NYPD Shooting Incident

2023-10-03

Abstract

In the provided R Markdown document, an analysis is presented based on the 'NYPD Shooting Incidents Dataset,' which is sourced from https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD. This dataset encompasses comprehensive records of shooting incidents that transpired within New York City, spanning from January 1, 2016, to December 31, 2022. The analysis delves into various aspects of this dataset to uncover insights and patterns within the context of these incidents over the specified timeframe.

Goals of The Analysis

For this analysis, I set out with specific objectives in mind. Firstly, I aimed to identify the New York City borough with the highest incidence of accidents. Additionally, I sought to determine the predominant racial profile of the main perpetrators within that particular borough. To achieve these goals, I employed a combination of meticulous data cleaning, informative tables, and visually engaging representations to extract meaningful insights from the dataset.

Libraries Needed for the Analysis

```
library(tidyverse)
library(dplyr)
library(tibble)
library(forecast)
library(knitr)
library(kableExtra)
```

```
#Read in Data set and Preview the data set
nypd <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
```

Here, I read in the NYPD CSV file and create a variable name called npyd to access it.

```
#Removing unwanted columns in clean_nypd
clean_nypd <- nypd %>%
  select(-LOC_OF_OCCUR_DESC, -LOC_CLASSFCTN_DESC, - LOCATION_DESC, - Lon_Lat, -INCIDENT_KEY)

#head(clean_nypd) #Preview of the data with removed unwanted columns
```

Looking at the data provided, I created another variable named clean_nypd which contained a copy of the original file. I did this so that the orginal file remains untouched. Upon observing the data, I removed columns that I was not interested in analyzing and columns that contained a majority of empty cells.

```
#Changing variable type to appropriate type
clean_nypd %>%
  mutate(OCCUR_DATE = as.Date(OCCUR_DATE, format = "%m/%d/%Y"),
         OCCUR_TIME = as.POSIXct(OCCUR_TIME, format = "%H:%M:%S"),
         BORO = as.factor(BORO),
         PRECINCT = as.factor(PRECINCT),
```

Table 1: Counts of Incidents in Boroughs

Boroughs	Frequency
BRONX	7937
BROOKLYN	10933
MANHATTAN	3572
QUEENS	4094
STATEN ISLAND	776

```
         JURISDICTION_CODE = as.factor(JURISDICTION_CODE),
         PERP_AGE_GROUP = as.factor(PERP_AGE_GROUP),
         PERP_SEX = as.factor(PERP_SEX),
         PERP_RACE = as.factor(PERP_RACE),
         VIC_AGE_GROUP = as.factor(VIC_AGE_GROUP),
         VIC_RACE = as.factor(VIC_RACE))
```

```
## # A tibble: 27,312 x 16
##    OCCUR_DATE OCCUR_TIME          BORO     PRECINCT JURISDICTION_CODE
##    <date>     <dttm>              <fct>    <fct>    <fct>
##  1 2021-05-27 1970-01-01 21:30:00 QUEENS   105      0
##  2 2014-06-27 1970-01-01 17:40:00 BRONX    40       0
##  3 2015-11-21 1970-01-01 03:56:00 QUEENS   108      0
##  4 2015-10-09 1970-01-01 18:30:00 BRONX    44       0
##  5 2009-02-19 1970-01-01 22:58:00 BRONX    47       0
##  6 2020-10-21 1970-01-01 21:36:00 BROOKLYN 81       0
##  7 2012-06-17 1970-01-01 22:47:00 QUEENS   114      0
##  8 2010-03-08 1970-01-01 19:41:00 BROOKLYN 81       0
##  9 2012-02-05 1970-01-01 05:45:00 QUEENS   105      0
## 10 2012-08-26 1970-01-01 01:10:00 QUEENS   101      0
## # i 27,302 more rows
## # i 11 more variables: STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <fct>,
## #   PERP_SEX <fct>, PERP_RACE <fct>, VIC_AGE_GROUP <fct>, VIC_SEX <chr>,
## #   VIC_RACE <fct>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>
```

