NYPD Shooting Incident Data Report

Data Science as a Field - Week 3 Assignment

For this assignment we are exploring historical shooting incidents that occurred in New York City from 2006 - 2020. The data comes from data.gov website and was collected by the New York City police department.

We will explore the data to determine dangerous boroughs in New York City and also see if we are able to model the number of gun related deaths.

Datasets:

1) Historical shooting incidents NYC (2006 - 2020); source: data.gov
2) Population data for each NYC borough; source: Census Data - Google

Data Ingestion

First let's import the library we will be using:

```
library(dplyr)
library(tidyverse)
library(tibble)
library(lubridate)
library(ggplot2)
library(scales)
library(patchwork)
library(formatR)
library(prophet)
```

Now let's grab our data from the gov website:

```
# URL with our data
nypd_url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
# Reading in our data, and specifying that strings become
# factors
shooting_data <- read.csv(nypd_url, stringsAsFactors = TRUE)
# Converting to a tibble
shooting_data <- as_tibble(shooting_data)
```

Data Exploration

Now let's view the data:

```r
# Looking at our data
head(shooting_data)
```

```
## # A tibble: 6 x 19
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     PRECINCT JURISDICTION_CODE
##          <int> <fct>      <fct>      <fct>       <int>             <int>
## 1     24050482 08/27/2006 05:35:00   BRONX          52                 0
## 2     77673979 03/11/2011 12:03:00   QUEENS        106                 0
## 3    203350417 10/06/2019 01:09:00   BROOKLYN       77                 0
## 4     80584527 09/04/2011 03:35:00   BRONX          40                 0
## 5     90843766 05/27/2013 21:16:00   QUEENS        100                 0
## 6     92393427 09/01/2013 04:17:00   BROOKLYN       67                 0
## # ... with 13 more variables: LOCATION_DESC <fct>,
## #   STATISTICAL_MURDER_FLAG <fct>, PERP_AGE_GROUP <fct>, PERP_SEX <fct>,
## #   PERP_RACE <fct>, VIC_AGE_GROUP <fct>, VIC_SEX <fct>, VIC_RACE <fct>,
## #   X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>, Longitude <dbl>,
## #   Lon_Lat <fct>
```

```r
# Taking a look at our columns / data types
str(shooting_data)
```

```
## tibble [23,585 x 19] (S3: tbl_df/tbl/data.frame)
##  $ INCIDENT_KEY           : int [1:23585] 24050482 77673979 203350417 80584527 90843766 92393427 7305
##  $ OCCUR_DATE             : Factor w/ 5054 levels "01/01/2006","01/01/2007",..: 3290 906 3879 3412 19
##  $ OCCUR_TIME             : Factor w/ 1401 levels "00:00:00","00:01:00",..: 336 685 70 216 1238 258 1
##  $ BORO                   : Factor w/ 5 levels "BRONX","BROOKLYN",..: 1 4 2 1 4 2 2 2 4 4 ...
##  $ PRECINCT               : int [1:23585] 52 106 77 40 100 67 77 81 101 106 ...
##  $ JURISDICTION_CODE      : int [1:23585] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LOCATION_DESC          : Factor w/ 40 levels "","ATM","BANK",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ STATISTICAL_MURDER_FLAG: Factor w/ 2 levels "false","true": 2 1 1 1 1 1 1 1 1 1 ...
##  $ PERP_AGE_GROUP         : Factor w/ 10 levels "","<18","1020",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PERP_SEX               : Factor w/ 4 levels "","F","M","U": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PERP_RACE              : Factor w/ 8 levels "","AMERICAN INDIAN/ALASKAN NATIVE",..: 1 1 1 1 1 1 1
##  $ VIC_AGE_GROUP          : Factor w/ 6 levels "<18","18-24",..: 3 5 2 1 2 1 1 3 2 2 ...
##  $ VIC_SEX                : Factor w/ 3 levels "F","M","U": 1 2 1 2 2 2 2 2 2 2 ...
##  $ VIC_RACE               : Factor w/ 7 levels "AMERICAN INDIAN/ALASKAN NATIVE",..: 4 6 3 3 3 3 3 3 3
##  $ X_COORD_CD             : num [1:23585] 1017542 1027543 995325 1007453 1041267 ...
##  $ Y_COORD_CD             : num [1:23585] 255919 186095 185155 233952 157134 ...
##  $ Latitude               : num [1:23585] 40.9 40.7 40.7 40.8 40.6 ...
##  $ Longitude              : num [1:23585] -73.9 -73.8 -74 -73.9 -73.8 ...
##  $ Lon_Lat                : Factor w/ 10055 levels "POINT (-73.70204616699993 40.74174860900007)",..
```

