2020 NYC Shootings by Borough

May 20, 2021

This report will explore the 2020 shootings by borough in New York City. The raw dataset is sourced from the Data.Gov website. It contains details about each shooting reported to the NYPD from 2006 through 2020, across the five NYC boroughs.

I will import the data, tidy the data, perform preliminary analysis, and model the data. Visualizations are added throughout the report as an aid. Each step taken in this report is clearly marked with an individual header. Due to the assignment's requirements to show all code from each R Markdown block, all of the code results are placed in this document as well.

Step 1: Attach the libraries used in this report

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(ggplot2)
library(dplyr)
library(plotfunctions)
```

```
##
## Attaching package: 'plotfunctions'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(jtools)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

Step 2: Import the data and create a dataframe

```
url_in <- "https://data.cityofnewyork.us/api/views/833y-fsy8/"
file_name <- ("rows.csv")
url <- str_c(url_in, file_name)

shootings <- read_csv(url[1])
```

```
## Parsed with column specification:
## cols(
##   INCIDENT_KEY = col_integer(),
##   OCCUR_DATE = col_character(),
##   OCCUR_TIME = col_time(format = ""),
##   BORO = col_character(),
##   PRECINCT = col_integer(),
##   JURISDICTION_CODE = col_integer(),
##   LOCATION_DESC = col_character(),
##   STATISTICAL_MURDER_FLAG = col_character(),
##   PERP_AGE_GROUP = col_character(),
##   PERP_SEX = col_character(),
##   PERP_RACE = col_character(),
##   VIC_AGE_GROUP = col_character(),
##   VIC_SEX = col_character(),
##   VIC_RACE = col_character(),
##   X_COORD_CD = col_number(),
##   Y_COORD_CD = col_number(),
##   Latitude = col_double(),
##   Longitude = col_double(),
##   Lon_Lat = col_character()
## )
```

```
shootings_df <- data.frame(shootings)
attach(shootings_df)
```

Step 3: Tidy the data

First, I will check which columns contain missing data, and their data types.

```
names(which(colSums(is.na(shootings_df)) > 0))
```

```
## [1] "JURISDICTION_CODE" "LOCATION_DESC"     "PERP_AGE_GROUP"
## [4] "PERP_SEX"          "PERP_RACE"
```

```
str(JURISDICTION_CODE)
```

```
##  int [1:23568] 0 0 0 0 0 0 0 0 2 0 ...
```

```
str(LOCATION_DESC)
```

```
##  chr [1:23568] NA NA NA "PVT HOUSE" NA NA NA "MULTI DWELL - APT BUILD" ...
```

```
str(PERP_AGE_GROUP)
```

```
##  chr [1:23568] NA "<18" "18-24" "25-44" "25-44" "45-64" "18-24" NA ...
```

```
str(PERP_SEX)
```

```
##  chr [1:23568] NA "M" "M" "M" "M" "M" "M" NA "M" "M" NA "M" "F" "M" ...
```

```
str(PERP_RACE)
```

```
##  chr [1:23568] NA "BLACK" "WHITE HISPANIC" "BLACK" "BLACK HISPANIC" ...
```

After investigating the types of those five variables, I found all but "JURISDICTION_CODE" to be a character variable. I will replace the all the missing values in my dataset with the word "UNSPECIFIED." The numeric "JURISDICTION_CODE" does not cause concern because I will not be using that variable in my overall analysis.

```
shootings_df[is.na(shootings_df)] = "UNSPECIFIED"
```

Since my analysis uses 2020 data, I need a way to subset my dataset for 2020 observations only. So I will convert the variable "OCCUR_DATE" to a date format. After re-formatting this variable, I will extract the year and subset my dataframe to contain only 2020 data.

```
shootings_df$newdate <- as.Date(shootings_df$OCCUR_DATE,"%m/%d/%Y")
shootings_df$syear <- year(shootings_df$newdate)
```

```
shootings_2020 <- subset(shootings_df, syear == 2020)
shootings_2020 <- as.data.frame(shootings_2020)
attach(shootings_2020)
```

```
## The following objects are masked from shootings_df:
##
##     BORO, INCIDENT_KEY, JURISDICTION_CODE, Latitude,
##     LOCATION_DESC, Lon_Lat, Longitude, OCCUR_DATE, OCCUR_TIME,
##     PERP_AGE_GROUP, PERP_RACE, PERP_SEX, PRECINCT,
##     STATISTICAL_MURDER_FLAG, VIC_AGE_GROUP, VIC_RACE, VIC_SEX,
##     X_COORD_CD, Y_COORD_CD
```