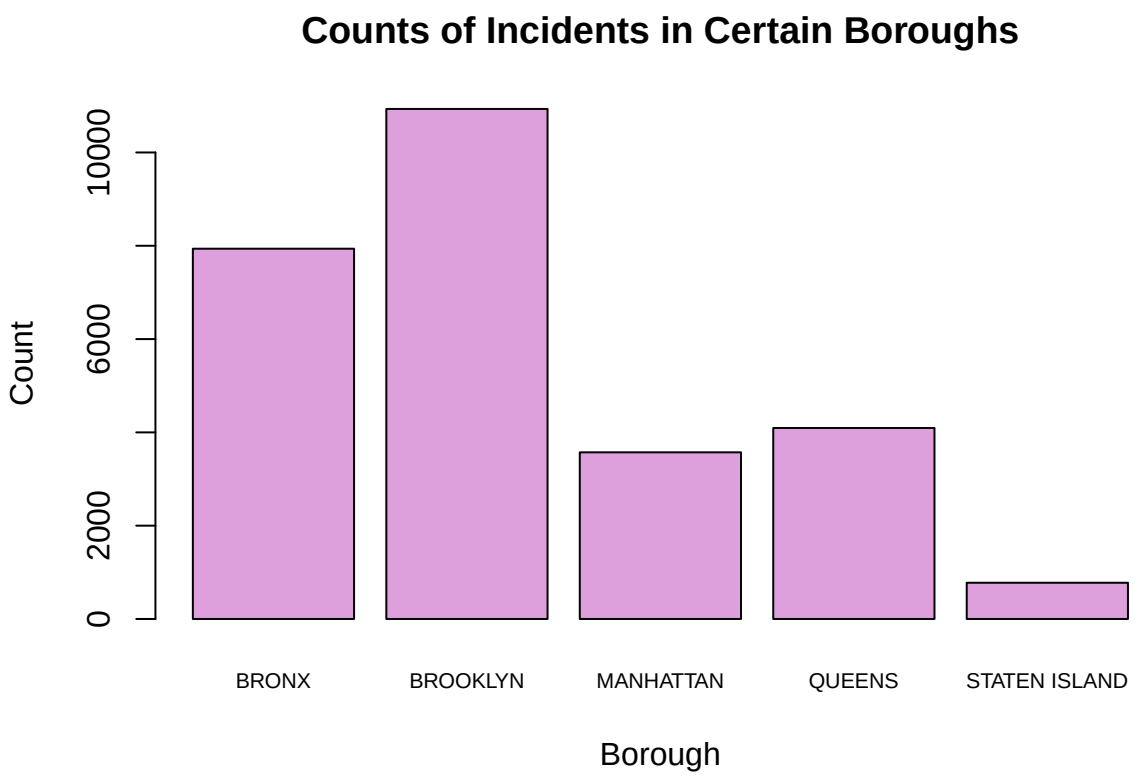
Here, I changed the variable type of each variable so that the variable type makes more intuitive sense. Doing so also allows me to work with the data more easily.

Creating Tables and Visual Analysis

```
#Creating a table that gathers the counts of observation in each BORO and plotted for a visualization
boro_counts_table <- table(clean_nypd$BORO)
kable(
  boro_counts_table,
  caption = "Counts of Incidents in Boroughs",
  col.names = c("Boroughs", "Frequency")
) %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE)
```

```
barplot(boro_counts_table,
        main = "Counts of Incidents in Certain Boroughs",
        xlab = "Borough",
        ylab = "Count",
```

```
        col = "plum",
        border = "black",
        ylim = c(0, max(boro_counts_table) + 15),
        names.arg = NULL,
        cex.names = 0.7)
```



Since a majority of these incidents have taken place within the *Brooklyn Borough*, let's delve deeper and examine the predominant racial demographic among the main perpetrators in Brooklyn

```
# Filter the clean_nypd data for Brooklyn
brooklyn_data <- clean_nypd[clean_nypd$BORO == "BROOKLYN", ]

# Create the table for perpetrator race in Brooklyn
perp_race_group_table <- table(brooklyn_data$PERP_RACE)

# Create a nice looking table with kable function
kable(
  perp_race_group_table,
  caption = "Perpetrator Race Group Table for Brooklyn",
  col.names = c("Race Group", "Frequency")
) %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE)
```

Here, I created a two tables which display the counts of people in boroughs and the counts of Perpetrator Race for Brooklyn. For a visual aid, I created a plot using the information from the table. From the graphs, we learn that a majority of the crimes happened in Brooklyn and that a majority of perpetrators were Black.

