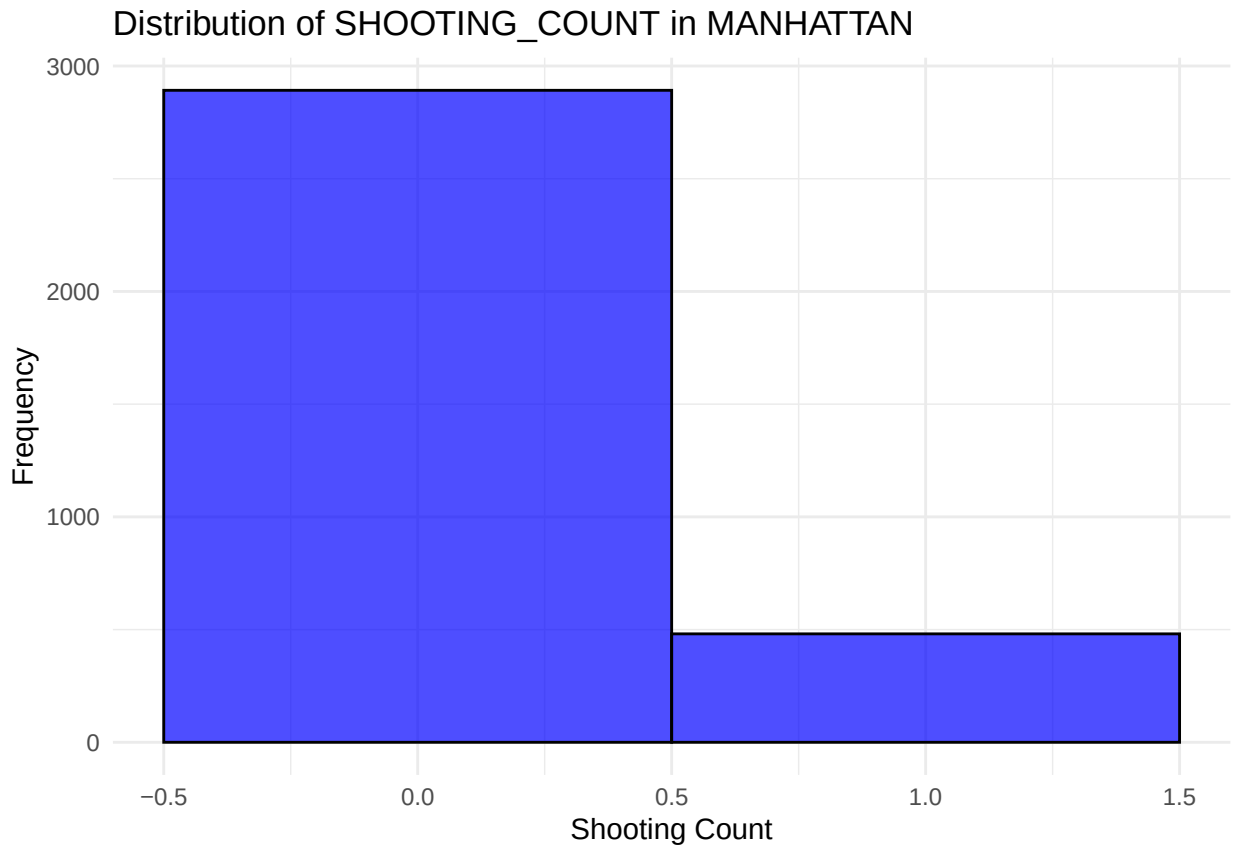



```r
# Define the most densely populated coordinates for each borough
densest_coordinates <- list(
  "MANHATTAN" = c(-73.9851, 40.7580),
  "BROOKLYN" = c(-73.9496, 40.6501),
  "QUEENS" = c(-73.8317, 40.7282),
  "BRONX" = c(-73.8648, 40.8448),
  "STATEN ISLAND" = c(-74.1502, 40.5795)
)

# Function to fill in zero-shooting rows for each unique combination of OCCUR_DATE and TimeOfDay
fill_zero_shooting_rows <- function(df, borough_name) {
  full_dates <- seq(min(df$OCCUR_DATE, na.rm = TRUE), max(df$OCCUR_DATE, na.rm = TRUE), by = "day")
  time_of_day_levels <- c("Morning", "Afternoon", "Evening", "Night")

  zero_shooting_df <- expand.grid(
    OCCUR_DATE = full_dates,
    TimeOfDay = time_of_day_levels
  ) %>%
    mutate(
      BORO = borough_name,
      Year = year(OCCUR_DATE),
      Month = factor(month(OCCUR_DATE, label = TRUE, abbr = TRUE), levels = month.abb),
      DayOfWeek = factor(wday(OCCUR_DATE, label = TRUE, abbr = TRUE), levels = c("Sun", "Mon", "Tue", "
      SHOOTING_COUNT = 0,
      Longitude = densest_coordinates[[borough_name]][1],
      Latitude = densest_coordinates[[borough_name]][2]
```

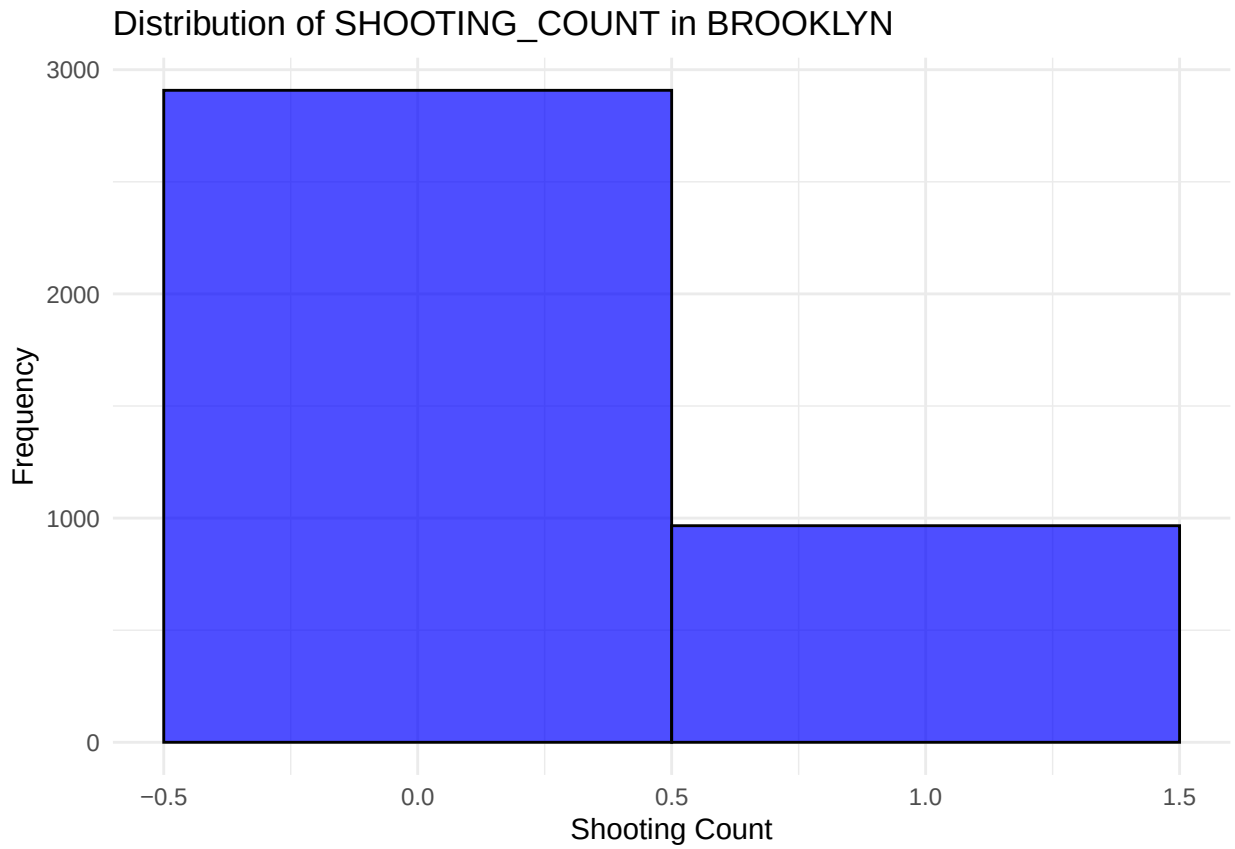
```
    )

  # Merge with original data and fill missing combinations with zero-shooting rows
  combined_df <- full_join(df, zero_shooting_df, by = c("OCCUR_DATE", "TimeOfDay", "BORO", "Year", "Mon
  combined_df <- combined_df %>%
    mutate(SHOOTING_COUNT = ifelse(is.na(SHOOTING_COUNT.x), SHOOTING_COUNT.y, SHOOTING_COUNT.x)) %>%
    select(-c(SHOOTING_COUNT.x, SHOOTING_COUNT.y)) %>%
    arrange(OCCUR_DATE, TimeOfDay)

  return(combined_df)
}

# Main function to process each borough without aggregating data
process_borough_data <- function(df, borough_name) {
  shooting_df <- df %>%
    filter(BORO == borough_name)

  if (!"SHOOTING_COUNT" %in% colnames(shooting_df)) {
    shooting_df$SHOOTING_COUNT <- 1  # Default to 1 for rows indicating shootings
  }

  expanded_df <- fill_zero_shooting_rows(shooting_df, borough_name)

  expanded_df <- expanded_df %>%
    mutate(
      Year_Binary = ifelse(Year == min(Year, na.rm = TRUE), 0, 1),
      TimeOfDay_Num = as.numeric(factor(TimeOfDay, levels = c("Morning", "Afternoon", "Evening", "Night
      TimeOfDay_Sin = sin(2 * pi * TimeOfDay_Num / 4),
      TimeOfDay_Cos = cos(2 * pi * TimeOfDay_Num / 4),
      Month_Num = as.numeric(Month),
      Month_Sin = sin(2 * pi * Month_Num / 12),
      Month_Cos = cos(2 * pi * Month_Num / 12),
      DayOfWeek_Num = as.numeric(DayOfWeek),
      DayOfWeek_Sin = sin(2 * pi * DayOfWeek_Num / 7),
      DayOfWeek_Cos = cos(2 * pi * DayOfWeek_Num / 7)
    )

  return(expanded_df)
}

# Plot the distribution of SHOOTING_COUNT for each borough
plot_shooting_count_distribution <- function(borough_data, borough_name) {
  ggplot(borough_data, aes(x = SHOOTING_COUNT)) +
    geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
    labs(
      title = paste("Distribution of SHOOTING_COUNT in", borough_name),
      x = "Shooting Count",
      y = "Frequency"
    ) +
    theme_minimal()
}

# Process each borough and store results without aggregation
```

```r
borough_datasets <- lapply(names(densest_coordinates), function(borough) {
  process_borough_data(nypd_data, borough)
})
names(borough_datasets) <- names(densest_coordinates)

# Display the head of each borough's dataset to ensure it's ordered by day and time of day
for (borough in names(borough_datasets)) {
  cat("\nShowing head of the dataset for", borough, "ordered by day and time of day:\n")
  ordered_df <- borough_datasets[[borough]] %>%
    arrange(OCCUR_DATE, TimeOfDay)
  print(head(ordered_df, 10))
  df <- borough_datasets[[borough]]
  cat("\nPlotting distribution for", borough, ":\n")
  print(plot_shooting_count_distribution(df, borough))

}
```

