NYPD Shooting Incident Study

H. Shteyn

2023-07-19

This analysis aims to understand the patterns of NYPD shootings using police data and make predictions based on relevant features

Importing Data

Technicalities: The data was downloaded from https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD .

Libraries that are used in the project: "tidyverse", "lubridate", "xgboost".

It can be installed in the following way:

install.packages("tidyverse")

install.packages("lubridate")

install.packages("xgboost")

In the following step, we import necessary libraries and read the data.

```
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(lubridate))
suppressPackageStartupMessages(library(xgboost))


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

```
summary(df)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:27312       Length:27312      Length:27312
##  1st Qu.: 63860880   Class :character   Class1:hms        Class :character
##  Median : 90372218   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :120860536                      Mode  :numeric
##  3rd Qu.:188810230
##  Max.   :261190187
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character
##                     Mean   : 65.64   Mean   :0.3269
```

```
##                       3rd Qu.: 81.00   3rd Qu.:0.0000
##                       Max.   :123.00   Max.   :2.0000
##                                        NA's   :2
##  LOCATION_DESC       STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312        Mode :logical           Length:27312
##  Class :character    FALSE:22046             Class :character
##  Mode  :character    TRUE :5266              Mode  :character
##
##
##
##
##    PERP_SEX           PERP_RACE          VIC_AGE_GROUP        VIC_SEX
##  Length:27312        Length:27312       Length:27312       Length:27312
##  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:27312        Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character    1st Qu.:1000029   1st Qu.:182834   1st Qu.:40.67
##  Mode  :character    Median :1007731   Median :194487   Median :40.70
##                      Mean   :1009449   Mean   :208127   Mean   :40.74
##                      3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82
##                      Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                         NA's   :10
##    Longitude        Lon_Lat
##  Min.   :-74.25   Length:27312
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :10
```

Data Post-Processing

Processing data after the importing

Checking how many values in the variable, if less than 10 - make it factor Setting the threshold for unique values to 10. Afterwards the relevant columns were chosen and data was grouped by the relevant for the prediction model variables. The subsequent code illustrates these steps.

```
# handling missing values for Random forest model
# loop over the numerical columns
numeric_columns <- sapply(df, is.numeric)  # Find numeric columns
df[numeric_columns] <- lapply(df[numeric_columns], function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x


# loop over the categorical columns
categorical_columns <- sapply(df, is.factor)  # Find categorical columns
df[categorical_columns] <- lapply(df[categorical_columns], function(x) ifelse(is.na(x), names(which.max
```

```r
N = 10

# Loop through each column in the dataframe
for (colname in names(df)) {
  # Check if the column is character type
  if (is.character(df[[colname]])) {
    # Count the unique values
    num_unique_values <- length(unique(df[[colname]]))

    # If the count is less than N, convert to factor
    if (num_unique_values < N) {
      df[[colname]] <- as.factor(df[[colname]])
    }
  }
}

# convert character to date/time format
df$OCCUR_DATE = mdy(df$OCCUR_DATE)
df$OCCUR_TIME = hms(df$OCCUR_TIME)


# Print columns types after changing variables format
sapply(df, class)
```

```
##            INCIDENT_KEY              OCCUR_DATE              OCCUR_TIME
##               "numeric"                  "Date"                "Period"
##                    BORO       LOC_OF_OCCUR_DESC                PRECINCT
##                "factor"                "factor"               "numeric"
##       JURISDICTION_CODE      LOC_CLASSFCTN_DESC           LOCATION_DESC
##               "numeric"             "character"             "character"
## STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP                PERP_SEX
##               "logical"             "character"                "factor"
##               PERP_RACE           VIC_AGE_GROUP                 VIC_SEX
##                "factor"                "factor"                "factor"
##                VIC_RACE              X_COORD_CD              Y_COORD_CD
##                "factor"               "numeric"               "numeric"
##                Latitude               Longitude                 Lon_Lat
##               "numeric"               "numeric"             "character"
```

```r
# show all the column names
names(df)
```

```
##  [1] "INCIDENT_KEY"            "OCCUR_DATE"
##  [3] "OCCUR_TIME"              "BORO"
##  [5] "LOC_OF_OCCUR_DESC"       "PRECINCT"
##  [7] "JURISDICTION_CODE"       "LOC_CLASSFCTN_DESC"
##  [9] "LOCATION_DESC"           "STATISTICAL_MURDER_FLAG"
## [11] "PERP_AGE_GROUP"          "PERP_SEX"
## [13] "PERP_RACE"               "VIC_AGE_GROUP"
## [15] "VIC_SEX"                 "VIC_RACE"
## [17] "X_COORD_CD"              "Y_COORD_CD"
## [19] "Latitude"                "Longitude"
## [21] "Lon_Lat"
```