Table 2: Perpetrator Race Group Table for Brooklyn

Race Group	Frequency
(null)	263
ASIAN / PACIFIC ISLANDER	39
BLACK	4814
BLACK HISPANIC	263
UNKNOWN	769
WHITE	82
WHITE HISPANIC	422

Time Series Model

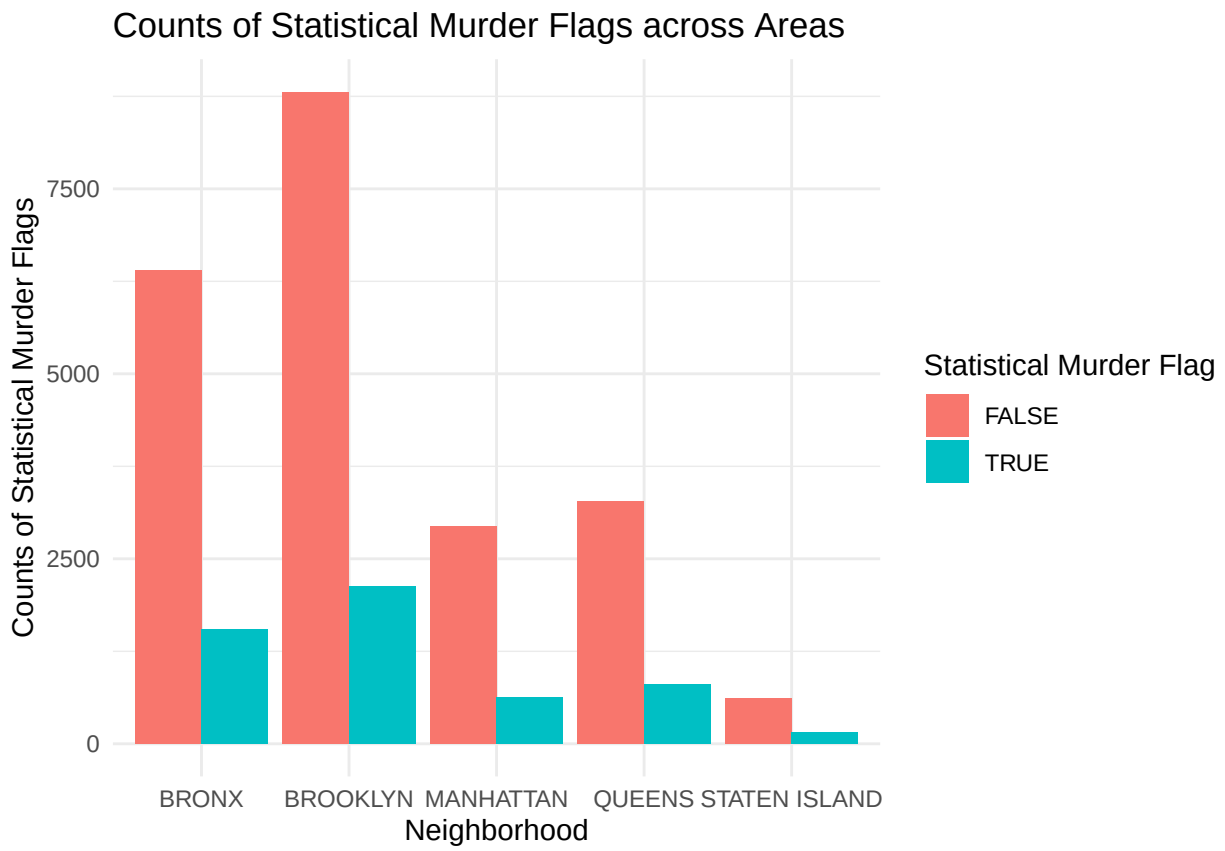
```
#Arrange the time to prepare the data for time series modeling
clean_nypd_1 <- clean_nypd %>%
  arrange(OCCUR_TIME)

#Confirm that all the dates follow the same format of hours/minutes/seconds
clean_nypd_1$OCCUR_TIME <- as.POSIXct(clean_nypd_1$OCCUR_TIME, format = "%H:%M:%S")

#create a new variable and use the ts (time series) function with frequency = 24 for daily observations
nypd_time_series <- ts(clean_nypd_1$OCCUR_TIME, frequency =24)

#auto.arima fits an auto regressive integrated moving average model
arima_model_nypd <- auto.arima(nypd_time_series)
summary(arima_model_nypd)
```

```
## Series: nypd_time_series
## ARIMA(5,2,0)
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5
##       -0.8529  -0.6781  -0.5158  -0.3348  -0.1689
## s.e.   0.0060   0.0076   0.0081   0.0076   0.0060
##
## sigma^2 = 212.1:  log likelihood = -111902.3
## AIC=223816.5   AICc=223816.5   BIC=223865.8
##
## Training set error measures:
##                         ME     RMSE      MAE          MPE       MAPE    MASE
## Training set -3.156433e-05 14.56307 5.911625 -4.79118e-05 0.05615105 0.07789
##                      ACF1
## Training set -0.02558891
```