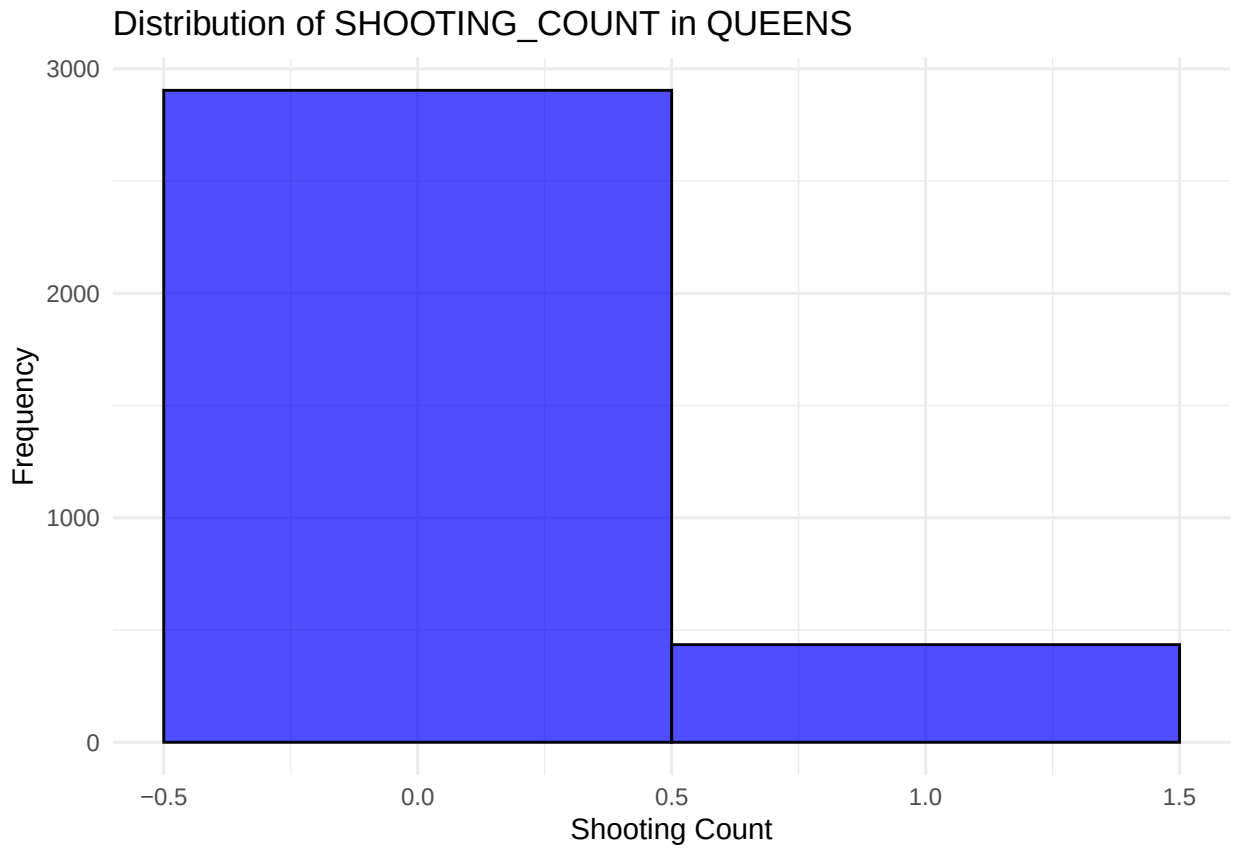
```
##
## Showing head of the dataset for MANHATTAN ordered by day and time of day:
## # A tibble: 10 x 36
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO      LOC_OF_OCCUR_DESC PRECINCT
##           <dbl> <date>     <time>     <chr>     <chr>                <dbl>
##  1           NA 2022-01-01    NA      MANHATTAN <NA>                    NA
##  2           NA 2022-01-01    NA      MANHATTAN <NA>                    NA
##  3    238531161 2022-01-01 06:59      MANHATTAN OUTSIDE                 34
##  4    238533195 2022-01-01 06:15      MANHATTAN OUTSIDE                 25
##  5           NA 2022-01-01    NA      MANHATTAN <NA>                    NA
##  6    238531159 2022-01-01 01:12      MANHATTAN OUTSIDE                 34
##  7    238532487 2022-01-01 05:45      MANHATTAN OUTSIDE                 23
##  8    238531160 2022-01-01 05:20      MANHATTAN OUTSIDE                 10
##  9           NA 2022-01-01    NA      MANHATTAN <NA>                    NA
## 10           NA 2022-01-02    NA      MANHATTAN <NA>                    NA
## # i 30 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>, Year <dbl>, Month <ord>, DayOfWeek <ord>,
## #   TimeOfDay <chr>, SHOOTING_COUNT <dbl>, Year_Binary <dbl>,
## #   TimeOfDay_Num <dbl>, TimeOfDay_Sin <dbl>, TimeOfDay_Cos <dbl>, ...
##
## Plotting distribution for MANHATTAN :
```



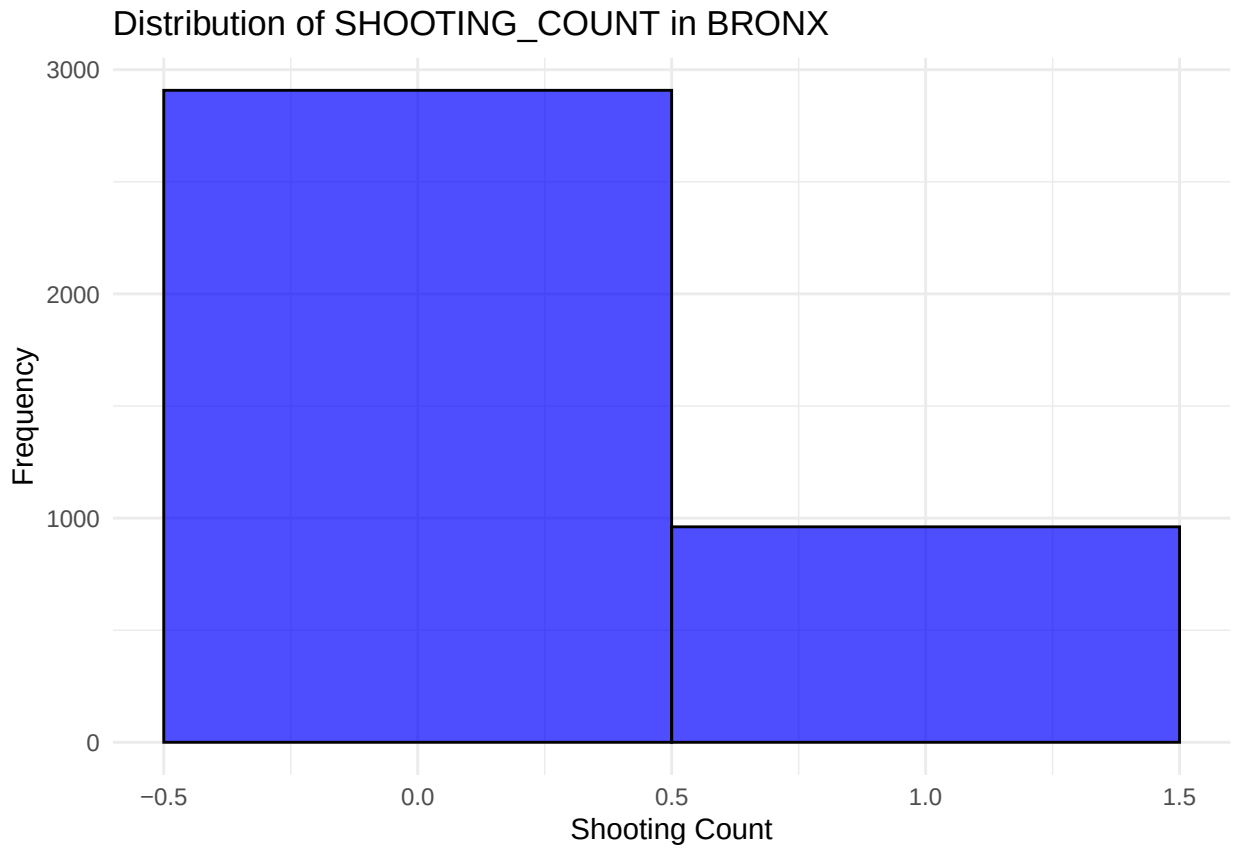

```
##
## Showing head of the dataset for BROOKLYN ordered by day and time of day:
## # A tibble: 10 x 36
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     LOC_OF_OCCUR_DESC PRECINCT
##           <dbl> <date>     <time>     <chr>    <chr>                <dbl>
##  1           NA 2022-01-02    NA      BROOKLYN <NA>                    NA
##  2    238555659 2022-01-02 21:45      BROOKLYN INSIDE                  79
##  3           NA 2022-01-02    NA      BROOKLYN <NA>                    NA
##  4           NA 2022-01-02    NA      BROOKLYN <NA>                    NA
##  5           NA 2022-01-02    NA      BROOKLYN <NA>                    NA
##  6           NA 2022-01-03    NA      BROOKLYN <NA>                    NA
##  7           NA 2022-01-03    NA      BROOKLYN <NA>                    NA
##  8           NA 2022-01-03    NA      BROOKLYN <NA>                    NA
##  9    238561496 2022-01-03 00:35      BROOKLYN OUTSIDE                 71
## 10           NA 2022-01-03    NA      BROOKLYN <NA>                    NA
## # i 30 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>, Year <dbl>, Month <ord>, DayOfWeek <ord>,
## #   TimeOfDay <chr>, SHOOTING_COUNT <dbl>, Year_Binary <dbl>,
## #   TimeOfDay_Num <dbl>, TimeOfDay_Sin <dbl>, TimeOfDay_Cos <dbl>, ...
##
## Plotting distribution for BROOKLYN :
```