```r
# select only relevant columns
df_columns <- df %>% select(c("INCIDENT_KEY", "OCCUR_DATE", "BORO", "PRECINCT", "JURISDICTION_CODE", "S

# group data by multiple variables for the model
df_group_all = df_columns %>%
  group_by(BORO, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP , VIC_SEX, VIC_RACE) %>%
  summarize(INCIDENT_count = n(), .groups = 'drop') %>%
  select(INCIDENT_count, BORO, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP , VIC_SEX, VIC_RACE)
```

```r
# Dataframe summary after processing
summary(df_group_all)
```

```
##  INCIDENT_count              BORO     PERP_AGE_GROUP       PERP_SEX
##  Min.   :   1.00   BRONX        :629   Length:2391        (null):  91
##  1st Qu.:   1.00   BROOKLYN     :601   Class :character   F     : 230
##  Median :   2.00   MANHATTAN    :455   Mode  :character   M     :1723
##  Mean   :  11.42   QUEENS       :501                      U     : 157
##  3rd Qu.:   5.00   STATEN ISLAND:205                      NA's  : 190
##  Max.   :1656.00
##
##          PERP_RACE   VIC_AGE_GROUP VIC_SEX
##  BLACK         :787   <18    :356   F: 686
##  WHITE HISPANIC:458   1022   :  1   M:1696
##  BLACK HISPANIC:324   18-24  :657   U:   9
##  UNKNOWN       :271   25-44  :776
##  WHITE         :176   45-64  :434
##  (Other)       :185   65+    :120
##  NA's          :190   UNKNOWN: 47
##                            VIC_RACE
##  AMERICAN INDIAN/ALASKAN NATIVE: 10
##  ASIAN / PACIFIC ISLANDER      :175
##  BLACK                         :803
##  BLACK HISPANIC                :439
##  UNKNOWN                       : 52
##  WHITE                         :323
##  WHITE HISPANIC                :589
```

Visualizing Data

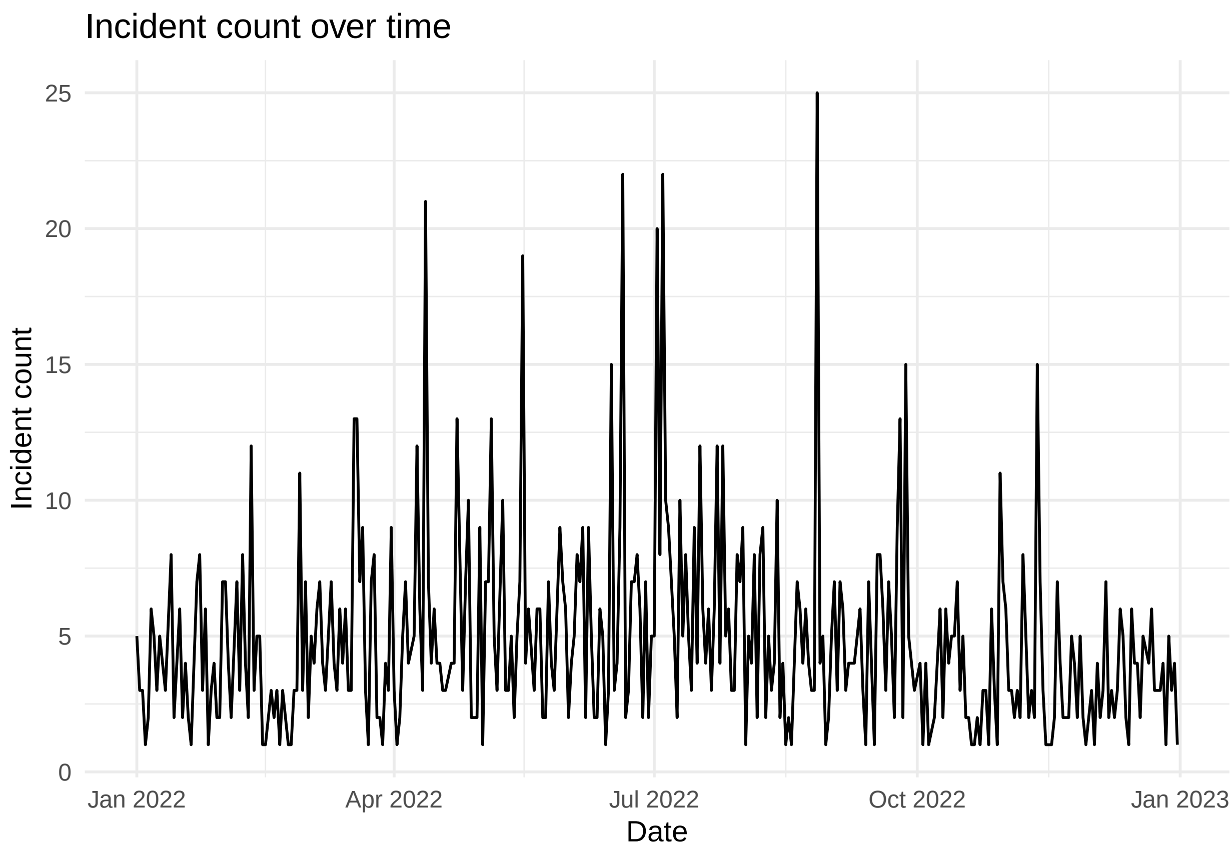
2 types of visualizations were created: Incident count over time and Incidents per BORO To facilitate the necessary analysis and visualizations, the data was grouped by relevant columns.