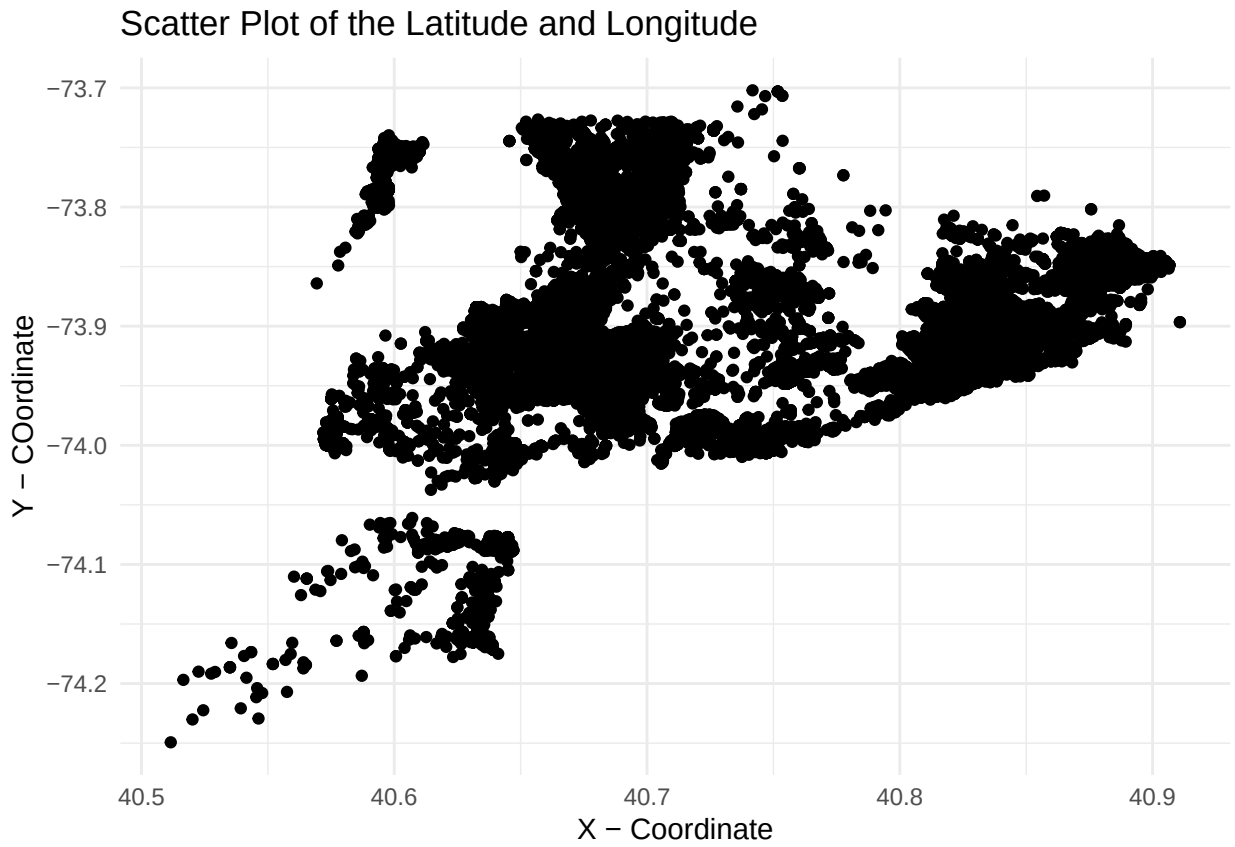
Creating a time series model can be helpful when we want to observe temporal patterns that may not be immediately apparent. Creating this type of model can also be used to forecast future events and occurrence which can enhance public safety. Moreover, the judicious allocation of police resources becomes far more efficient, ensuring that funding is appropriately directed toward essential law enforcement activities rather than left to arbitrary budgeting