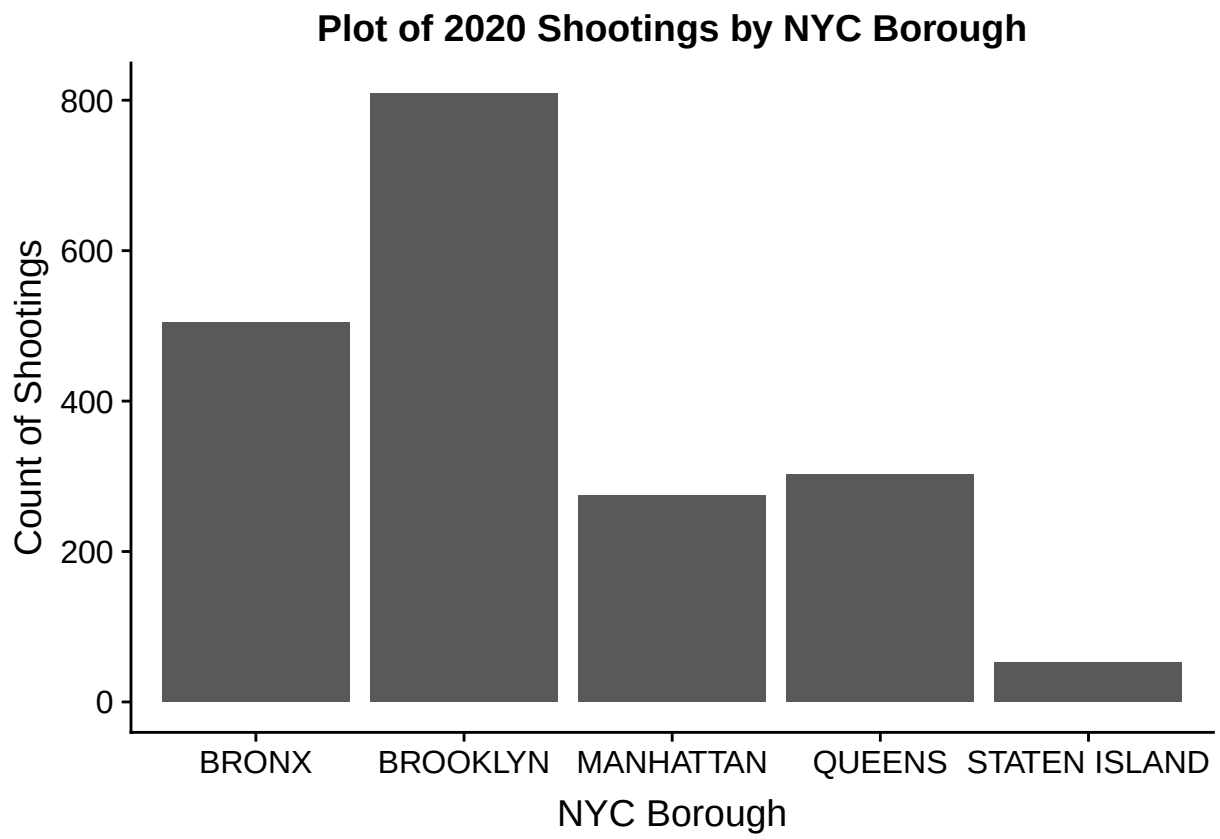
Step 4: Analyze the data

Since the shootings all had different identification numbers, I will create a variable that contains the count of shootings by borough.

```
shootings_2020<-shootings_2020%>%group_by(BORO)%>%mutate(count_shootings=n())
```

Since I want to see the relationship between borough and number of shootings, I will plot the data to get a preliminary glance.

```
ggplot(shootings_2020, aes(x = shootings_2020$BORO)) + geom_bar() + ggtitle("Plot of 2020 Shootings by
```



Step 5: Model the data

As seen in the prior graph, Brooklyn has the highest number of shootings in 2020. However, I want to explore this further by running a linear regression model. My regression model will include my shooting occurrence variable as the dependent variable, and each borough as an explanatory variable.