We can see that each shooting incident is recorded per row. We also can see that some columns have the wrong data type Like date which is coded as a factor type. We will fix this issue a little further down.

For now, let's take a look at the summary of our data

```r
summary(shooting_data)
```

```
##   INCIDENT_KEY            OCCUR_DATE        OCCUR_TIME                BORO
##  Min.   :  9953245   07/05/2020:   47   23:30:00:  159   BRONX        :6701
##  1st Qu.: 55322804   09/04/2011:   31   01:30:00:  141   BROOKLYN     :9734
```

```
##  Median : 83435362   07/26/2020:   29   00:30:00:  136   MANHATTAN    :2922
##  Mean   :102280741   08/11/2007:   26   02:00:00:  129   QUEENS       :3532
##  3rd Qu.:150911774   09/04/2006:   25   21:00:00:  128   STATEN ISLAND: 696
##  Max.   :230611229   08/15/2020:   24   22:30:00:  126
##                      (Other)   :23403   (Other) :22766
##     PRECINCT      JURISDICTION_CODE                    LOCATION_DESC
##  Min.   :  1.00   Min.   :0.000                               :13581
##  1st Qu.: 44.00   1st Qu.:0.000     MULTI DWELL - PUBLIC HOUS: 4240
##  Median : 69.00   Median :0.000     MULTI DWELL - APT BUILD  : 2553
##  Mean   : 66.21   Mean   :0.333     PVT HOUSE                :  857
##  3rd Qu.: 81.00   3rd Qu.:0.000     GROCERY/BODEGA           :  574
##  Max.   :123.00   Max.   :2.000     BAR/NIGHT CLUB           :  562
##                   NA's   :2         (Other)                  : 1218
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX          PERP_RACE
##  false:19085                    :8295    : 8261   BLACK         :10025
##  true : 4500             18-24  :5508   F:  335                 : 8261
##                          25-44  :4714   M:13490   WHITE HISPANIC: 1988
##                          UNKNOWN:3148   U: 1499   UNKNOWN       : 1836
##                          <18    :1368             BLACK HISPANIC: 1096
##                          45-64  : 495             WHITE         :  255
##                          (Other):  57             (Other)       :  124
##  VIC_AGE_GROUP   VIC_SEX                            VIC_RACE
##  <18    : 2525   F: 2204   AMERICAN INDIAN/ALASKAN NATIVE:    9
##  18-24  : 9003   M:21370   ASIAN / PACIFIC ISLANDER      :  327
##  25-44  :10303   U:   11   BLACK                         :16869
##  45-64  : 1541             BLACK HISPANIC                : 2245
##  65+    :  154             UNKNOWN                       :   65
##  UNKNOWN:   59             WHITE                         :  620
##                            WHITE HISPANIC                : 3450
##    X_COORD_CD        Y_COORD_CD        Latitude       Longitude
##  Min.   : 914928   Min.   :125757   Min.   :40.51   Min.   :-74.25
##  1st Qu.: 999925   1st Qu.:182539   1st Qu.:40.67   1st Qu.:-73.94
##  Median :1007654   Median :193470   Median :40.70   Median :-73.92
##  Mean   :1009379   Mean   :207300   Mean   :40.74   Mean   :-73.91
##  3rd Qu.:1016782   3rd Qu.:239163   3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :1066815   Max.   :271128   Max.   :40.91   Max.   :-73.70
##
##                                         Lon_Lat
##  POINT (-73.88151014499994 40.67141260500006) :   66
##  POINT (-73.84760778699996 40.88745131300004) :   47
##  POINT (-73.91339091999998 40.670655072000045):   47
##  POINT (-73.88143295699996 40.67110691100004) :   44
##  POINT (-74.17125343299995 40.63898537500006) :   44
##  POINT (-73.91983075699994 40.83732351100008) :   42
##  (Other)                                      :23295
```