```r
# Filter the data starting from 2022
df_group_date_2023 <- df_columns %>%
  filter(OCCUR_DATE >= as.Date("2022-01-01"))

# calculate the count of shootings per date
df_group_date = df_group_date_2023 %>%
  group_by(OCCUR_DATE) %>%
  summarize(INCIDENT_count = n(), .groups = 'drop') %>%
  select(OCCUR_DATE,INCIDENT_count)
```

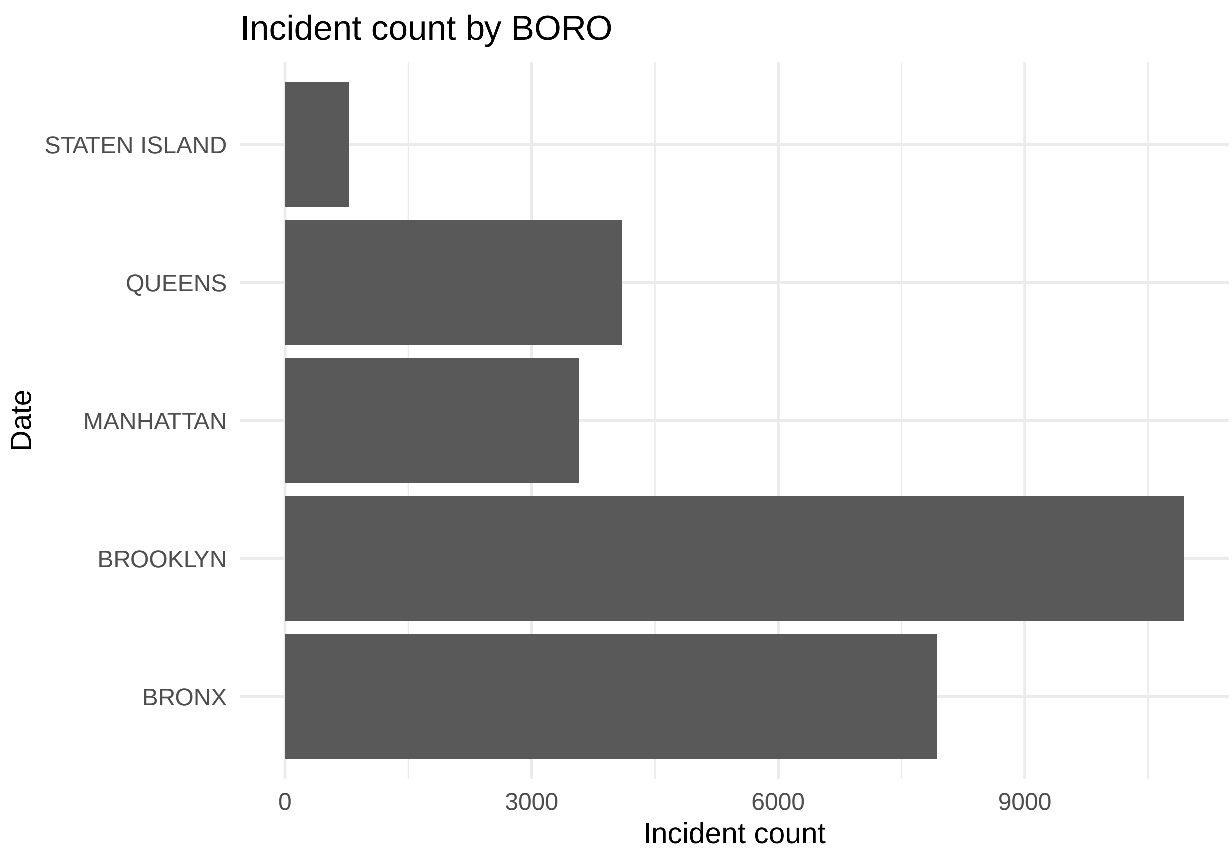
```r
# calculate the count of shootings per borough
df_group_BORO = df_columns %>%
  group_by(BORO) %>%
  summarize(INCIDENT_count = n(), .groups = 'drop') %>%
  select(BORO, INCIDENT_count)
```