Since I want to regress the shooting occurences against the different boroughs, I will create indicator variables for every borough.

```
shootings_2020$bronx <- ifelse(shootings_2020$BORO == 'BRONX',1,0)
shootings_2020$brooklyn <- ifelse(shootings_2020$BORO == 'BROOKLYN',1,0)
shootings_2020$manhattan <- ifelse(shootings_2020$BORO == 'MANHATTAN',1,0)
shootings_2020$queens <- ifelse(shootings_2020$BORO == 'QUEENS',1,0)
shootings_2020$staten <- ifelse(shootings_2020$BORO == 'STATEN ISLAND',1,0)
```

After creating these indicator variables, I am able to build my linear regression model. My model leaves out one of the indicator variables to avoid redundancy. I will run my model and provide effect plots of my model.

```
model <- lm(count_shootings ~ bronx + brooklyn + queens + staten, data = shootings_2020)
summary(model)
```

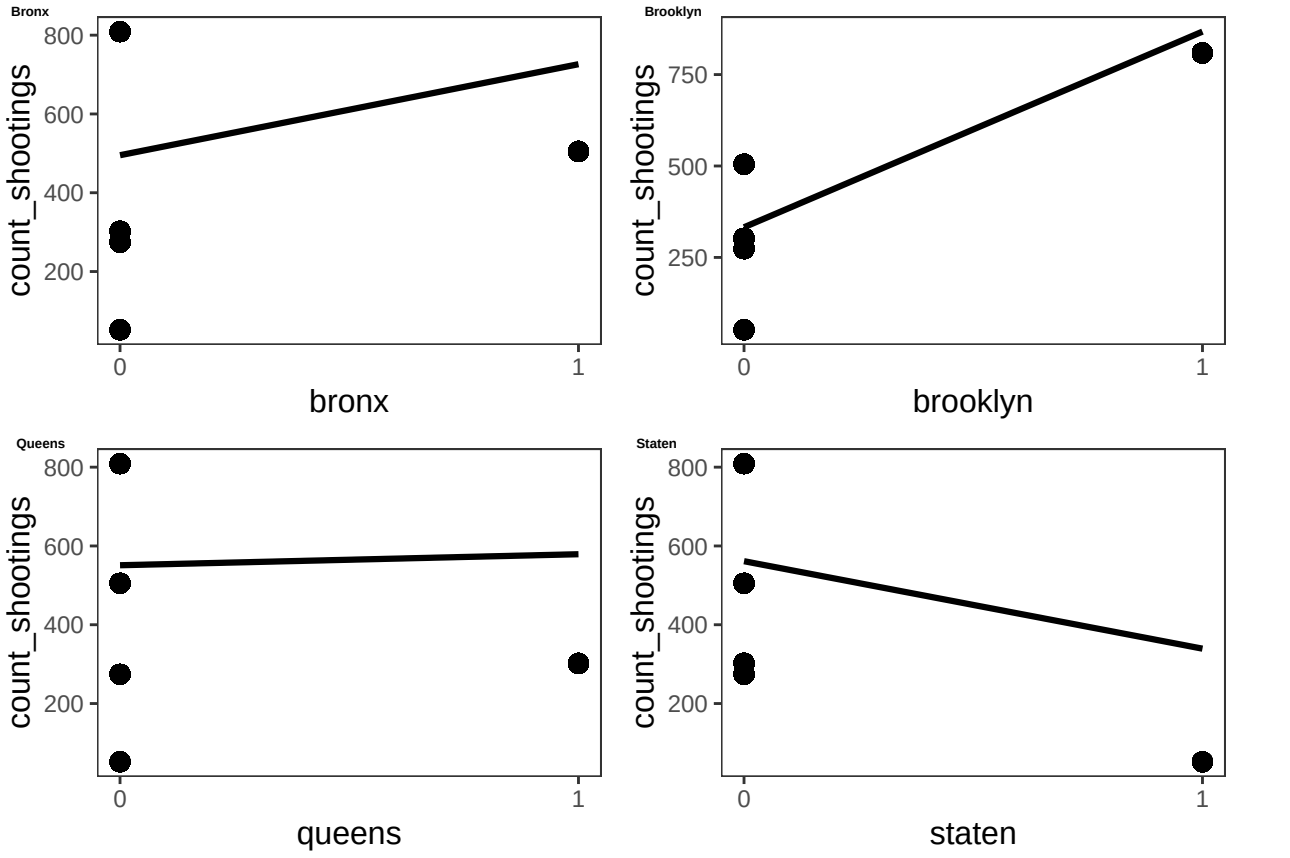
```
## Warning in summary.lm(model): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = count_shootings ~ bronx + brooklyn + queens + staten,
##     data = shootings_2020)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.427e-12  0.000e+00  3.900e-15  3.900e-15  2.641e-12
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  2.740e+02  7.455e-15  3.675e+16   <2e-16 ***
## bronx        2.310e+02  9.260e-15  2.495e+16   <2e-16 ***
## brooklyn     5.350e+02  8.626e-15  6.202e+16   <2e-16 ***
## queens       2.800e+01  1.030e-14  2.719e+15   <2e-16 ***
## staten      -2.220e+02  1.867e-14 -1.189e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.234e-13 on 1937 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.766e+33 on 4 and 1937 DF,  p-value: < 2.2e-16
```