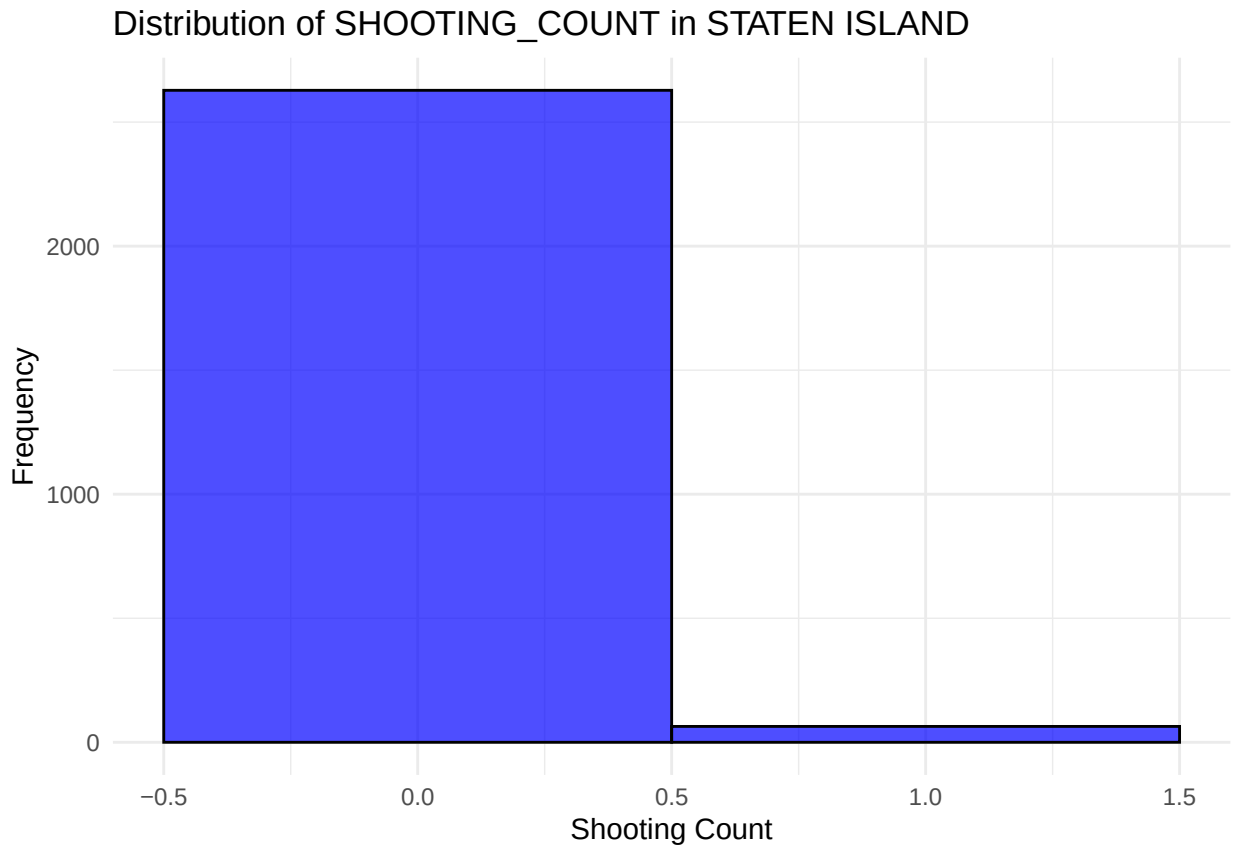

```
##
## Showing head of the dataset for QUEENS ordered by day and time of day:
## # A tibble: 10 x 36
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO   LOC_OF_OCCUR_DESC PRECINCT
##           <dbl> <date>     <time>     <chr>  <chr>                <dbl>
##  1           NA 2022-01-03    NA      QUEENS <NA>                    NA
##  2    238592807 2022-01-03 21:29      QUEENS OUTSIDE                112
##  3           NA 2022-01-03    NA      QUEENS <NA>                    NA
##  4           NA 2022-01-03    NA      QUEENS <NA>                    NA
##  5           NA 2022-01-03    NA      QUEENS <NA>                    NA
##  6           NA 2022-01-04    NA      QUEENS <NA>                    NA
##  7           NA 2022-01-04    NA      QUEENS <NA>                    NA
##  8           NA 2022-01-04    NA      QUEENS <NA>                    NA
##  9           NA 2022-01-04    NA      QUEENS <NA>                    NA
## 10           NA 2022-01-05    NA      QUEENS <NA>                    NA
## # i 30 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>, Year <dbl>, Month <ord>, DayOfWeek <ord>,
## #   TimeOfDay <chr>, SHOOTING_COUNT <dbl>, Year_Binary <dbl>,
## #   TimeOfDay_Num <dbl>, TimeOfDay_Sin <dbl>, TimeOfDay_Cos <dbl>, ...
##
## Plotting distribution for QUEENS :
```




```
##
## Showing head of the dataset for BRONX ordered by day and time of day:
## # A tibble: 10 x 36
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO  LOC_OF_OCCUR_DESC PRECINCT
##           <dbl> <date>     <time>     <chr> <chr>                <dbl>
##  1    238555660 2022-01-02 17:14      BRONX OUTSIDE                 47
##  2    238555360 2022-01-02 15:52      BRONX OUTSIDE                 49
##  3           NA 2022-01-02    NA      BRONX <NA>                    NA
##  4           NA 2022-01-02    NA      BRONX <NA>                    NA
##  5           NA 2022-01-02    NA      BRONX <NA>                    NA
##  6           NA 2022-01-02    NA      BRONX <NA>                    NA
##  7           NA 2022-01-03    NA      BRONX <NA>                    NA
##  8           NA 2022-01-03    NA      BRONX <NA>                    NA
##  9    238592450 2022-01-03 10:20      BRONX INSIDE                  44
## 10           NA 2022-01-03    NA      BRONX <NA>                    NA
## # i 30 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>, Year <dbl>, Month <ord>, DayOfWeek <ord>,
## #   TimeOfDay <chr>, SHOOTING_COUNT <dbl>, Year_Binary <dbl>,
## #   TimeOfDay_Num <dbl>, TimeOfDay_Sin <dbl>, TimeOfDay_Cos <dbl>, ...
##
## Plotting distribution for BRONX :
```




```
##
## Showing head of the dataset for STATEN ISLAND ordered by day and time of day:
## # A tibble: 10 x 36
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO          LOC_OF_OCCUR_DESC PRECINCT
##           <dbl> <date>     <time>     <chr>         <chr>                <dbl>
##  1           NA 2022-01-16    NA      STATEN ISLAND <NA>                    NA
##  2           NA 2022-01-16    NA      STATEN ISLAND <NA>                    NA
##  3           NA 2022-01-16    NA      STATEN ISLAND <NA>                    NA
##  4    239214374 2022-01-16 02:12      STATEN ISLAND OUTSIDE                120
##  5           NA 2022-01-16    NA      STATEN ISLAND <NA>                    NA
##  6           NA 2022-01-17    NA      STATEN ISLAND <NA>                    NA
##  7           NA 2022-01-17    NA      STATEN ISLAND <NA>                    NA
##  8           NA 2022-01-17    NA      STATEN ISLAND <NA>                    NA
##  9           NA 2022-01-17    NA      STATEN ISLAND <NA>                    NA
## 10           NA 2022-01-18    NA      STATEN ISLAND <NA>                    NA
## # i 30 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>, Year <dbl>, Month <ord>, DayOfWeek <ord>,
## #   TimeOfDay <chr>, SHOOTING_COUNT <dbl>, Year_Binary <dbl>,
## #   TimeOfDay_Num <dbl>, TimeOfDay_Sin <dbl>, TimeOfDay_Cos <dbl>, ...
##
## Plotting distribution for STATEN ISLAND :
```




```r
# Function to select relevant columns
select_relevant_features <- function(df) {
  selected_df <- df %>%
    select(
      Latitude, Longitude, SHOOTING_COUNT, TimeOfDay_Sin, TimeOfDay_Cos,
      Month_Sin, Month_Cos, DayOfWeek_Sin, DayOfWeek_Cos
    ) %>%
    drop_na()
  return(selected_df)
}

# Function to add interaction and lagged features
add_features <- function(df) {
  if ("TimeOfDay_Sin" %in% colnames(df) && "DayOfWeek_Sin" %in% colnames(df)) {
    # Create a numeric interaction by multiplying TimeOfDay_Sin and DayOfWeek_Sin
    df <- df %>%
      mutate(TimeOfDay_DayOfWeek_Interaction = TimeOfDay_Sin * DayOfWeek_Sin)
  }
  return(df)
}
```

```r
# Function to split data
split_train_test <- function(df, train_ratio = 0.8) {
  # Create a random sample of indices for the training data
  train_indices <- sample(1:nrow(df), size = floor(train_ratio * nrow(df)))
```

```
  # Training set
  x_train <- df[train_indices, ] %>% select(-SHOOTING_COUNT)
  y_train <- df[train_indices, "SHOOTING_COUNT", drop = TRUE]

  # Testing set (the rest of the data not in the training set)
  x_test <- df[-train_indices, ] %>% select(-SHOOTING_COUNT)
  y_test <- df[-train_indices, "SHOOTING_COUNT", drop = TRUE]

  # Return a list of train and test data
  return(list(x_train = x_train, y_train = y_train, x_test = x_test, y_test = y_test))
}


# Function to scale data
scale_data <- function(x_train, x_test) {
  # Calculate the mean and standard deviation of the training data
  train_mean <- colMeans(x_train)
  train_std <- apply(x_train, 2, sd)

  # Scale the training data
  x_train_scaled <- scale(x_train, center = train_mean, scale = train_std)

  # Scale the test data using the same mean and standard deviation as the training data
  x_test_scaled <- scale(x_test, center = train_mean, scale = train_std)

  return(list(x_train_scaled = x_train_scaled, x_test_scaled = x_test_scaled))
}
```

```
calculate_classification_metrics <- function(y_true, y_pred) {
  confusion <- confusionMatrix(y_pred, y_true)
  accuracy <- confusion$overall['Accuracy']
  precision <- confusion$byClass['Pos Pred Value']
  recall <- confusion$byClass['Sensitivity']
  f1_score <- 2 * ((precision * recall) / (precision + recall))

  data.frame(
    Accuracy = accuracy,
    Precision = precision,
    Recall = recall,
    F1_Score = f1_score
  )
}

# Function to plot ROC curve using ROCR
plot_roc_curve <- function(y_test, test_pred_prob, title = "ROC Curve") {
  # Ensure y_test and test_pred_prob are numeric
  y_test <- as.numeric(as.character(y_test))
  test_pred_prob <- as.numeric(test_pred_prob)

  # Create ROCR prediction and performance objects
  pred <- prediction(test_pred_prob, y_test)
  perf <- performance(pred, "tpr", "fpr")
```

```
  # Plot the ROC curve
  plot(perf, col = "blue", main = title)
  auc <- performance(pred, "auc")
  auc_value <- auc@y.values[[1]]
  legend("bottomright", legend = paste("AUC =", round(auc_value, 3)), col = "blue", lwd = 2)
}


plot_actual_vs_predicted <- function(y_true, y_pred, borough, title = "Actual vs Predicted") {
  # Create a confusion matrix table
  confusion_table <- table(Predicted = y_pred, Actual = y_true)

  # Plot the confusion matrix as a mosaic plot
  mosaicplot(
    confusion_table,
    main = paste(borough, ":", title),
    xlab = "Actual",
    ylab = "Predicted",
    color = TRUE,
    las = 1
  )
}


# After training your model and generating predictions:
evaluate_models <- function(y_true, y_pred, y_pred_prob, borough) {
  # Calculate and print metrics with borough name
  cat("\nMetrics for", borough, ":\n")
  metrics <- calculate_classification_metrics(y_true, y_pred)
  print(metrics)

  # Plot actual vs predicted with borough label
  plot_actual_vs_predicted(y_true, y_pred, borough, "Logistic Model: Actual vs Predicted")

  # Plot ROC curve with `ROCR`
  plot_roc_curve(y_true, y_pred_prob, paste(borough, ": Logistic Model ROC Curve"))
}
```