```r
# Incident count over time
ggplot(df_group_date, aes(x = OCCUR_DATE, y = INCIDENT_count)) +
  geom_line() +
  labs(x = "Date", y = "Incident count", title = "Incident count over time") +
  theme_minimal()
```




```r
# incidents per BORO

ggplot(df_group_BORO, aes(x = BORO, y = INCIDENT_count)) +
  geom_bar(stat = "identity") +
  labs(x = "Date", y = "Incident count", title = "Incident count by BORO") +
  theme_minimal() +
  coord_flip()
```



Modeling the data

Building a Linear Model

We predict the incident count using the following features: IC_AGE_GROUP, BORO, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VIC_RACE

```
# make linear model
mod = lm(INCIDENT_count ~ VIC_AGE_GROUP + BORO + PERP_AGE_GROUP + PERP_SEX + PERP_RACE + VIC_SEX + VIC_I

# Model summary
summary(mod)
```

```
##
## Call:
## lm(formula = INCIDENT_count ~ VIC_AGE_GROUP + BORO + PERP_AGE_GROUP +
##     PERP_SEX + PERP_RACE + VIC_SEX + VIC_RACE, data = df_group_all)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -47.29 -10.10  -3.40   3.79 676.53
##
## Coefficients: (2 not defined because of singularities)
##                                           Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                            -15.8067    11.7175  -1.349 0.177486
## VIC_AGE_GROUP1022                      -26.9366    31.5577  -0.854 0.393439
## VIC_AGE_GROUP18-24                       6.5956     2.2032   2.994 0.002788
## VIC_AGE_GROUP25-44                       8.1653     2.1726   3.758 0.000176
## VIC_AGE_GROUP45-64                      -1.6900     2.4318  -0.695 0.487143
## VIC_AGE_GROUP65+                        -6.9418     3.7175  -1.867 0.061991
## VIC_AGE_GROUPUNKNOWN                    -9.2268     5.5680  -1.657 0.097647
## BOROBROOKLYN                             2.2609     1.8950   1.193 0.232976
## BOROMANHATTAN                           -4.9672     2.0312  -2.445 0.014549
## BOROQUEENS                              -4.0503     2.0013  -2.024 0.043108
## BOROSTATEN ISLAND                      -11.2717     2.7527  -4.095 4.38e-05
## PERP_AGE_GROUP<18                        0.8784     5.4771   0.160 0.872605
## PERP_AGE_GROUP1020                     -30.0191    31.9245  -0.940 0.347161
## PERP_AGE_GROUP18-24                     10.0500     5.2977   1.897 0.057954
## PERP_AGE_GROUP224                       -5.8149    31.8654  -0.182 0.855221
## PERP_AGE_GROUP25-44                      9.1882     5.3195   1.727 0.084265
## PERP_AGE_GROUP45-64                      0.3705     5.5850   0.066 0.947116
## PERP_AGE_GROUP65+                        1.8945     7.2102   0.263 0.792767
## PERP_AGE_GROUP940                       -9.6455    31.8648  -0.303 0.762148
## PERP_AGE_GROUPUNKNOWN                    3.0051     5.1606   0.582 0.560418
## PERP_SEXF                              -21.9216     4.2885  -5.112 3.47e-07
## PERP_SEXM                               -5.8178     3.6677  -1.586 0.112839
## PERP_SEXU                                    NA         NA      NA       NA
## PERP_RACEAMERICAN INDIAN/ALASKAN NATIVE -13.6936    22.3396  -0.613 0.539959
## PERP_RACEASIAN / PACIFIC ISLANDER       -7.8357     3.6880  -2.125 0.033729
## PERP_RACEBLACK                          10.9400     1.8866   5.799 7.66e-09
## PERP_RACEBLACK HISPANIC                 -2.7749     2.3008  -1.206 0.227935
## PERP_RACEUNKNOWN                        -3.5432     3.0975  -1.144 0.252790
## PERP_RACEWHITE                          -1.5700     2.9236  -0.537 0.591300
## PERP_RACEWHITE HISPANIC                      NA         NA      NA       NA
## VIC_SEXM                                10.9006     1.5318   7.116 1.51e-12
## VIC_SEXU                                 6.9118    13.8519   0.499 0.617846
## VIC_RACEASIAN / PACIFIC ISLANDER         3.0584    11.4797   0.266 0.789939
## VIC_RACEBLACK                           22.6377    11.2351   2.015 0.044039
## VIC_RACEBLACK HISPANIC                   8.3387    11.2844   0.739 0.460012
## VIC_RACEUNKNOWN                         -0.1388    12.3977  -0.011 0.991065
## VIC_RACEWHITE                            7.3491    11.3679   0.646 0.518041
## VIC_RACEWHITE HISPANIC                  10.9431    11.2570   0.972 0.331104
##
## (Intercept)
## VIC_AGE_GROUP1022
## VIC_AGE_GROUP18-24                     **
## VIC_AGE_GROUP25-44                     ***
## VIC_AGE_GROUP45-64
## VIC_AGE_GROUP65+                       .
## VIC_AGE_GROUPUNKNOWN                   .
## BOROBROOKLYN
## BOROMANHATTAN                          *
## BOROQUEENS                             *
## BOROSTATEN ISLAND                      ***
## PERP_AGE_GROUP<18
## PERP_AGE_GROUP1020
## PERP_AGE_GROUP18-24                    .
## PERP_AGE_GROUP224
```