Creating the effect plots

```
p1 <- effect_plot(model, pred = bronx, interval = TRUE, plot.points = TRUE)
p2 <- effect_plot(model, pred = brooklyn, interval = TRUE, plot.points = TRUE)
p3 <- effect_plot(model, pred = queens, interval = TRUE, plot.points = TRUE)
p4 <- effect_plot(model, pred = staten, interval = TRUE, plot.points = TRUE)
plot_grid(p1, p2, p3, p4, labels = c('Bronx', 'Brooklyn', 'Queens','Staten'), label_size = 5)
```



Step 6: Identify Bias

I had expected to see that lower-income boroughs would have a higher amount of shootings. This is due to my understanding that lower-income neighborhoods typically have a higher crime rate. With this bias, I was expecting to see the Bronx have the highest number of shootings, since its median household income is the least when compared against the other 4 boroughs (Chen, 2018).

However, when looking at my results, I saw that the Bronx actually had a lower number of shootings than Brooklyn, which contradicted my expectations. Recognizing this bias was important, and as a result, I realize it is best to keep an open mind before performing any future analysis.

Step 7: Limitations and Conclusion

As seen in my regression results, I received a warning stating that my model might be a "perfect fit." I know there is no such thing as a "perfect fit," so this is concerning to me. One reason I might have gotten these results is by using a count of shootings by borough as my dependent variable. Additionally, it might be due to using only indicator variables in my model. Please understand that the dataset contained mostly categorical data, so it was difficult for me to form a meaningful model. I think a deeper understanding of data modeling would improve my model, which I hope to learn throughout this M.S. program.

From the results, I can see that Staten Island had the lowest number of shootings in 2020, and Brooklyn had the greatest number of shootings, with all my estimates being significant. However, based on the limitations mentioned above, this model and conclusion are open for revision and further research as I progress through this course and overall M.S. program.

Citations

1. NYC JSON, Data.Gov/data.cityofnewyork.us (2020). *NYPD Shooting Incident Data (Historic)*. Retrieved from https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic
2. Chen, S. (2018, September 14). The Bronx Is Great, Thonx. The New York Times. https://www.nytimes.com/2018/09/14/realestate/the-bronx-is-great-thonx.html