Now we have a good idea of what the data represents, let's start to clean our dataset.

Data Cleaning

First, let's see if there is any missing data:

```r
missing_data <- shooting_data %>%
    mutate_all(na_if, "")
data.frame(sapply(missing_data, function(x) sum(is.na(x))))
```

```
##                          sapply.missing_data..function.x..sum.is.na.x...
## INCIDENT_KEY                                                           0
## OCCUR_DATE                                                             0
## OCCUR_TIME                                                             0
## BORO                                                                   0
## PRECINCT                                                               0
## JURISDICTION_CODE                                                      2
## LOCATION_DESC                                                      13581
## STATISTICAL_MURDER_FLAG                                                0
## PERP_AGE_GROUP                                                      8295
## PERP_SEX                                                            8261
## PERP_RACE                                                           8261
## VIC_AGE_GROUP                                                          0
## VIC_SEX                                                                0
## VIC_RACE                                                               0
## X_COORD_CD                                                             0
## Y_COORD_CD                                                             0
## Latitude                                                               0
## Longitude                                                              0
## Lon_Lat                                                                0
```

We can see that some columns are missing a lot of data. let's clean up our data, by removing columns that have a lot of missing data. We also have a lot of redundant columns (like Lat and longitude) that won't be used in our analysis so we will remove them:

```r
columns_with_missing_data <- c("LOCATION_DESC", "PERP_AGE_GROUP",
    "PERP_SEX", "PERP_RACE")

unnecessary_columns <- c("INCIDENT_KEY", "PRECINCT", "VIC_AGE_GROUP",
    "VIC_SEX", "VIC_RACE", "X_COORD_CD", "Y_COORD_CD", "Latitude",
    "Longitude", "Lon_Lat", "JURISDICTION_CODE")

# Removing those columns
shooting_data_clean <- shooting_data %>%
    select(-c(all_of(columns_with_missing_data), all_of(unnecessary_columns)))
```

Now we will clean our date and time columns:

```r
shooting_data_clean <- shooting_data_clean %>%
    mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
    mutate(OCCUR_TIME = hms(OCCUR_TIME))
```

We will add year and month columns:

```r
shooting_data_clean <- shooting_data_clean %>%
    mutate(YEAR = year(OCCUR_DATE)) %>%
    mutate(MONTH = month(OCCUR_DATE))
```

And finally we will add a count of our shooting incidents, and transform the murder flag to binary values (0, 1). This flag represents the number of shooting related deaths.

```
shooting_data_clean$INCIDENTS <- 1
# Transforming flag to binary
shooting_data_clean$STATISTICAL_MURDER_FLAG <- as.integer(as.logical(shooting_data_clean$STATISTICAL_MU
```

Now let's take a look at our data now after cleaning:

```
head(shooting_data_clean)
```

```
## # A tibble: 6 x 7
##   OCCUR_DATE OCCUR_TIME BORO     STATISTICAL_MURDER_FLAG  YEAR MONTH INCIDENTS
##   <date>     <Period>   <fct>                      <int> <dbl> <dbl>     <dbl>
## 1 2006-08-27 5H 35M 0S  BRONX                          1  2006     8         1
## 2 2011-03-11 12H 3M 0S  QUEENS                         0  2011     3         1
## 3 2019-10-06 1H 9M 0S   BROOKLYN                       0  2019    10         1
## 4 2011-09-04 3H 35M 0S  BRONX                          0  2011     9         1
## 5 2013-05-27 21H 16M 0S QUEENS                         0  2013     5         1
## 6 2013-09-01 4H 17M 0S  BROOKLYN                       0  2013     9         1
```