Including Plots

```
#Bar Graph of Counts of Statistical Murder Flags across different boroughs
ggplot(clean_nypd, aes(x = BORO, fill = STATISTICAL_MURDER_FLAG)) +
  geom_bar(position = "dodge", stat = "count") +
  labs(x = "Neighborhood",
       y = "Counts of Statistical Murder Flags",
       fill = "Statistical Murder Flag",
       title = "Counts of Statistical Murder Flags across Areas") +
  theme_minimal()
```




```
ggplot(clean_nypd, aes(x = Latitude, y = Longitude)) +
  geom_point() +
  labs(
    x = "X - Coordinate",
    y = "Y - COordinate",
    title = "Scatter Plot of the Latitude and Longitude"
  ) +
  theme_minimal()
```



In this step of the analysis, two graphs were made. One bar graph which displayed how many statistical murder flags there were in each borough, and one scatter graph of the XY coordinates of the crime. From the previous analysis, we learned that Brooklyn has the most counts in crime. Brooklyn's coordinates are 40.650002, -73.949997. Observing the Scatter Plot of the Latitude and Longitude, we observe that a majority of the clustering appears around Brooklyn's coordinate.

Bias Identification and Conclusion

In conclusion, this analysis of NYPD shooting incident data has provided valuable insights into the patterns and dynamics of such incidents within the boroughs of New York City. However, it is essential to acknowledge the potential sources of bias that may impact the accuracy and generalization of our findings. One possible bias that could be present in our analysis is *location bias*. The data primarily focuses on specific neighborhoods, including the *Bronx*, *Manhattan*, *Brooklyn*, *Queens*, and *Staten Island*. Recognizing the diversity of New York City, it's important to note that police activities can significantly vary by location. It makes sense that a majority of the clustered data points laid wihtin a certain region as the crimes all centered in New York, specifically Brooklyn. Brooklyn, with its status as the most densely populated borough among all, provides a good explanation for its prominence in crime statistics and the prevalence of murder incidents. Such variations may influence the nature of reported incidents, potentially skewing the overall representation of NYPD shooting incidents. Furthermore, underreporting or overreporting in certain neighborhoods can introduce bias into our analysis, as these discrepancies may not accurately reflect the true incidence rates. A majority of the perpetrators were Black and there may be some bias in that since they are generally more biased and targeted against.Additionally, it's worth acknowledging the data's completeness issues. The dataset contained numerous rows and columns with missing values, which posed a challenge to our analysis. To address this, I removed columns that were not relevant for our visualizations or analytical goals. While this helped streamline the analysis, it's important to recognize that data completeness issues can affect the overall validity of our findings.

Summary

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22621)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] kableExtra_1.3.4 knitr_1.44       forecast_8.21.1  lubridate_1.9.3
##  [5] forcats_1.0.0    stringr_1.5.0    dplyr_1.1.3      purrr_1.0.2
##  [9] readr_2.1.4      tidyr_1.3.0      tibble_3.2.1     ggplot2_3.4.3
## [13] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.4      xfun_0.40         lattice_0.21-8    tzdb_0.4.0
##  [5] quadprog_1.5-8    vctrs_0.6.3       tools_4.3.1       generics_0.1.3
##  [9] curl_5.0.2        parallel_4.3.1    fansi_1.0.4       xts_0.13.1
## [13] pkgconfig_2.0.3   webshot_0.5.5     lifecycle_1.0.3   farver_2.1.1
## [17] compiler_4.3.1    munsell_0.5.0     htmltools_0.5.6   yaml_2.3.7
## [21] pillar_1.9.0      crayon_1.5.2      nlme_3.1-162      fracdiff_1.5-2
## [25] tidyselect_1.2.0  rvest_1.0.3       digest_0.6.33     stringi_1.7.12
## [29] labeling_0.4.3    tseries_0.10-54   fastmap_1.1.1     grid_4.3.1
## [33] colorspace_2.1-0  cli_3.6.1         magrittr_2.0.3    utf8_1.2.3
## [37] withr_2.5.1       scales_1.2.1      bit64_4.0.5       timechange_0.2.0
## [41] TTR_0.24.3        rmarkdown_2.25    httr_1.4.7        quantmod_0.4.25
## [45] bit_4.0.5         nnet_7.3-19       timeDate_4022.108 zoo_1.8-12
## [49] hms_1.1.3         urca_1.3-3        evaluate_0.22     lmtest_0.9-40
## [53] viridisLite_0.4.2 rlang_1.1.1       Rcpp_1.0.11       glue_1.6.2
## [57] xml2_1.3.5        svglite_2.1.1     rstudioapi_0.15.0 vroom_1.6.3
## [61] R6_2.5.1          systemfonts_1.0.4
```