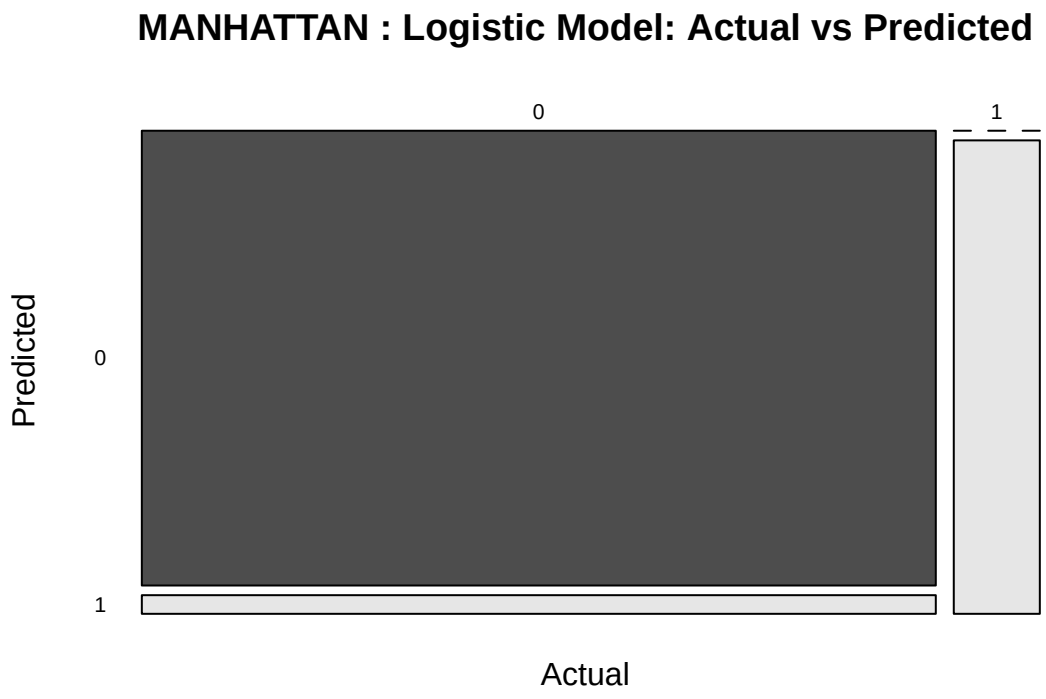
```
# Apply the process for each borough with added checks
for (boro in names(borough_datasets)) {
  cat("\nProcessing dataset for", boro, ":\n")
  df <- borough_datasets[[boro]]

  # Preprocess data
  df <- select_relevant_features(df)
  df <- add_features(df)

  # Split and scale data
  split_data <- split_train_test(df)
  scaled_data <- scale_data(split_data$x_train, split_data$x_test)

  preds <- train_predict_models(scaled_data$x_train_scaled, split_data$y_train, scaled_data$x_test_scal
```

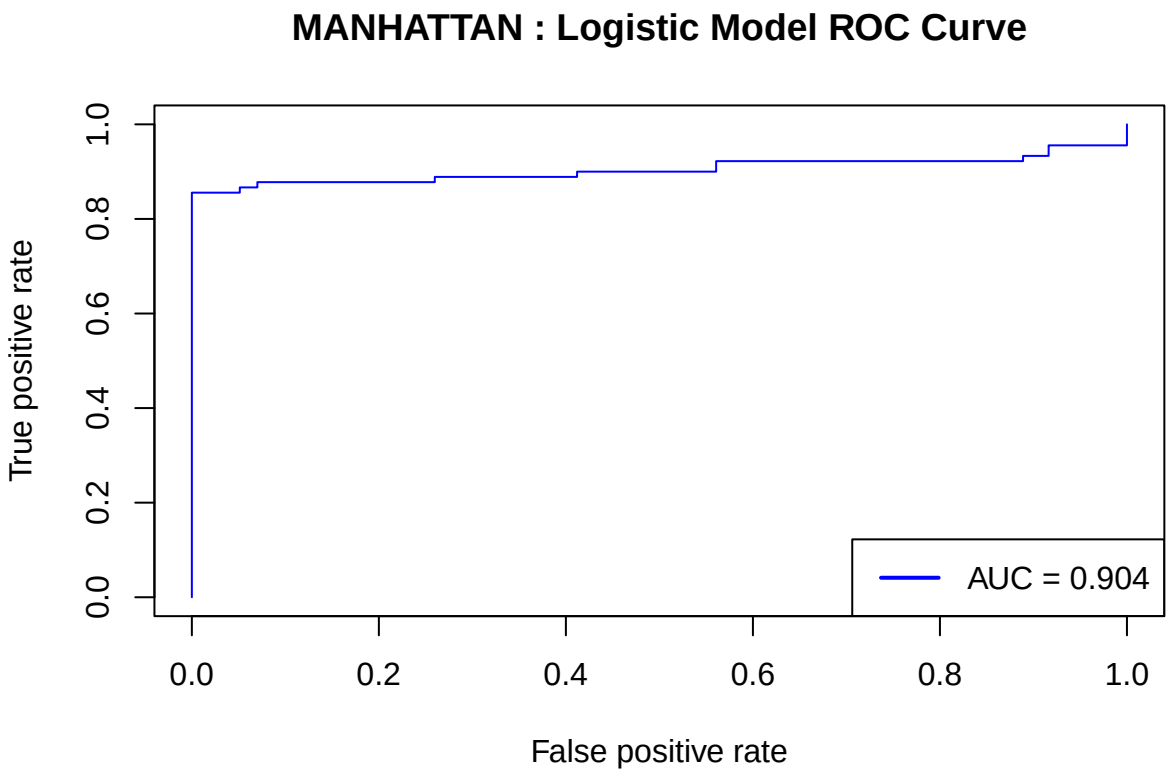
```r
  # Ensure y_test and y_pred are vectors
  y_test_vector <- factor(split_data$y_test)
  y_pred_vector <- factor(preds$test_pred_binary)

  # Align levels between y_test_vector and y_pred_vector
  common_levels <- union(levels(y_test_vector), levels(y_pred_vector))
  y_test_vector <- factor(y_test_vector, levels = common_levels)
  y_pred_vector <- factor(y_pred_vector, levels = common_levels)

  # Keep the probabilities as numeric
  y_pred_prob_vector <- preds$test_pred_prob

  # Evaluate models with borough name included
  eval <- evaluate_models(y_test_vector, y_pred_vector, y_pred_prob_vector, boro)
}
```

```
##
## Processing dataset for MANHATTAN :
##
## Full Model AIC: 828.5327
## Full Model BIC: 887.5354
##
## Calculating VIF for Multicollinearity:
##                         Latitude                        Longitude
##                         4.942633                         4.988282
##                    TimeOfDay_Sin                    TimeOfDay_Cos
##                         1.040347                         1.066656
##                        Month_Sin                        Month_Cos
##                         1.003568                         1.014192
##                    DayOfWeek_Sin                    DayOfWeek_Cos
##                         1.155738                         1.014823
## TimeOfDay_DayOfWeek_Interaction
##                         1.166203
##
## ANOVA Analysis:
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: SHOOTING_COUNT
##
## Terms added sequentially (first to last)
##
##
##                                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                            2697    2232.86
## Latitude                        1  1002.94      2696    1229.92 < 2.2e-16 ***
## Longitude                       1   379.13      2695     850.78 < 2.2e-16 ***
## TimeOfDay_Sin                   1    12.63      2694     838.15 0.0003792 ***
## TimeOfDay_Cos                   1    16.07      2693     822.08  6.11e-05 ***
## Month_Sin                       1     0.83      2692     821.25 0.3609553
## Month_Cos                       1     3.01      2691     818.24 0.0826406 .
## DayOfWeek_Sin                   1     3.37      2690     814.86 0.0662127 .
```

```
## DayOfWeek_Cos                     1     6.26      2689     808.60 0.0123641 *
## TimeOfDay_DayOfWeek_Interaction  1     0.07      2688     808.53 0.7903939
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Performing Backward Feature Selection...
##
## Selected Model AIC: 825.4429
## Selected Model BIC: 872.645
##
## Selected Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     TimeOfDay_Cos + Month_Cos + DayOfWeek_Sin + DayOfWeek_Cos,
##     family = binomial(link = "logit"), data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.4854     0.1131 -21.971  < 2e-16 ***
## Latitude       -1.8701     0.2152  -8.689  < 2e-16 ***
## Longitude       4.4632     0.3137  14.227  < 2e-16 ***
## TimeOfDay_Sin  -0.4059     0.1109  -3.659 0.000253 ***
## TimeOfDay_Cos   0.4112     0.1063   3.868 0.000110 ***
## Month_Cos      -0.1684     0.1019  -1.653 0.098334 .
## DayOfWeek_Sin   0.1932     0.1029   1.878 0.060390 .
## DayOfWeek_Cos   0.2582     0.1039   2.485 0.012938 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2232.86  on 2697  degrees of freedom
## Residual deviance:  809.44  on 2690  degrees of freedom
## AIC: 825.44
##
## Number of Fisher Scoring iterations: 7
##
##
## Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     TimeOfDay_Cos + Month_Cos + DayOfWeek_Sin + DayOfWeek_Cos,
##     family = binomial(link = "logit"), data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.4854     0.1131 -21.971  < 2e-16 ***
## Latitude       -1.8701     0.2152  -8.689  < 2e-16 ***
## Longitude       4.4632     0.3137  14.227  < 2e-16 ***
## TimeOfDay_Sin  -0.4059     0.1109  -3.659 0.000253 ***
## TimeOfDay_Cos   0.4112     0.1063   3.868 0.000110 ***
## Month_Cos      -0.1684     0.1019  -1.653 0.098334 .
```

```
## DayOfWeek_Sin   0.1932     0.1029   1.878 0.060390 .
## DayOfWeek_Cos   0.2582     0.1039   2.485 0.012938 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2232.86  on 2697  degrees of freedom
## Residual deviance:  809.44  on 2690  degrees of freedom
## AIC: 825.44
##
## Number of Fisher Scoring iterations: 7
##
##
## Metrics for MANHATTAN :
##           Accuracy Precision Recall  F1_Score
## Accuracy 0.9644444 0.9605911      1 0.9798995
```

