NYPD_Shooting_Report

Wojciech Gajewski

2024-08-15

Load the dataset into R:

Tidy the Data

Perform data cleaning and tidying to make the dataset more manageable:

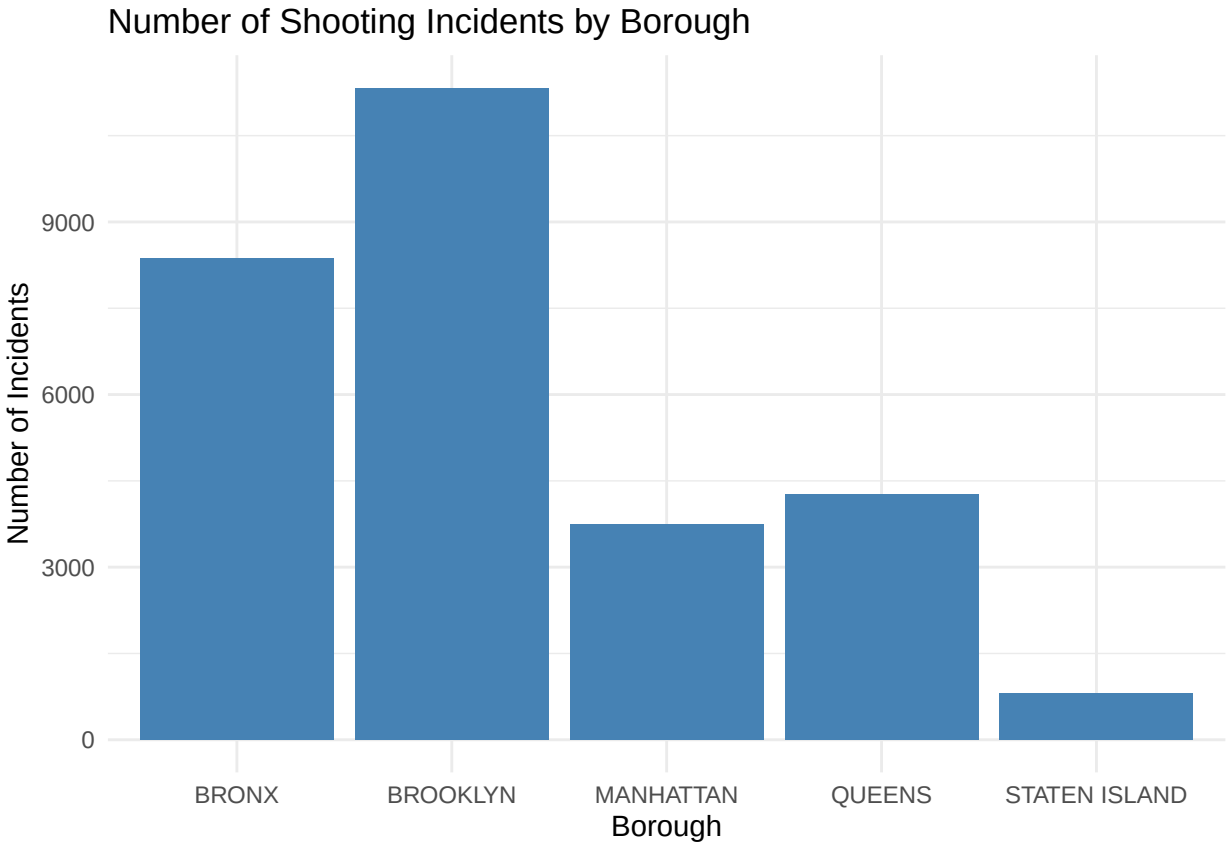
```
df <- df %>%
  select(INCIDENT_KEY, OCCUR_DATE, OCCUR_TIME, BORO, LOC_OF_OCCUR_DESC, PRECINCT, Latitude, Longitude, S

# Convert OCCUR_DATE to Date format and OCCUR_TIME to Time format
df <- df %>%
  mutate(OCCUR_DATE = as.Date(OCCUR_DATE, format="%m/%d/%Y"),
         OCCUR_TIME = hms::as_hms(OCCUR_TIME))

# Check for missing values
sum(is.na(df))
```

```
## [1] 118
```