```
## PERP_AGE_GROUP25-44                     .
## PERP_AGE_GROUP45-64
## PERP_AGE_GROUP65+
## PERP_AGE_GROUP940
## PERP_AGE_GROUPUNKNOWN
## PERP_SEXF                               ***
## PERP_SEXM
## PERP_SEXU
## PERP_RACEAMERICAN INDIAN/ALASKAN NATIVE
## PERP_RACEASIAN / PACIFIC ISLANDER       *
## PERP_RACEBLACK                          ***
## PERP_RACEBLACK HISPANIC
## PERP_RACEUNKNOWN
## PERP_RACEWHITE
## PERP_RACEWHITE HISPANIC
## VIC_SEXM                                ***
## VIC_SEXU
## VIC_RACEASIAN / PACIFIC ISLANDER
## VIC_RACEBLACK                           *
## VIC_RACEBLACK HISPANIC
## VIC_RACEUNKNOWN
## VIC_RACEWHITE
## VIC_RACEWHITE HISPANIC
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.42 on 2145 degrees of freedom
##   (210 observations deleted due to missingness)
## Multiple R-squared:  0.1246, Adjusted R-squared:  0.1104
## F-statistic: 8.726 on 35 and 2145 DF,  p-value: < 2.2e-16
```

Linear model results and thoughts:

As it is clearly seen from the summary, the R-squared value is 0.1246, suggesting that about 12.46% of the variability in INCIDENT_count can be explained by the predictors in the model. The "Adjusted R-squared" is 0.1104, meaning that about 11.04% of the variance in INCIDENT_count can be explained by our predictors, adjusting for the number of predictors. So from the summary output, it seems that the model does not predict INCIDENT_count very well since it only explains around 11% of its variability.

In order to improve prediction model it was decided to try xgboost algorithm for this matter as it should be more precise

Pre-processing Data (processing data for the xgboost prediction)

```r
# Split data into train and test
set.seed(42)
train_index <- sample(1:nrow(df_group_all), 0.7*nrow(df_group_all))
train_data <- df_group_all[train_index, ]
test_data <- df_group_all[-train_index, ]

# Omit rows with NA values for both train and test datasets
train_data <- na.omit(train_data)
test_data <- na.omit(test_data)
```

```r
# For numeric columns
for (colname in names(train_data)) {
  if (is.numeric(train_data[[colname]])) {
    train_data[[colname]][is.na(train_data[[colname]])] <- mean(train_data[[colname]], na.rm = TRUE)
  }
}

# For categorical columns
for (colname in names(train_data)) {
  if (is.factor(train_data[[colname]])) {
    mode_value <- levels(train_data[[colname]])[which.max(table(train_data[[colname]]))]
    train_data[[colname]][is.na(train_data[[colname]])] <- mode_value
  }
}