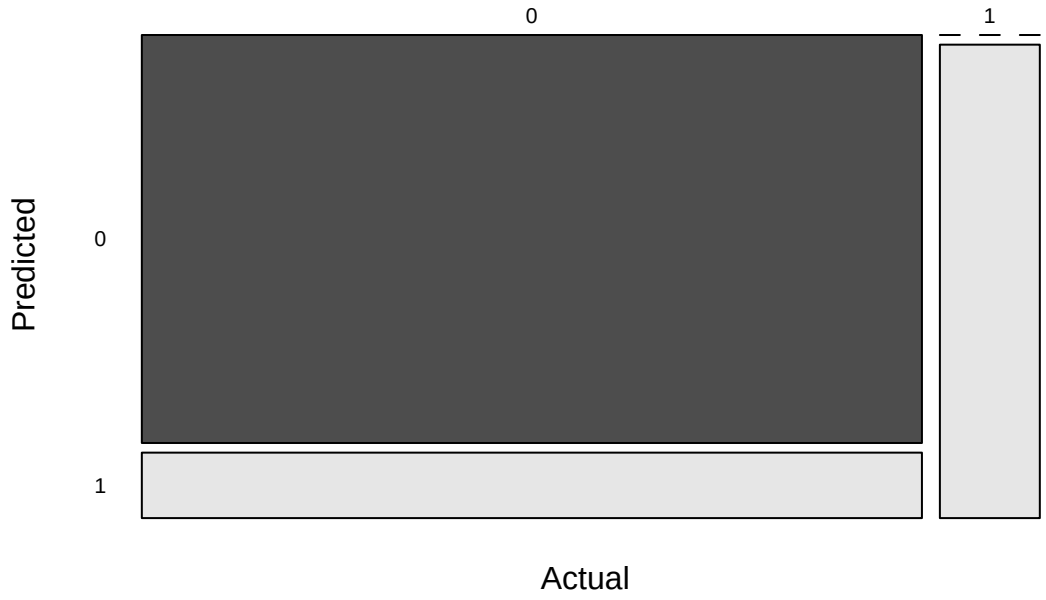


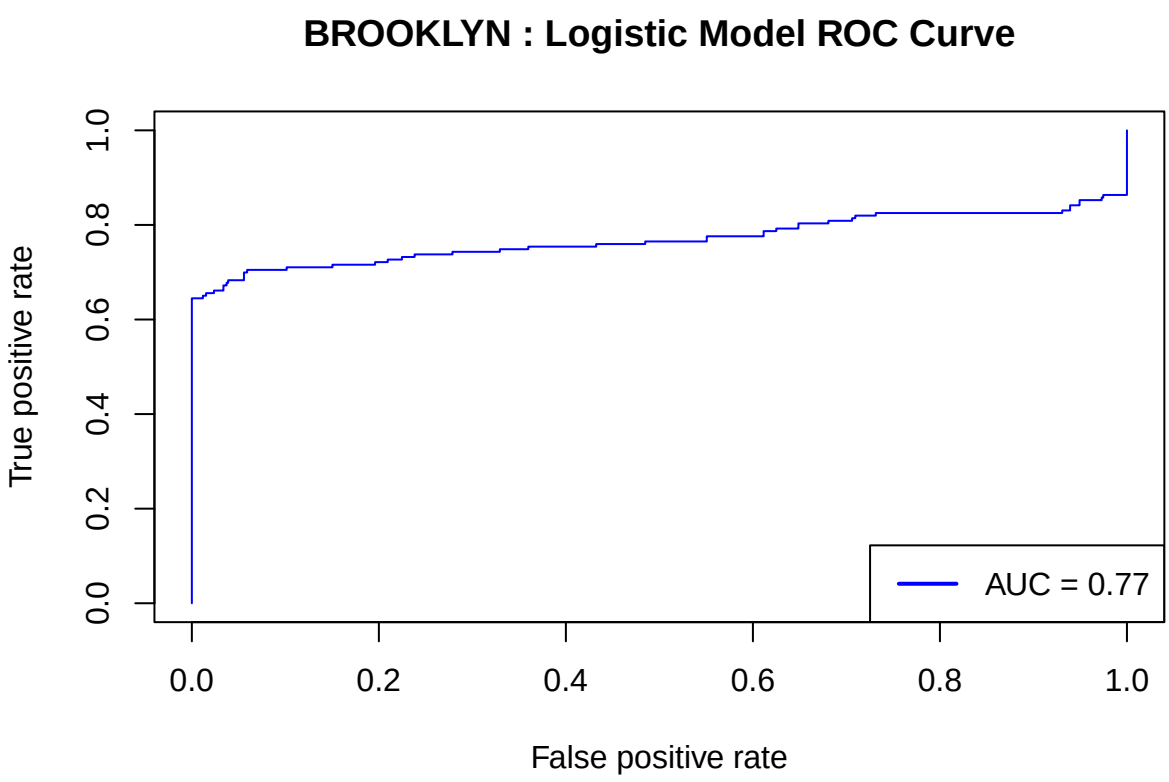



```
##
## Processing dataset for BROOKLYN :
##
## Full Model AIC: 2705.822
## Full Model BIC: 2766.21
##
## Calculating VIF for Multicollinearity:
##                          Latitude                         Longitude
##                          1.017018                          1.021725
##                     TimeOfDay_Sin                     TimeOfDay_Cos
##                          1.020095                          1.005795
##                         Month_Sin                         Month_Cos
##                          1.003828                          1.002721
##                     DayOfWeek_Sin                     DayOfWeek_Cos
##                          1.100535                          1.007230
## TimeOfDay_DayOfWeek_Interaction
##                          1.102039
##
## ANOVA Analysis:
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: SHOOTING_COUNT
##
## Terms added sequentially (first to last)
```

```
##
##
##                                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                                3098     3503.4
## Latitude                         1   456.83      3097     3046.5 < 2.2e-16 ***
## Longitude                        1   277.52      3096     2769.0 < 2.2e-16 ***
## TimeOfDay_Sin                    1    56.15      3095     2712.9 6.719e-14 ***
## TimeOfDay_Cos                    1     0.02      3094     2712.8  0.895251
## Month_Sin                        1     2.93      3093     2709.9  0.087124 .
## Month_Cos                        1     7.79      3092     2702.1  0.005253 **
## DayOfWeek_Sin                    1     5.48      3091     2696.6  0.019270 *
## DayOfWeek_Cos                    1     9.60      3090     2687.0  0.001945 **
## TimeOfDay_DayOfWeek_Interaction  1     1.22      3089     2685.8  0.268941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Performing Backward Feature Selection...
##
## Selected Model AIC: 2703.255
## Selected Model BIC: 2751.565
##
## Selected Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     Month_Sin + Month_Cos + DayOfWeek_Sin + DayOfWeek_Cos, family = binomial(link = "logit"),
##     data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.32816    0.05063 -26.234  < 2e-16 ***
## Latitude       0.79121    0.05792  13.660  < 2e-16 ***
## Longitude      0.80736    0.05576  14.479  < 2e-16 ***
## TimeOfDay_Sin -0.37602    0.05075  -7.409 1.27e-13 ***
## Month_Sin     -0.08103    0.04902  -1.653  0.09830 .
## Month_Cos     -0.13647    0.04902  -2.784  0.00536 **
## DayOfWeek_Sin  0.11214    0.04904   2.287  0.02222 *
## DayOfWeek_Cos  0.15077    0.04912   3.069  0.00214 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3503.4  on 3098  degrees of freedom
## Residual deviance: 2687.3  on 3091  degrees of freedom
## AIC: 2703.3
##
## Number of Fisher Scoring iterations: 5
##
##
## Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
```

```
##     Month_Sin + Month_Cos + DayOfWeek_Sin + DayOfWeek_Cos, family = binomial(link = "logit"),
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.32816    0.05063 -26.234  < 2e-16 ***
## Latitude       0.79121    0.05792  13.660  < 2e-16 ***
## Longitude      0.80736    0.05576  14.479  < 2e-16 ***
## TimeOfDay_Sin -0.37602    0.05075  -7.409 1.27e-13 ***
## Month_Sin     -0.08103    0.04902  -1.653  0.09830 .
## Month_Cos     -0.13647    0.04902  -2.784  0.00536 **
## DayOfWeek_Sin  0.11214    0.04904   2.287  0.02222 *
## DayOfWeek_Cos  0.15077    0.04912   3.069  0.00214 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3503.4  on 3098  degrees of freedom
## Residual deviance: 2687.3  on 3091  degrees of freedom
## AIC: 2703.3
##
## Number of Fisher Scoring iterations: 5
##
##
## Metrics for BROOKLYN :
##           Accuracy Precision Recall  F1_Score
## Accuracy 0.8774194 0.8617176      1 0.9257232
```

