NYPD_Shooting_Incident

John Schlangen

2023-08-08

NYPD Shooting Incident

The following project is conducted using a publicly available dataset with data regarding shooting incidents in New York City since 2006. The link below is where the .csv file can be found for this analysis. https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD

```
df <- read_csv('https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD')
```

```
## Rows: 27312 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.
```

Initial Data Cleaning

After reviewing the data, the ideas that appear to be the most interesting to investigate further include incidents by borough, murders by borough, and the sex of the victim/perpetrator.

With that understanding, we can remove unnecessary fields and begin our analysis. We will select 5 fields that we will use, remove any null values, and ensure that the OCCUR_DATE field is a date type.

```
df <- df %>%
  select(OCCUR_DATE, BORO, VIC_SEX, PERP_SEX, STATISTICAL_MURDER_FLAG) %>%
  drop_na() %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE))

summary(df)
```

```
##    OCCUR_DATE            BORO              VIC_SEX            PERP_SEX
##  Min.   :2006-01-01   Length:18002       Length:18002       Length:18002
##  1st Qu.:2008-08-03   Class :character   Class :character   Class :character
##  Median :2011-11-12   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2013-05-07
##  3rd Qu.:2018-04-21
```

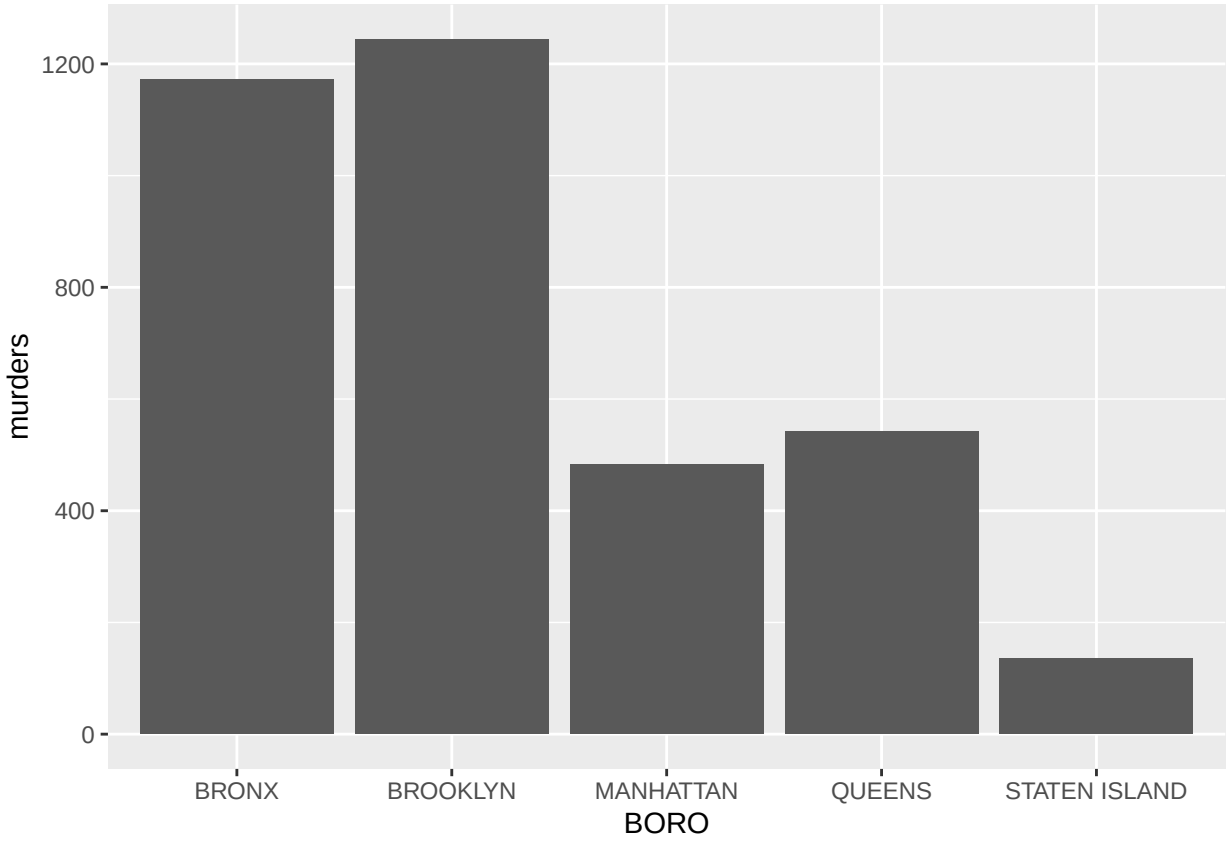
```
##  Max.   :2022-12-31
##  STATISTICAL_MURDER_FLAG
##  Mode :logical
##  FALSE:14425
##  TRUE :3577
##
##
##
```