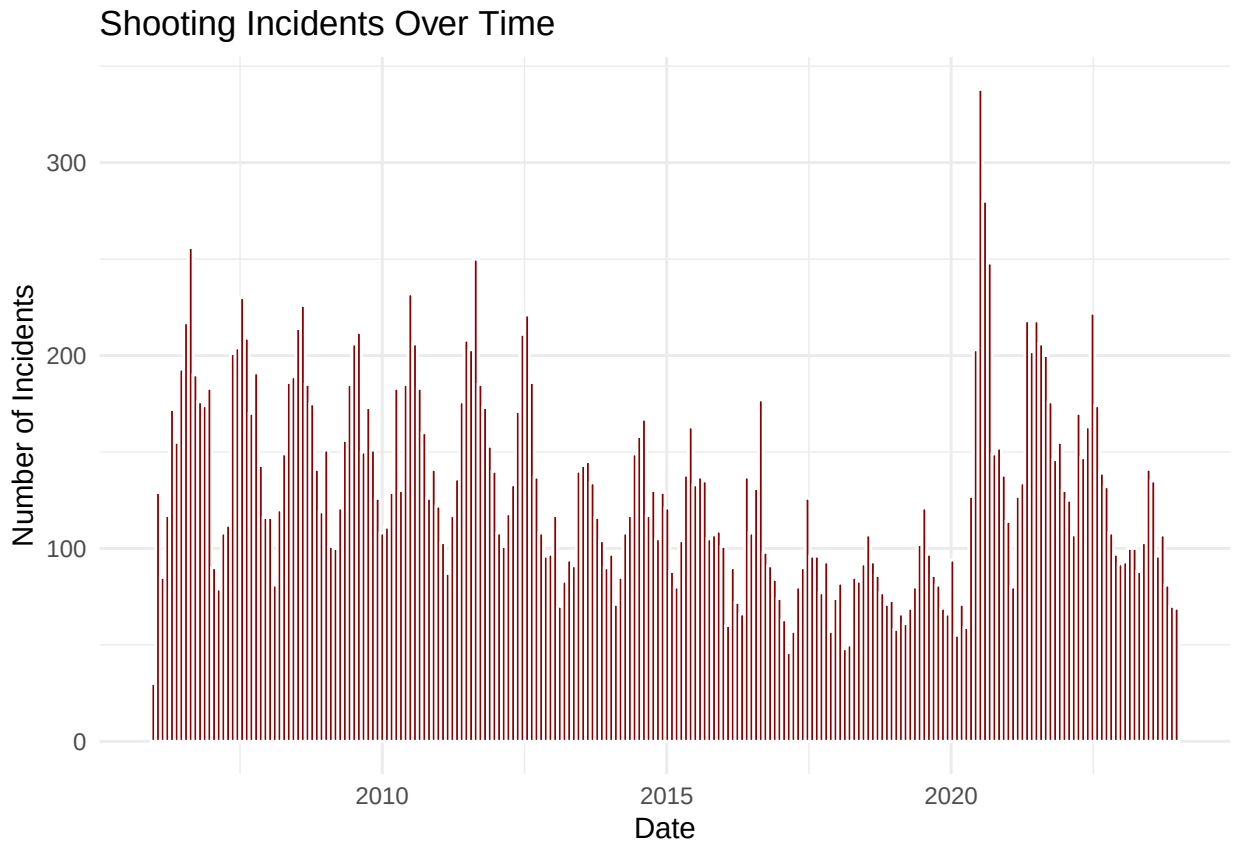
```
# Remove rows with missing values
df <- df %>% drop_na()
head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO LOC_OF_OCCUR_DESC PRECINCT
## 1    279683077 2023-12-29   03:43:00   QUEENS            INSIDE      113
## 2    279709792 2023-12-29   21:22:00 BROOKLYN           OUTSIDE       75
## 3    279758069 2023-12-29   18:40:00    BRONX           OUTSIDE       40
## 4    279609499 2023-12-27   19:47:00    BRONX           OUTSIDE       42
## 5    279547333 2023-12-26   23:43:00   QUEENS           OUTSIDE      106
## 6    279547332 2023-12-26   23:31:00    BRONX           OUTSIDE       46
##   Latitude Longitude STATISTICAL_MURDER_FLAG
## 1 40.68554 -73.77277                  false
## 2 40.65695 -73.87651                  false
## 3 40.81238 -73.90494                  false
## 4 40.82758 -73.88625                  false
## 5 40.68888 -73.81735                  false
## 6 40.85295 -73.90318                  false
```

Visualization 1: Number of Incidents by Borough



Visualization 2: Shooting Incidents Over Time



Build a Simple Model

We'll build a simple model to predict whether an incident was a murder or not based on the available features.

Model: Random Forest

```
# Convert STATISTICAL_MURDER_FLAG to a factor
df$STATISTICAL_MURDER_FLAG <- as.factor(df$STATISTICAL_MURDER_FLAG)

set.seed(123)
train_index <- createDataPartition(df$STATISTICAL_MURDER_FLAG, p = 0.7, list = FALSE)
train_data <- df[train_index, ]
test_data <- df[-train_index, ]

colSums(is.na(train_data))
```

```
##            INCIDENT_KEY               OCCUR_DATE               OCCUR_TIME
##                       0                        0                        0
##                    BORO       LOC_OF_OCCUR_DESC                 PRECINCT
##                       0                        0                        0
##                Latitude               Longitude STATISTICAL_MURDER_FLAG
```

```
##                        0                        0                        0
```

```
preProcess_missingdata_model <- preProcess(train_data, method = 'medianImpute')
train_data <- train_data[, colSums(is.na(train_data)) == 0]

rf_model <- randomForest(STATISTICAL_MURDER_FLAG ~ ., data = train_data, importance = TRUE)
predictions <- predict(rf_model, newdata = test_data)

confusion_matrix <- confusionMatrix(predictions, test_data$STATISTICAL_MURDER_FLAG)
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction false true
##      false  6628 1442
##      true    266  214
##
##                Accuracy : 0.8002
##                  95% CI : (0.7916, 0.8087)
##     No Information Rate : 0.8063
##     P-Value [Acc > NIR] : 0.9242
##
##                   Kappa : 0.1241
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9614
##             Specificity : 0.1292
##          Pos Pred Value : 0.8213
##          Neg Pred Value : 0.4458
##              Prevalence : 0.8063
##          Detection Rate : 0.7752
##    Detection Prevalence : 0.9439
##       Balanced Accuracy : 0.5453
##
##        'Positive' Class : false
##
```

Summary of Bias in the Model:

Accuracy and NIR:

The model's accuracy (80.02%) is close to the No Information Rate (80.63%), suggesting it mostly predicts the majority class ("false" for non-murder).

Kappa Statistic:

A low Kappa (0.1241) indicates poor performance beyond random chance, highlighting the model's difficulty in handling class imbalance.

Sensitivity and Specificity:

Sensitivity (96.14% for "false"): The model is highly sensitive to predicting non-murders. Specificity (12.92% for "true"): The model struggles to correctly identify murders, reflecting bias towards the majority class.

Bias Implications:

The model is biased towards predicting the majority class (non-murders), leading to poor detection of the minority class (murders). This imbalance is critical, especially in safety-related predictions, where identifying murders accurately is crucial.