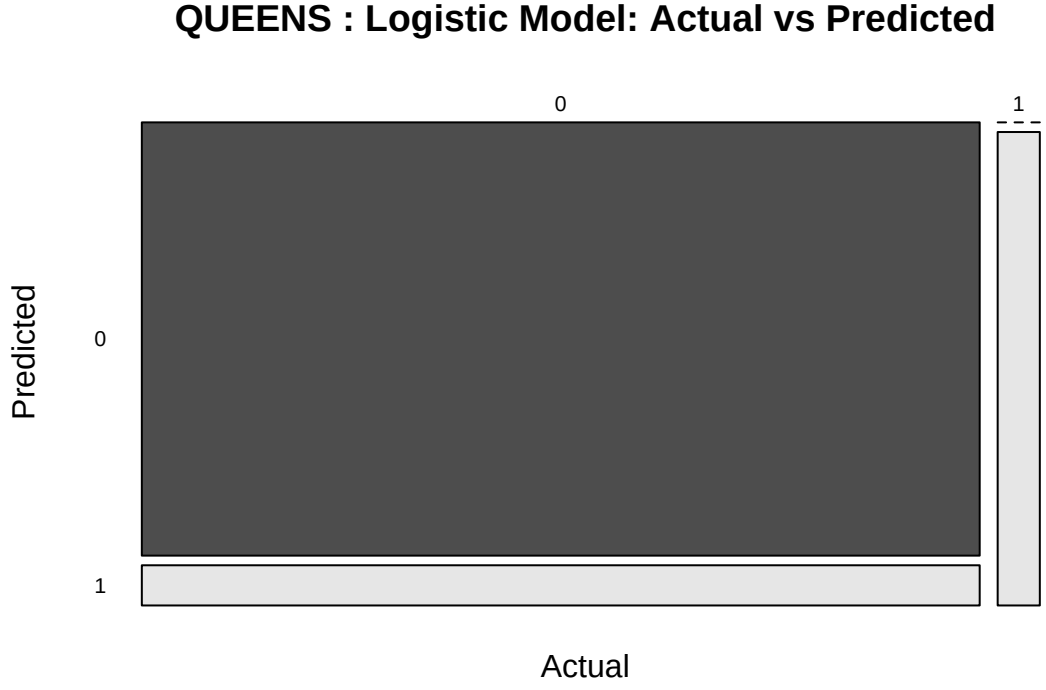

BROOKLYN : Logistic Model: Actual vs Predicted
0
1
Predicted
0
1
Actual

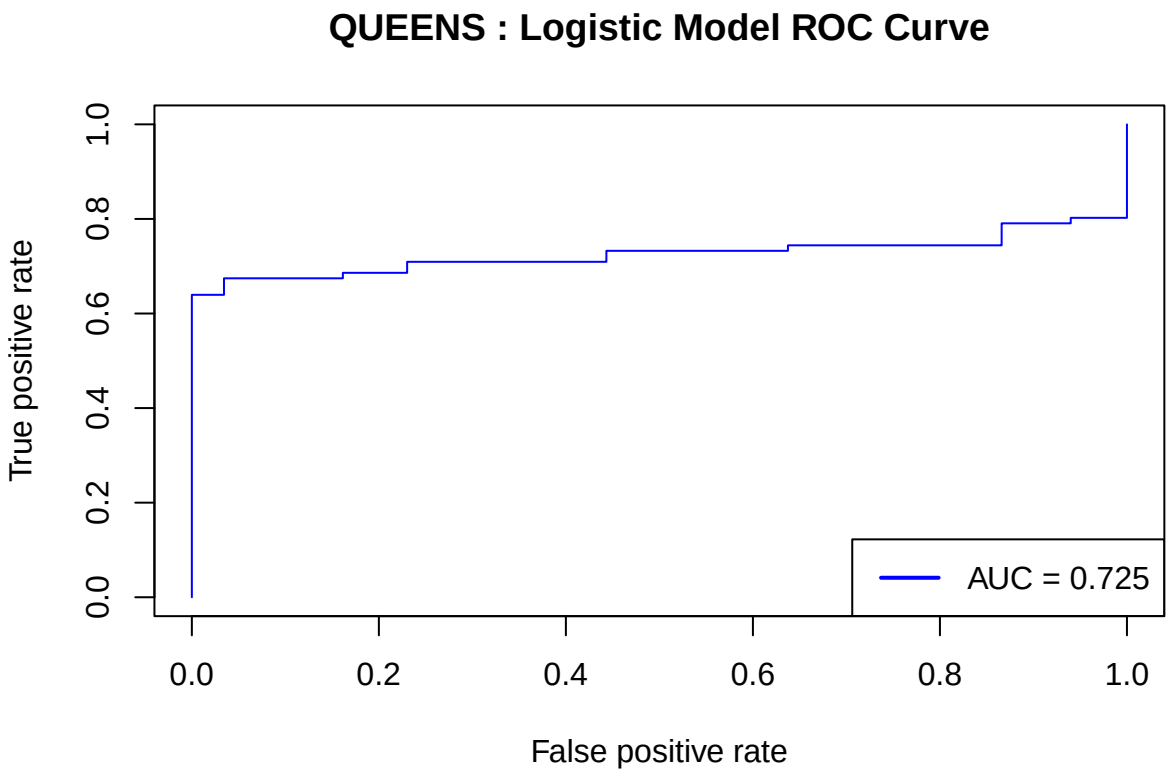



```
##
## Processing dataset for QUEENS :
##
## Full Model AIC: 1440.66
## Full Model BIC: 1499.562
##
## Calculating VIF for Multicollinearity:
##                          Latitude                         Longitude
##                          2.646750                          2.653799
##                     TimeOfDay_Sin                     TimeOfDay_Cos
##                          1.041389                          1.011972
##                         Month_Sin                         Month_Cos
##                          1.009447                          1.005607
##                     DayOfWeek_Sin                     DayOfWeek_Cos
##                          1.151907                          1.005024
## TimeOfDay_DayOfWeek_Interaction
##                          1.145027
##
## ANOVA Analysis:
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: SHOOTING_COUNT
##
## Terms added sequentially (first to last)
```

```
##
##
##                                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                             2670     2070.8
## Latitude                         1   445.03      2669     1625.8 < 2.2e-16 ***
## Longitude                        1   168.53      2668     1457.2 < 2.2e-16 ***
## TimeOfDay_Sin                    1    22.77      2667     1434.5 1.828e-06 ***
## TimeOfDay_Cos                    1     0.76      2666     1433.7  0.382129
## Month_Sin                        1     0.13      2665     1433.6  0.713937
## Month_Cos                        1     0.26      2664     1433.3  0.612735
## DayOfWeek_Sin                    1     0.78      2663     1432.5  0.378645
## DayOfWeek_Cos                    1     9.54      2662     1423.0  0.002005 **
## TimeOfDay_DayOfWeek_Interaction  1     2.33      2661     1420.7  0.126717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Performing Backward Feature Selection...
##
## Selected Model AIC: 1434.776
## Selected Model BIC: 1464.227
##
## Selected Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     DayOfWeek_Cos, family = binomial(link = "logit"), data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.16741    0.07616 -28.459  < 2e-16 ***
## Latitude      -2.42294    0.15708 -15.425  < 2e-16 ***
## Longitude     -1.11735    0.09611 -11.626  < 2e-16 ***
## TimeOfDay_Sin -0.36004    0.07507  -4.796 1.62e-06 ***
## DayOfWeek_Cos  0.22583    0.07312   3.089  0.00201 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2070.8  on 2670  degrees of freedom
## Residual deviance: 1424.8  on 2666  degrees of freedom
## AIC: 1434.8
##
## Number of Fisher Scoring iterations: 6
##
##
## Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     DayOfWeek_Cos, family = binomial(link = "logit"), data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)   -2.16741    0.07616 -28.459  < 2e-16 ***
## Latitude      -2.42294    0.15708 -15.425  < 2e-16 ***
## Longitude     -1.11735    0.09611 -11.626  < 2e-16 ***
## TimeOfDay_Sin -0.36004    0.07507  -4.796 1.62e-06 ***
## DayOfWeek_Cos  0.22583    0.07312   3.089  0.00201 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2070.8  on 2670  degrees of freedom
## Residual deviance: 1424.8  on 2666  degrees of freedom
## AIC: 1434.8
##
## Number of Fisher Scoring iterations: 6
##
##
## Metrics for QUEENS :
##           Accuracy Precision Recall F1_Score
## Accuracy 0.9191617 0.9150943      1 0.955665
```