With our data now clean, we can begin to do some initial questioning and data analysis to get a feel for our data. Based on the summary, we can see that approximately 19.9% of the incidents have the value `TRUE` for the field `STATISTICAL_MURDER_FLAG`.

Investigating further, we will look to see which boroughs have the most murders, and which boroughs have the highest % of female perpetrators.

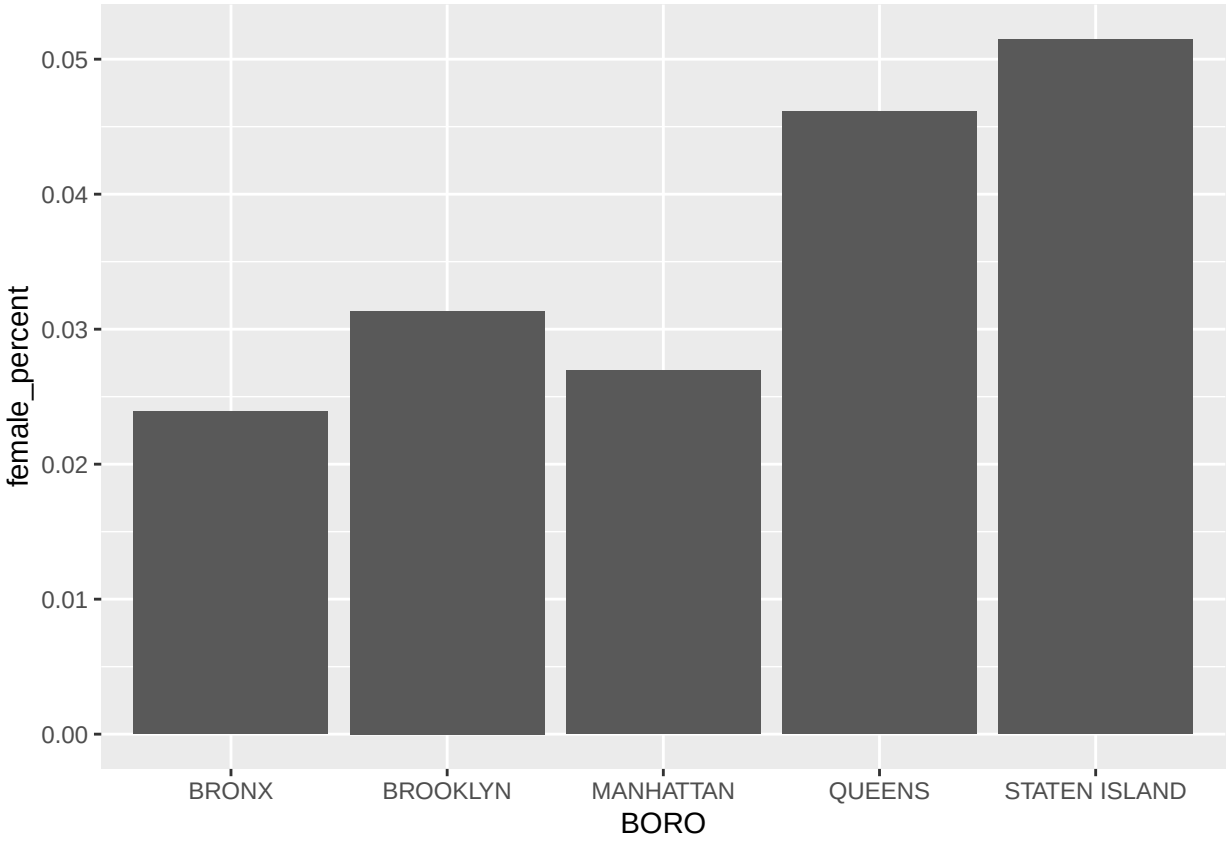
Boroughs with the Most Murders

```
most_murders <- df %>%
  group_by(BORO) %>%
  summarize(murders = sum(STATISTICAL_MURDER_FLAG)) %>%
  arrange(desc(murders))

most_murders %>%
  ggplot(aes(x = BORO, y = murders)) +
  geom_bar(stat = "identity")
```