R Session Info

```
sessioninfo::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 3.5.1 (2018-07-02)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  tz       America/Denver
##  date     2021-05-20
##
## - Packages -------------------------------------------------------------------
##  package       * version date       source
##  assertthat      0.2.0   2017-04-11 CRAN (R 3.5.1)
##  backports       1.1.2   2017-12-13 CRAN (R 3.5.0)
##  bindr           0.1.1   2018-03-13 CRAN (R 3.5.1)
##  bindrcpp        0.2.2   2018-03-29 CRAN (R 3.5.1)
##  broom           0.5.0   2018-07-17 CRAN (R 3.5.1)
##  cellranger      1.1.0   2016-07-27 CRAN (R 3.5.1)
```

```
##  cli             1.0.0   2017-11-05 CRAN (R 3.5.1)
##  clisymbols      1.2.0   2017-05-21 CRAN (R 3.5.1)
##  colorspace      1.3-2   2016-12-14 CRAN (R 3.5.1)
##  cowplot       * 0.9.3   2018-07-15 CRAN (R 3.5.1)
##  crayon          1.3.4   2017-09-16 CRAN (R 3.5.1)
##  curl            3.2     2018-03-28 CRAN (R 3.5.0)
##  digest          0.6.15  2018-01-28 CRAN (R 3.5.1)
##  dplyr         * 0.7.6   2018-06-29 CRAN (R 3.5.1)
##  evaluate        0.11    2018-07-17 CRAN (R 3.5.1)
##  forcats       * 0.3.0   2018-02-19 CRAN (R 3.5.1)
##  ggplot2       * 3.0.0   2018-07-03 CRAN (R 3.5.1)
##  glue            1.3.0   2018-07-17 CRAN (R 3.5.1)
##  gtable          0.2.0   2016-02-26 CRAN (R 3.5.1)
##  haven           1.1.2   2018-06-27 CRAN (R 3.5.1)
##  hms             0.4.2   2018-03-10 CRAN (R 3.5.1)
##  htmltools       0.3.6   2017-04-28 CRAN (R 3.5.1)
##  httr            1.3.1   2017-08-20 CRAN (R 3.5.1)
##  jsonlite        1.5     2017-06-01 CRAN (R 3.5.0)
##  jtools        * 1.0.0   2018-05-08 CRAN (R 3.5.1)
##  knitr           1.20    2018-02-20 CRAN (R 3.5.1)
##  labeling        0.3     2014-08-23 CRAN (R 3.5.0)
##  lattice         0.20-35 2017-03-25 CRAN (R 3.5.1)
##  lazyeval        0.2.1   2017-10-29 CRAN (R 3.5.1)
##  lubridate     * 1.7.4   2018-04-11 CRAN (R 3.5.1)
##  magrittr        1.5     2014-11-22 CRAN (R 3.5.1)
##  modelr          0.1.2   2018-05-11 CRAN (R 3.5.1)
##  munsell         0.5.0   2018-06-12 CRAN (R 3.5.1)
##  nlme            3.1-137 2018-04-07 CRAN (R 3.5.1)
##  pillar          1.3.0   2018-07-14 CRAN (R 3.5.1)
##  pkgconfig       2.0.1   2017-03-21 CRAN (R 3.5.1)
##  plotfunctions * 1.3     2017-08-30 CRAN (R 3.5.1)
##  plyr            1.8.4   2016-06-08 CRAN (R 3.5.1)
##  purrr         * 0.2.5   2018-05-29 CRAN (R 3.5.1)
##  R6              2.2.2   2017-06-17 CRAN (R 3.5.0)
##  Rcpp            0.12.18 2018-07-23 CRAN (R 3.5.1)
##  readr         * 1.1.1   2017-05-16 CRAN (R 3.5.1)
##  readxl          1.1.0   2018-04-20 CRAN (R 3.5.1)
##  RevoUtils     * 11.0.1  2018-08-01 local
##  RevoUtilsMath * 11.0.0  2018-08-01 local
##  rlang           0.2.1   2018-05-30 CRAN (R 3.5.1)
##  rmarkdown       1.10    2018-06-11 CRAN (R 3.5.1)
##  rprojroot       1.3-2   2018-01-03 CRAN (R 3.5.1)
##  rstudioapi      0.7     2017-09-07 CRAN (R 3.5.1)
##  rvest           0.3.2   2016-06-17 CRAN (R 3.5.1)
##  scales          0.5.0   2017-08-24 CRAN (R 3.5.1)
##  sessioninfo     1.0.0   2017-06-21 CRAN (R 3.5.1)
##  stringi         1.1.7   2018-03-12 CRAN (R 3.5.0)
##  stringr       * 1.3.1   2018-05-10 CRAN (R 3.5.1)
##  tibble        * 1.4.2   2018-01-22 CRAN (R 3.5.1)
##  tidyr         * 0.8.1   2018-05-18 CRAN (R 3.5.1)
##  tidyselect      0.2.4   2018-02-26 CRAN (R 3.5.1)
##  tidyverse     * 1.2.1   2017-11-14 CRAN (R 3.5.1)
##  withr           2.1.2   2018-03-15 CRAN (R 3.5.1)
##  xml2            1.2.0   2018-01-24 CRAN (R 3.5.1)
```

```
##  yaml            2.2.0   2018-07-25 CRAN (R 3.5.1)
```