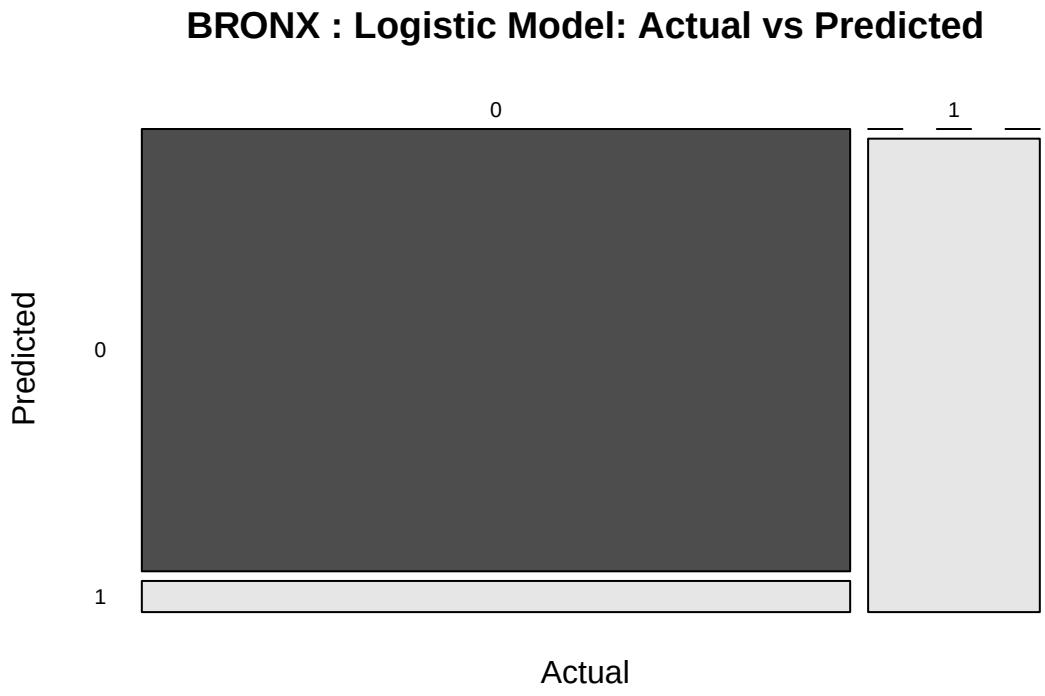


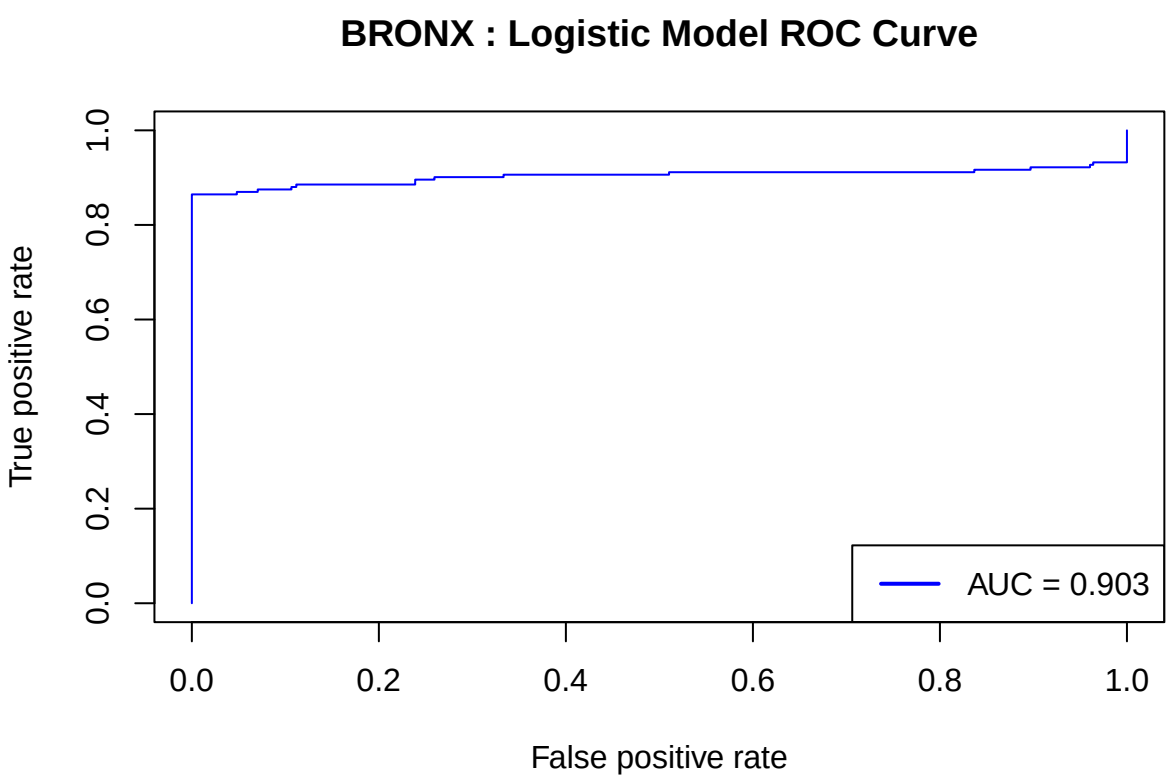



```
##
## Processing dataset for BRONX :
##
## Full Model AIC: 1751.758
## Full Model BIC: 1812.134
##
## Calculating VIF for Multicollinearity:
##                          Latitude                         Longitude
##                          1.585180                          1.578669
##                     TimeOfDay_Sin                     TimeOfDay_Cos
##                          1.022107                          1.004260
##                         Month_Sin                         Month_Cos
##                          1.005429                          1.007389
##                     DayOfWeek_Sin                     DayOfWeek_Cos
##                          1.144837                          1.001427
## TimeOfDay_DayOfWeek_Interaction
##                          1.157136
##
## ANOVA Analysis:
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: SHOOTING_COUNT
##
## Terms added sequentially (first to last)
```

```
##
##
##                                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                                3094     3470.4
## Latitude                         1     0.84      3093     3469.5  0.358833
## Longitude                        1  1691.18      3092     1778.4 < 2.2e-16 ***
## TimeOfDay_Sin                    1    34.03      3091     1744.3 5.434e-09 ***
## TimeOfDay_Cos                    1     0.40      3090     1743.9  0.524682
## Month_Sin                        1     0.67      3089     1743.3  0.414219
## Month_Cos                        1     7.66      3088     1735.6  0.005657 **
## DayOfWeek_Sin                    1     1.05      3087     1734.5  0.304724
## DayOfWeek_Cos                    1     0.24      3086     1734.3  0.623660
## TimeOfDay_DayOfWeek_Interaction  1     2.55      3085     1731.8  0.110330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Performing Backward Feature Selection...
##
## Selected Model AIC: 1747.205
## Selected Model BIC: 1789.468
##
## Selected Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     Month_Cos + DayOfWeek_Sin + TimeOfDay_DayOfWeek_Interaction,
##     family = binomial(link = "logit"), data = train_data)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -1.33772    0.06772 -19.754  < 2e-16 ***
## Latitude                         1.03718    0.06867  15.103  < 2e-16 ***
## Longitude                       -2.65875    0.11292 -23.544  < 2e-16 ***
## TimeOfDay_Sin                   -0.38102    0.06666  -5.716 1.09e-08 ***
## Month_Cos                       -0.18195    0.06553  -2.777  0.00549 **
## DayOfWeek_Sin                    0.10339    0.06865   1.506  0.13206
## TimeOfDay_DayOfWeek_Interaction  0.10556    0.06731   1.568  0.11681
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3470.4  on 3094  degrees of freedom
## Residual deviance: 1733.2  on 3088  degrees of freedom
## AIC: 1747.2
##
## Number of Fisher Scoring iterations: 6
##
##
## Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude + TimeOfDay_Sin +
##     Month_Cos + DayOfWeek_Sin + TimeOfDay_DayOfWeek_Interaction,
```

```
##     family = binomial(link = "logit"), data = train_data)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -1.33772    0.06772 -19.754  < 2e-16 ***
## Latitude                         1.03718    0.06867  15.103  < 2e-16 ***
## Longitude                       -2.65875    0.11292 -23.544  < 2e-16 ***
## TimeOfDay_Sin                   -0.38102    0.06666  -5.716 1.09e-08 ***
## Month_Cos                       -0.18195    0.06553  -2.777  0.00549 **
## DayOfWeek_Sin                    0.10339    0.06865   1.506  0.13206
## TimeOfDay_DayOfWeek_Interaction  0.10556    0.06731   1.568  0.11681
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3470.4  on 3094  degrees of freedom
## Residual deviance: 1733.2  on 3088  degrees of freedom
## AIC: 1747.2
##
## Number of Fisher Scoring iterations: 6
##
##
## Metrics for BRONX :
##           Accuracy Precision Recall  F1_Score
## Accuracy 0.9470284 0.9341894      1 0.9659751
```