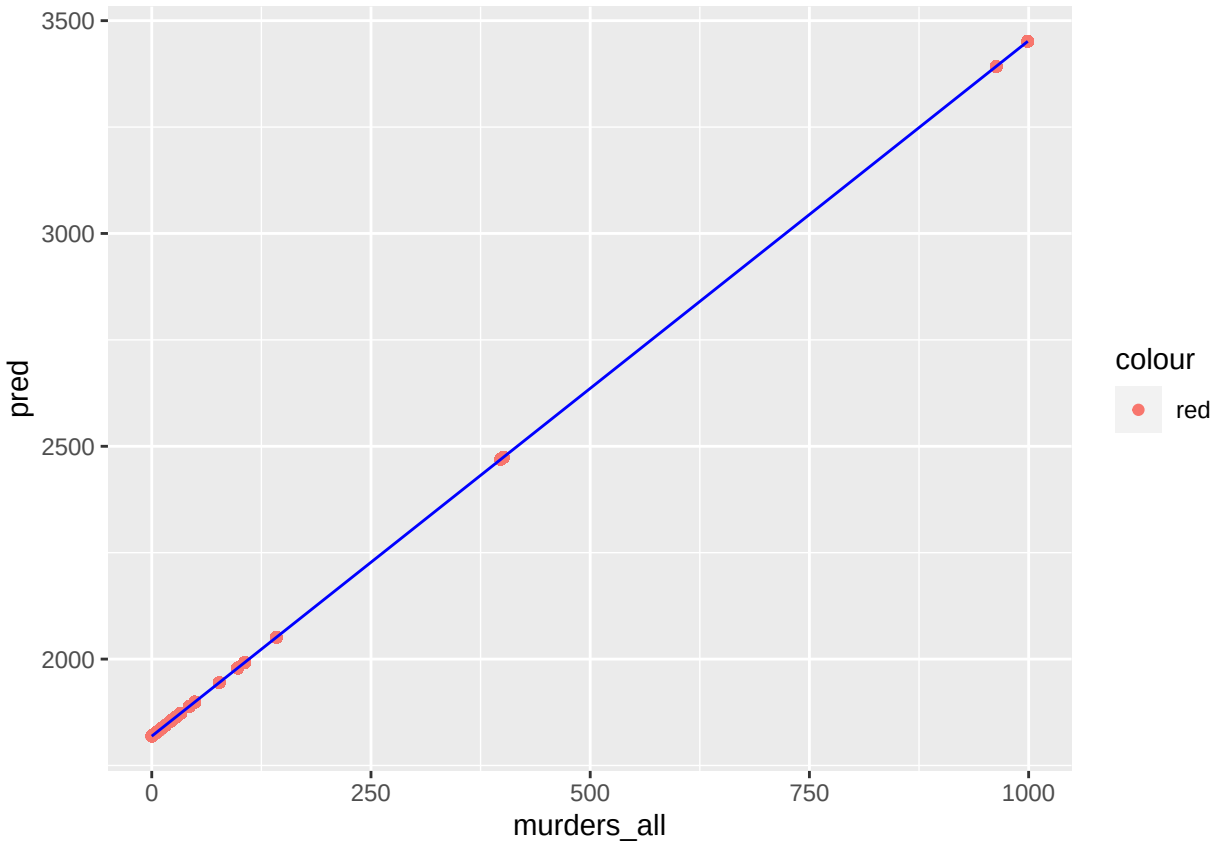
Boroughs with the Highest Female Perpetrators %

```
female <- df %>%
  group_by(BORO) %>%
  mutate(murders = STATISTICAL_MURDER_FLAG / sum(STATISTICAL_MURDER_FLAG)) %>%
  filter(PERP_SEX == 'F') %>%
  summarize(female_percent = sum(murders)) %>%
  arrange(desc(female_percent))

female %>%
  ggplot(aes(x = BORO, y = female_percent)) +
  geom_bar(stat = "identity")
```