# Convert data to matrix format as xgboost prefers matrix or DMatrix formats for input data.
train_matrix <- as.matrix(train_data[, -which(names(train_data) == "INCIDENT_count")])  # excluding tar
test_matrix <- as.matrix(test_data[, -which(names(test_data) == "INCIDENT_count")])

# Convert factors to numeric using one-hot encoding
train_data_one_hot <- model.matrix(INCIDENT_count ~ . - 1, data=train_data)
train_labels <- train_data$INCIDENT_count

test_data_one_hot <- model.matrix(INCIDENT_count ~ . - 1, data=test_data)
test_labels <- test_data$INCIDENT_count

# Combine datasets
all_data <- rbind(train_data, test_data)

# Apply one-hot encoding
all_data_one_hot <- model.matrix(INCIDENT_count ~ . - 1, data=all_data)

# Split them back
train_data_one_hot <- all_data_one_hot[1:nrow(train_data), ]
test_data_one_hot <- all_data_one_hot[(nrow(train_data) + 1):nrow(all_data_one_hot), ]
```

XGBOOST prediction and evaluation

```r
# Set parameters
params <- list(
  booster = "gbtree",
  objective = "reg:squarederror",
  eta = 0.01,
  max_depth = 6,
  eval_metric = "rmse"
)

# Train the model without watchlist and other bells and whistles
xgb_model <- xgboost(
  data = as.matrix(train_data_one_hot),
  label = train_labels,
```

```r
  params = params,
  nrounds = 500,
  print_every_n = 10000
)
```

```
## [1]  train-rmse:32.327059
## [500]    train-rmse:14.972216
```

```r
# Predicting using the test data
predictions <- predict(xgb_model, as.matrix(test_data_one_hot))


# Evaluate the Model
rmse <- sqrt(mean((predictions - test_data$INCIDENT_count)^2))
print(paste("Test RMSE: ", rmse))
```

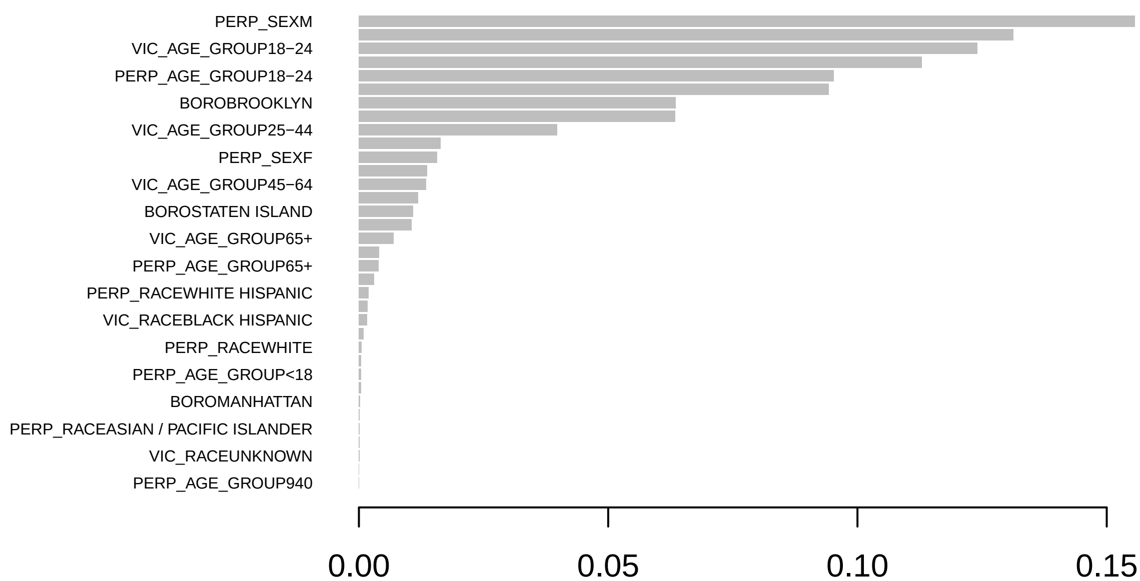
```
## [1] "Test RMSE:  30.0681964185064"
```

```r
# feature importance
# Plot feature importance
importance_matrix <- xgb.importance(feature_names = colnames(train_data_one_hot), model = xgb_model)
# print most important features:
print(importance_matrix)
```