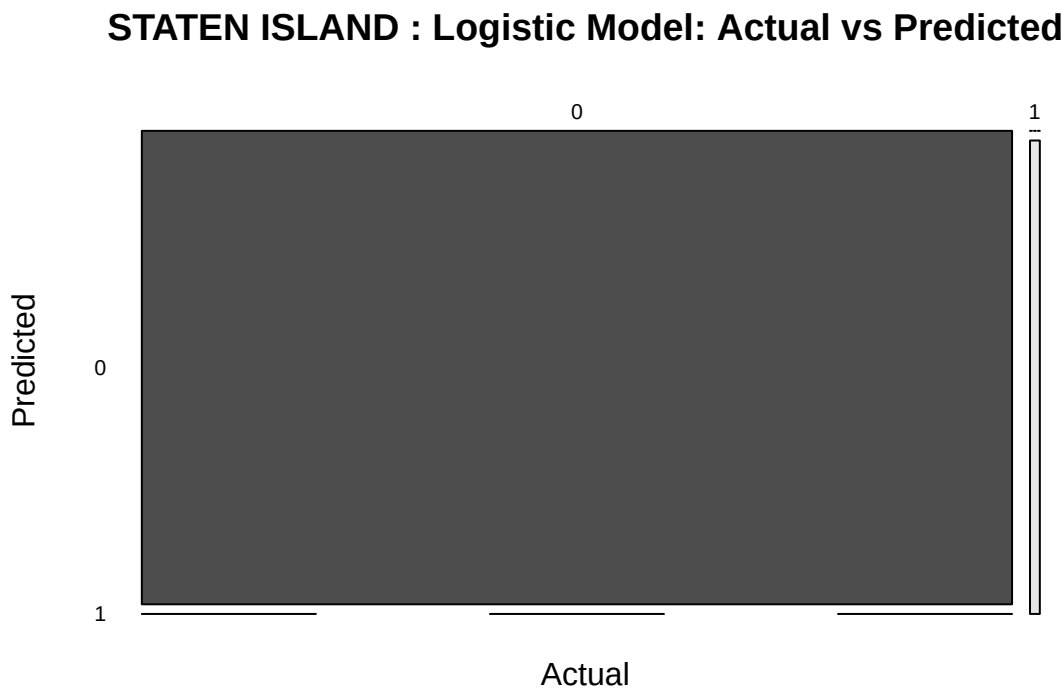


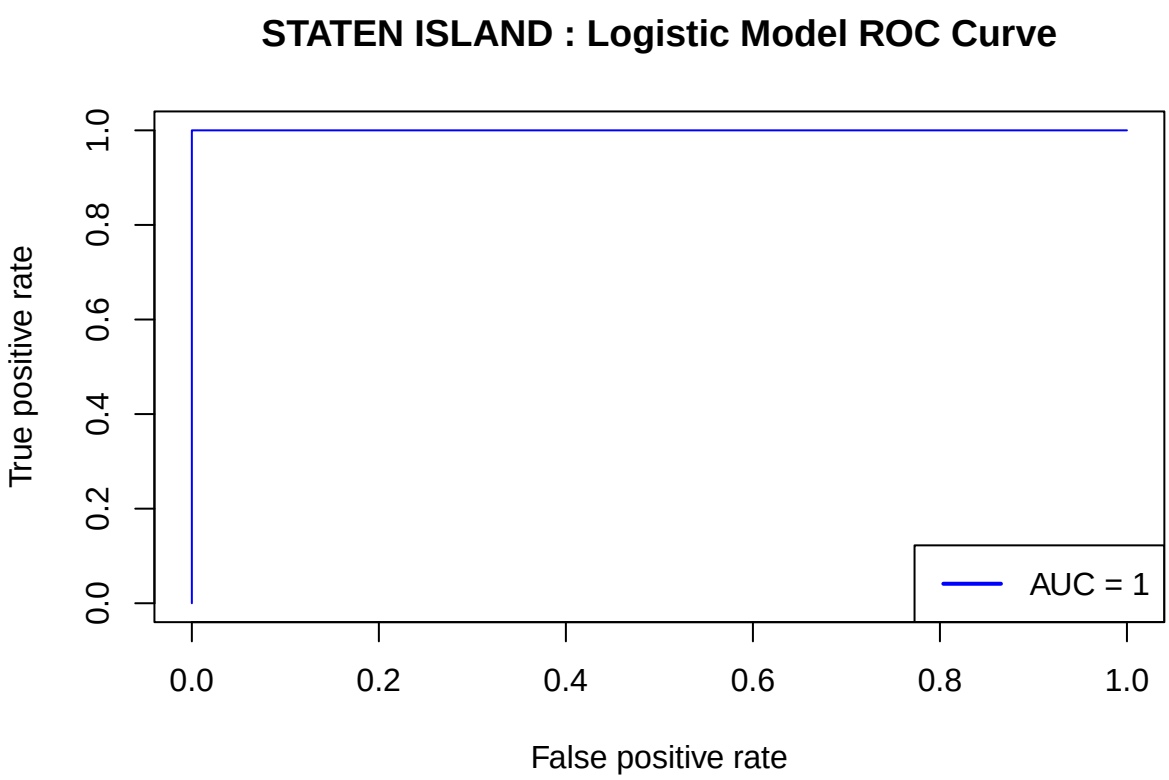



```
##
## Processing dataset for STATEN ISLAND :
##
## Full Model AIC: 196.0314
## Full Model BIC: 252.7775
##
## Calculating VIF for Multicollinearity:
##                         Latitude                        Longitude
##                         1.542365                         1.494872
##                    TimeOfDay_Sin                    TimeOfDay_Cos
##                         1.029977                         1.130792
##                        Month_Sin                        Month_Cos
##                         1.132971                         1.037923
##                    DayOfWeek_Sin                    DayOfWeek_Cos
##                         1.097410                         1.031651
## TimeOfDay_DayOfWeek_Interaction
##                         1.123905
##
## ANOVA Analysis:
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: SHOOTING_COUNT
##
## Terms added sequentially (first to last)
```

```
##
##
##                                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                                 2152     533.67
## Latitude                         1   335.79      2151     197.88 < 2.2e-16 ***
## Longitude                        1    18.85      2150     179.02 1.411e-05 ***
## TimeOfDay_Sin                    1     0.91      2149     178.12    0.3408
## TimeOfDay_Cos                    1     1.18      2148     176.94    0.2782
## Month_Sin                        1     0.18      2147     176.76    0.6699
## Month_Cos                        1     0.00      2146     176.75    0.9463
## DayOfWeek_Sin                    1     0.41      2145     176.34    0.5225
## DayOfWeek_Cos                    1     0.10      2144     176.24    0.7526
## TimeOfDay_DayOfWeek_Interaction  1     0.21      2143     176.03    0.6440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Performing Backward Feature Selection...
##
## Selected Model AIC: 185.0227
## Selected Model BIC: 202.0465
##
## Selected Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude, family = binomial(link = "logit"),
##     data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.9295     0.2737 -18.010  < 2e-16 ***
## Latitude      0.8811     0.1261   6.985 2.85e-12 ***
## Longitude     0.5086     0.1228   4.142 3.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 533.67  on 2152  degrees of freedom
## Residual deviance: 179.02  on 2150  degrees of freedom
## AIC: 185.02
##
## Number of Fisher Scoring iterations: 8
##
##
## Model Summary:
##
## Call:
## glm(formula = SHOOTING_COUNT ~ Latitude + Longitude, family = binomial(link = "logit"),
##     data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.9295     0.2737 -18.010  < 2e-16 ***
## Latitude      0.8811     0.1261   6.985 2.85e-12 ***
```

```
## Longitude     0.5086     0.1228   4.142 3.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 533.67  on 2152  degrees of freedom
## Residual deviance: 179.02  on 2150  degrees of freedom
## AIC: 185.02
##
## Number of Fisher Scoring iterations: 8
##
##
## Metrics for STATEN ISLAND :
##          Accuracy Precision Recall F1_Score
## Accuracy        1         1      1        1
```





Bias Analysis

1. Reporting Bias This dataset only includes shootings recorded by the NYPD. Therefore, unreported incidents or errors in recording can lead to an underestimation or misrepresentation in the shooting data. This type of reporting bias may skew our results, especially if certain areas or types of incidents are underreported.

2. Geographic Bias The data is collected solely from New York City and broken down by borough, so any analysis is specific to this location. Trends identified here may not be generalizable to other cities or regions. Additionally, some boroughs may have more densely populated areas, which can lead to higher numbers of reported shootings simply due to higher population density, rather than increased per capita shooting rates.

3. Temporal Bias The dataset only includes data from 2022 and 2023. This is a relatively short time frame and may not capture longer-term trends, seasonal patterns, or year-to-year variations. The limited period could lead to incorrect assumptions about trends if we generalize beyond this timeframe.

4. Analytical Bias In the modeling section, we used features such as Year, Month, DayOfWeek, TimeOfDay, and BORO. If these features are not representative of the true underlying causes of shooting incidents, the model might overfit to patterns that do not generalize well. Furthermore, the model assumes a Binomial distribution, which might not fully capture the variance in the data. Alternatively, aggregating this data would have allowed for a poisson based model where the data can range from 0 to infinity (theoretically).

5. Analysis Interpretation Bias The way we interpret the findings from our visualizations and model predictions can introduce bias. For example, higher predicted shooting counts in specific boroughs

might be attributed to socio-economic factors, which we haven't analyzed here. Such assumptions could lead to incorrect or overly simplified conclusions.

6. Unaccounted for Variable Bias Policing policies or socio-economic factors might influence shooting counts but are not part of this dataset.

Model Selection

The Logistic regression model is well-suited for classification tasks and is easily interpretable. In the future we may choose to model shooting counts as a poisson random variable by summing shooting counts over time periods which can provide a different insight.

Interpretation of Results

The results from our analysis suggest several notable trends:

Temporal Trends: Our visualizations show differences in shooting counts by month, time of day, and year.

Spatial Trends: Shootings stratified by boroughs revealed that certain boroughs experience notably higher shooting counts, with the Bronx and Brooklyn having more incidents than others. Additionally, The coordinates consistently showed statistical significance in our ANOVA tests and in our model summary.

Model Findings: Predictor significance varied by borough, but notably longitude and latitude remained consistent statistically significant predictors. In boroughs with more sparsely distributed shootings, through backward feature selection, the model found longitude and latitude to be the only statistically significant predictors. In boroughs with more shooting incidents, features such as time of day, day of week, and month of year also became significant.

These findings can support decision-makers in identifying higher-risk periods and areas for shootings. However, caution should be taken due to potential biases and the model's limitations in capturing complex causative factors.