Linear Model

```
model_df <- df %>%
  group_by(BORO, PERP_SEX, VIC_SEX) %>%
  mutate(murders_all = sum(STATISTICAL_MURDER_FLAG)) %>%
  group_by(PERP_SEX) %>%
  mutate(murders_perp = sum(STATISTICAL_MURDER_FLAG))

model <- lm(murders_perp ~ murders_all, data = model_df)

df %>%
  mutate(pred = predict(model)) %>%
  group_by(BORO, PERP_SEX, VIC_SEX) %>%
  mutate(murders_all = sum(STATISTICAL_MURDER_FLAG)) %>%
  ggplot(aes(x = murders_all, y = pred)) +
  geom_point(aes(color = 'red')) +
  geom_line(color = 'blue')
```



After the initial analysis, and looking at the first graph, we could see that Brooklyn had the highest number of murders based on the dataset. It was very interesting to see that, according to the second graph, Staten Island had the highest percentage of female perpetrators at just over 5% of the total murders for the borough.

Based on these findings, I would want to learn more regarding the populations and demographics of the boroughs. It would be interesting to understand why there are more female murder perpetrators in Staten Island than there are in the other boroughs.

Bias

I acknowledge the fact that there is bias everywhere, and it is especially important to mitigate bias whenever possible. In my analysis, I tried to reduce bias by calculating using ratios, and also by looking at female data for this project. Using ratios allowed me to factor population size into the equation, and using the female data allowed me to remove any personal bias/opinions as a male conducting this research.

```
sessionInfo
```

```
## function (package = NULL)
## {
##     z <- list()
##     z$R.version <- R.Version()
##     z$platform <- z$R.version$platform
##     if (nzchar(.Platform$r_arch))
##         z$platform <- paste(z$platform, .Platform$r_arch, sep = "/")
##     z$platform <- paste0(z$platform, " (", 8 * .Machine$sizeof.pointer,
##         "-bit)")
```

```
##     z$locale <- Sys.getlocale()
##     z$running <- osVersion
##     z$RNGkind <- RNGkind()
##     if (is.null(package)) {
##         package <- grep("^package:", search(), value = TRUE)
##         keep <- vapply(package, function(x) x == "package:base" ||
##             !is.null(attr(as.environment(x), "path")), NA)
##         package <- .rmpkg(package[keep])
##     }
##     pkgDesc <- lapply(package, packageDescription, encoding = NA)
##     if (length(package) == 0)
##         stop("no valid packages were specified")
##     basePkgs <- sapply(pkgDesc, function(x) !is.null(x$Priority) &&
##         x$Priority == "base")
##     z$basePkgs <- package[basePkgs]
##     if (any(!basePkgs)) {
##         z$otherPkgs <- pkgDesc[!basePkgs]
##         names(z$otherPkgs) <- package[!basePkgs]
##     }
##     loadedOnly <- loadedNamespaces()
##     loadedOnly <- loadedOnly[!(loadedOnly %in% package)]
##     if (length(loadedOnly)) {
##         names(loadedOnly) <- loadedOnly
##         pkgDesc <- c(pkgDesc, lapply(loadedOnly, packageDescription))
##         z$loadedOnly <- pkgDesc[loadedOnly]
##     }
##     z$matprod <- as.character(options("matprod"))
##     es <- extSoftVersion()
##     z$BLAS <- as.character(es["BLAS"])
##     z$LAPACK <- La_library()
##     l10n <- l10n_info()
##     if (!is.null(l10n["system.codepage"]))
##         z$system.codepage <- as.character(l10n["system.codepage"])
##     if (!is.null(l10n["codepage"]))
##         z$codepage <- as.character(l10n["codepage"])
##     class(z) <- "sessionInfo"
##     z
## }
## <bytecode: 0x0000020608f0d8b8>
## <environment: namespace:utils>
```