```
##                              Feature         Gain        Cover    Frequency
##  1:                        PERP_SEXM 1.556409e-01 0.0382430153 0.0397688783
##  2:                         VIC_SEXM 1.312504e-01 0.0934356226 0.1112970711
##  3:              VIC_AGE_GROUP18-24 1.240847e-01 0.0419053247 0.0561864913
##  4:                   PERP_RACEBLACK 1.129471e-01 0.0633412401 0.0699740984
##  5:             PERP_AGE_GROUP18-24 9.523790e-02 0.0499410967 0.0527595138
##  6:                    VIC_RACEBLACK 9.425492e-02 0.1289152928 0.0728431959
##  7:                    BOROBROOKLYN 6.352378e-02 0.0287337351 0.0487746563
##  8:             PERP_AGE_GROUP25-44 6.342259e-02 0.0376950525 0.0453476788
##  9:              VIC_AGE_GROUP25-44 3.982458e-02 0.0551148249 0.0714484957
## 10:           PERP_AGE_GROUPUNKNOWN 1.642711e-02 0.0044044014 0.0223152022
## 11:                        PERP_SEXF 1.567575e-02 0.0457682995 0.0363817494
## 12:                        PERP_SEXU 1.373708e-02 0.0215197985 0.0166567045
## 13:              VIC_AGE_GROUP45-64 1.349872e-02 0.0178034554 0.0368200837
## 14:                        BOROBRONX 1.187610e-02 0.0539812063 0.0562263399
## 15:               BOROSTATEN ISLAND 1.088617e-02 0.0463034076 0.0217971708
## 16:             PERP_AGE_GROUP45-64 1.060215e-02 0.0297244259 0.0165371588
## 17:                VIC_AGE_GROUP65+ 6.947000e-03 0.0347467289 0.0190077705
## 18:          VIC_RACEWHITE HISPANIC 4.016476e-03 0.0375564821 0.0360629607
## 19:               PERP_AGE_GROUP65+ 3.939196e-03 0.0273746129 0.0114365411
## 20:            VIC_AGE_GROUPUNKNOWN 3.043967e-03 0.0179311319 0.0090854752
## 21:         PERP_RACEWHITE HISPANIC 2.001237e-03 0.0167334571 0.0213588364
## 22:              PERP_AGE_GROUP1020 1.774457e-03 0.0438971643 0.0066945607
## 23:          VIC_RACEBLACK HISPANIC 1.626774e-03 0.0158700071 0.0196453477
## 24:                PERP_RACEUNKNOWN 9.703695e-04 0.0096576550 0.0088862323
## 25:                  PERP_RACEWHITE 5.092439e-04 0.0036246165 0.0126319984
## 26:                     BOROQUEENS 4.848467e-04 0.0020999070 0.0166965531
## 27:               PERP_AGE_GROUP<18 4.556497e-04 0.0041518809 0.0106794182
```

```
## 28:           PERP_RACEBLACK HISPANIC 4.133791e-04 0.0046142178 0.0106794182
## 29:                     BOROMANHATTAN 2.760409e-04 0.0028905859 0.0095636581
## 30:                     VIC_RACEWHITE 1.834398e-04 0.0052576112 0.0166168559
## 31: PERP_RACEASIAN / PACIFIC ISLANDER 1.816227e-04 0.0041714899 0.0030284917
## 32:  VIC_RACEASIAN / PACIFIC ISLANDER 1.355442e-04 0.0021092758 0.0062163778
## 33:                   VIC_RACEUNKNOWN 1.208171e-04 0.0091659046 0.0049412234
## 34:                          VIC_SEXU 2.533738e-05 0.0001370452 0.0007571229
## 35:                 PERP_AGE_GROUP940 4.603092e-06 0.0011800266 0.0008766687
##                               Feature         Gain        Cover    Frequency
```

```r
xgb.plot.importance(importance_matrix)
```




```r
# compare rmse with the baseline rmse
baseline_predictions <- rep(mean(train_labels), length(test_labels))
baseline_rmse <- sqrt(mean((test_labels - baseline_predictions)^2))
print(paste("Baseline RMSE: ", baseline_rmse))
```