Data Visualization and Analysis

It looks like we have a good clean dataset to work with. Now let's do some data exploration. Let's create a new dataframe where we group the shooting incidents by year:

```
shooting_data_year <- shooting_data_clean %>%
    group_by(YEAR) %>%
    summarize(INCIDENTS = sum(INCIDENTS), DEATHS = sum(STATISTICAL_MURDER_FLAG))
shooting_data_year
```

```
## # A tibble: 15 x 3
##     YEAR INCIDENTS DEATHS
##    <dbl>     <dbl>  <int>
##  1  2006      2055    445
##  2  2007      1887    373
##  3  2008      1959    362
##  4  2009      1828    348
##  5  2010      1912    405
##  6  2011      1939    373
##  7  2012      1717    288
##  8  2013      1339    223
##  9  2014      1464    249
## 10  2015      1434    283
## 11  2016      1208    223
## 12  2017       970    174
## 13  2018       958    204
## 14  2019       967    184
## 15  2020      1948    366
```

Now that we have this dataset, let's graph the number of incidents over the years:

```r
ggplot(shooting_data_year, aes(x = YEAR, y = INCIDENTS)) + geom_line(size = 1) +
    geom_point(size = 3) + ggtitle("NYC Shooting Incidents by Year (2006-2020)")
```

It looks like the number of shooting incidents were decreasing year over year, until we hit 2020. Before starting the analysis I thought 2020 would have much fewer incidents because of COVID (see bias section further down for more elaboration).

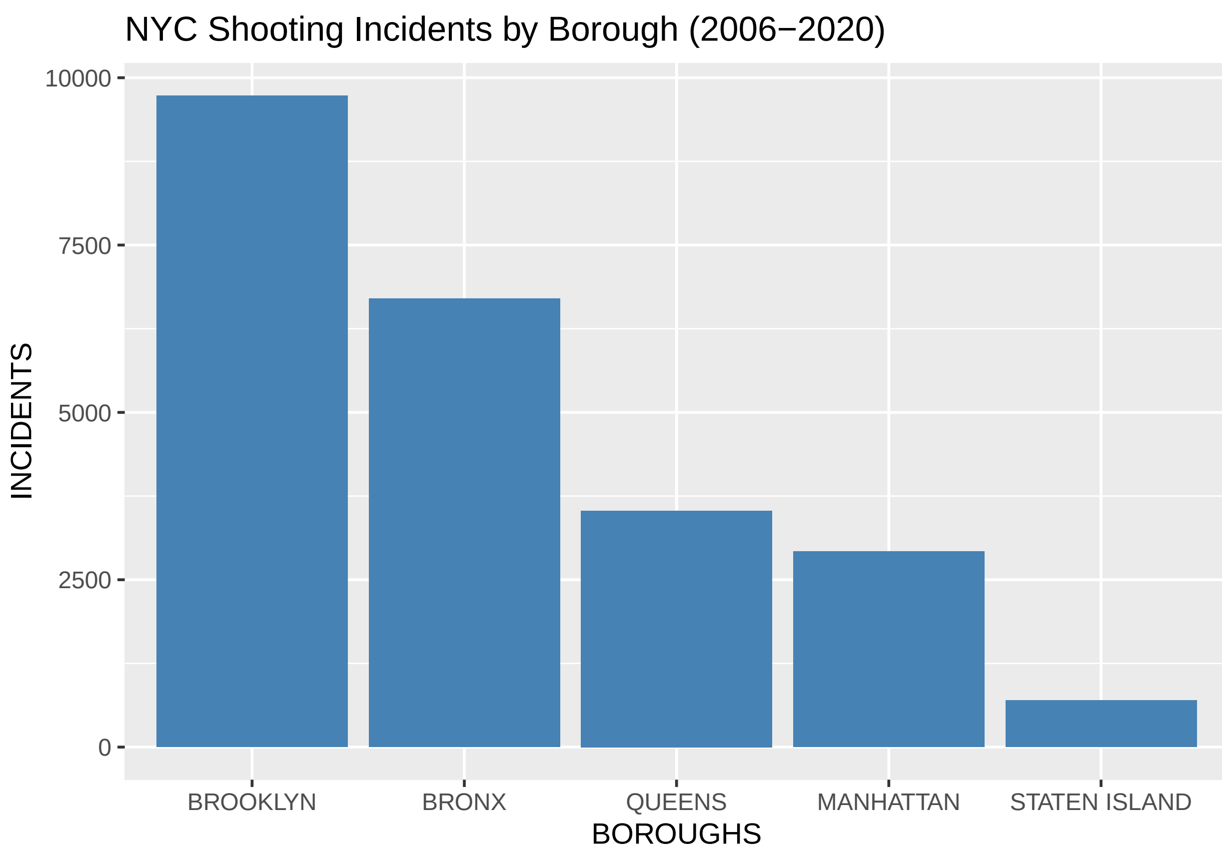
Now that we looked at the incidents over the years, let's create a new data set to see the incidents by NYC borough:

```r
# Group by borough
shooting_data_borough <- shooting_data_clean %>%
    group_by(BORO) %>%
    summarize(INCIDENTS = sum(INCIDENTS), DEATHS = sum(STATISTICAL_MURDER_FLAG))
shooting_data_borough
```

```
## # A tibble: 5 x 3
##   BORO          INCIDENTS DEATHS
##   <fct>             <dbl>  <int>
## 1 BRONX              6701   1247
## 2 BROOKLYN           9734   1898
## 3 MANHATTAN          2922    515
## 4 QUEENS             3532    697
## 5 STATEN ISLAND       696    143
```