```
## [1] "Baseline RMSE:  37.3078263802946"
```

Thoughts after comparing rmse with baseline rmse

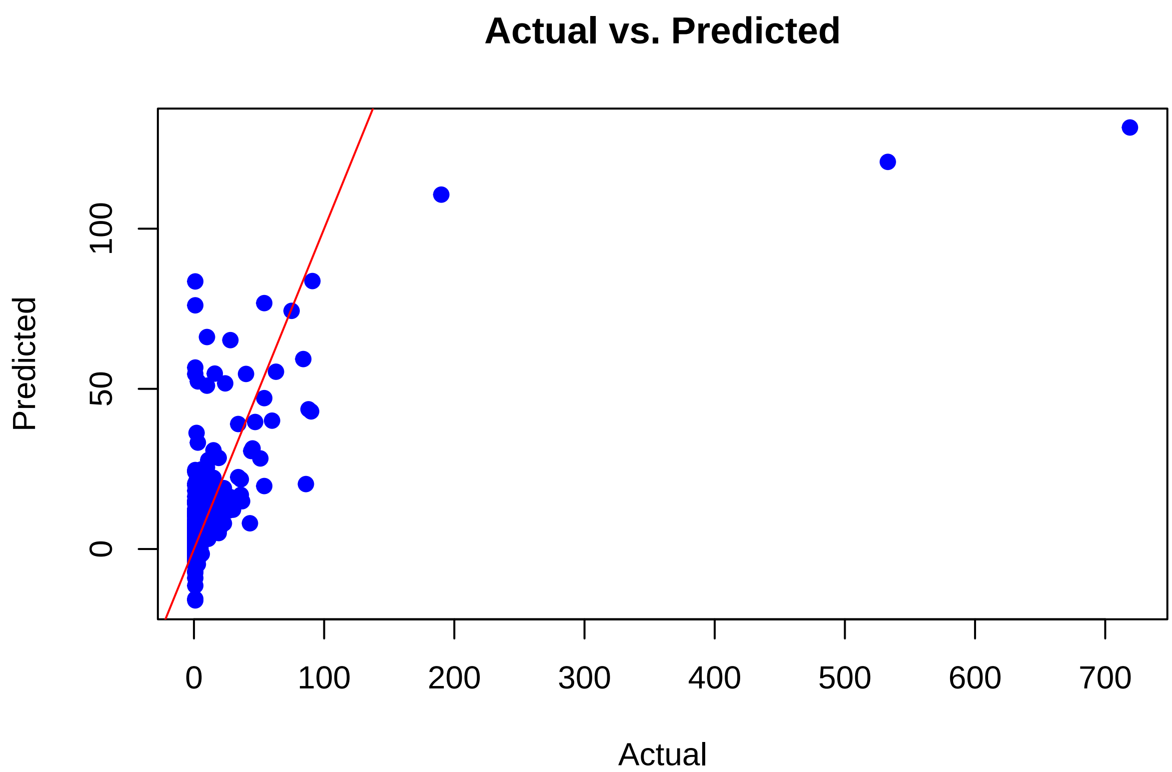
Given the baseline RMSE of 37.30783 and the RMSE of our XGBoost model at 30.0682, our model is performing significantly better than the baseline.

The RMSE of our XGBoost model is approximately 7.24 units lower than the baseline RMSE. This means that, on average, our model's predictions are closer to the actual values by about 7.24 units compared to simply predicting the mean of the INCIDENT_count for all observations.

Plotting the prediction

Scatter Plot: Actual vs. Predicted

```r
plot(test_labels, predictions, main="Actual vs. Predicted", xlab="Actual", ylab="Predicted", pch=19, co
abline(a=0, b=1, col="red")
```



Interpreting the results of the model

1. Model Performance:

- Baseline RMSE: 37.31
- XGBoost Test RMSE: 30.07
- The XGBoost model shows improved prediction accuracy compared to the baseline.

2. Top Features:

- PERP_SEXM (Perpetrator's Sex Male): Highly influential in predicting the target variable.
- VIC_SEXM (Victim's Sex Male): Holds significant weight in the model's decision-making.

- Age-Related Features: VIC_AGE_GROUP18-24 and PERP_AGE_GROUP18-24 show that age groups of both victim and perpetrator are vital for predictions.
- Race-Related Features: PERP_RACEBLACK and VIC_RACEBLACK suggest racial factors also play a role in predictions.

3. Less Influential Features:

- PERP_AGE_GROUP940: Has minimal influence on the model's predictions, indicating potential less variability or correlation with the outcome.

4. Conclusion:

Gender, age, and race emerge as significant predictors based on the model's feature importance. However, understanding the full context and domain is crucial for in-depth interpretation.

Possible Bias in Data and in analysis

1. Reporting Bias: Not all shooting incidents may be reported or recorded with equal likelihood. Incidents in certain areas or involving certain demographic groups might be over- or under-reported.
2. Analysing Bias:

- Exclusion of variables. Some important variable can be omitted from the model.
- Confirmation bias. Trying to confirm the existing notion about the relationship between two variables.
- P-value hacking. Re-running analyses until you get a statistically significant result by chance.
- Sampling bias. If some group is underrepresented in the data and this is not taken into account in the analysis.