It looks like some boroughs have more incidents than others, let's take a look:

```r
ggplot(shooting_data_borough, aes(x = reorder(BORO, -INCIDENTS),
    y = INCIDENTS)) + geom_bar(stat = "identity", fill = "steelblue") +
    xlab("BOROUGHS") + ggtitle("NYC Shooting Incidents by Borough (2006-2020)")
```



It looks like Brooklyn has the most incidents... But does having the most incident mean that it the most "gun violent" borough in NYC?

I decided to look at the population data of each of the boroughs (information taken from Google). Let's take a look:

```r
# Got population of each borough from Google
population_boroughs <- data.frame(BORO = c("QUEENS", "MANHATTAN",
    "BRONX", "BROOKLYN", "STATEN ISLAND"), POPULATION = c(2287000,
    1632000, 1435000, 2590000, 474893))
population_boroughs
```

```
##            BORO POPULATION
## 1        QUEENS    2287000
## 2     MANHATTAN    1632000
## 3         BRONX    1435000
## 4      BROOKLYN    2590000
## 5 STATEN ISLAND     474893
```

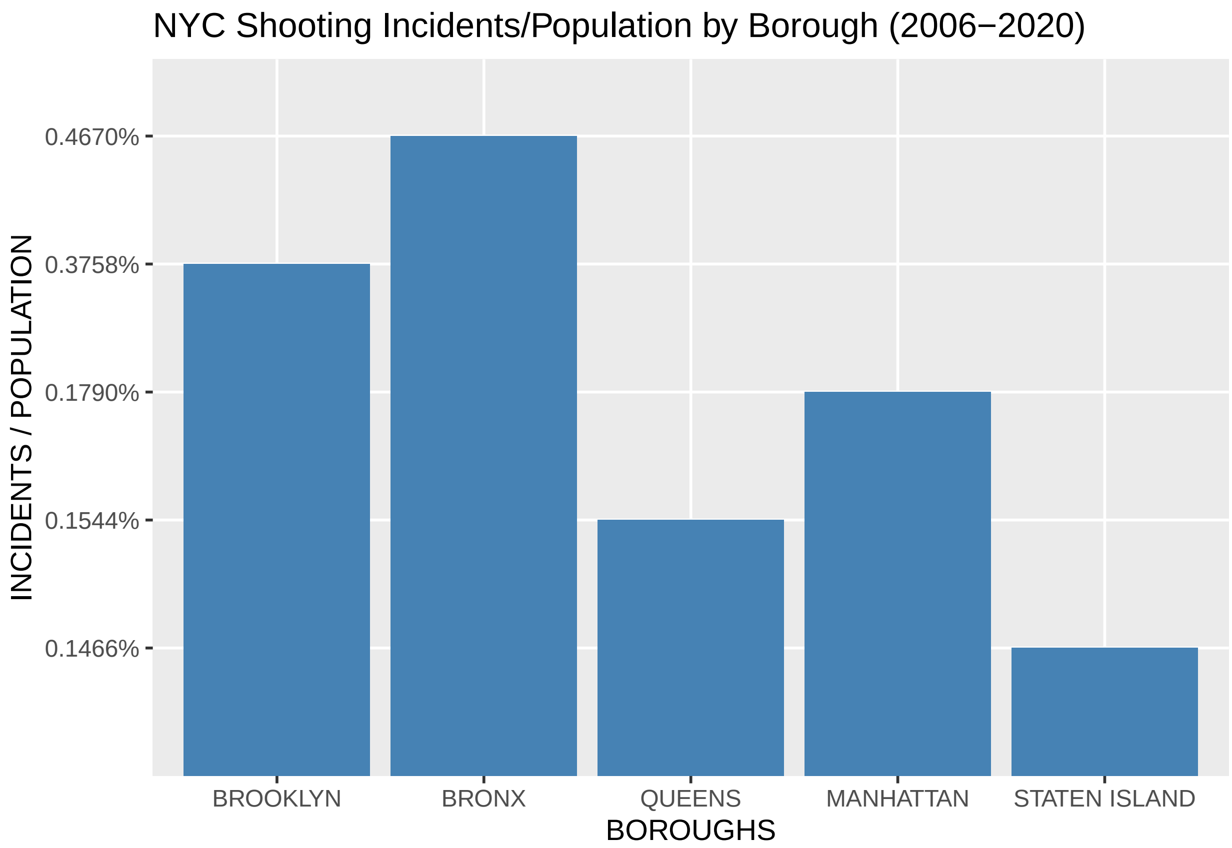
So it seems that Brooklyn has the highest population. So let's take a look at the number of incidents by population. First let's merge our new dataset with our boroughs data, and then divide the number of incidents by population:

```r
# Merging our datasets together into boroughs_comnplete.
boroughs_complete <- merge(x = shooting_data_borough, y = population_boroughs,
    by = "BORO")
boroughs_complete$INCIDENTS_VS_POPULATION <- percent(boroughs_complete$INCIDENTS/boroughs_complete$POPUL
boroughs_complete
```

```
##            BORO INCIDENTS DEATHS POPULATION INCIDENTS_VS_POPULATION
## 1         BRONX      6701   1247    1435000                 0.4670%
## 2      BROOKLYN      9734   1898    2590000                 0.3758%
## 3     MANHATTAN      2922    515    1632000                 0.1790%
## 4        QUEENS      3532    697    2287000                 0.1544%
## 5 STATEN ISLAND       696    143     474893                 0.1466%
```

So it looks like the Bronx has the most number of incidents by population. Let's visualize the data:

```r
ggplot(boroughs_complete, aes(x = reorder(BORO, -INCIDENTS),
    y = INCIDENTS_VS_POPULATION)) + geom_bar(stat = "identity",
    fill = "steelblue") + xlab("BOROUGHS") + ylab("INCIDENTS / POPULATION") +
    ggtitle("NYC Shooting Incidents/Population by Borough (2006-2020)")
```



So while Brooklyn has the most number of incidents, it seems that the Bronx has more gun incidents / population and is more dangerous.

Data Modeling

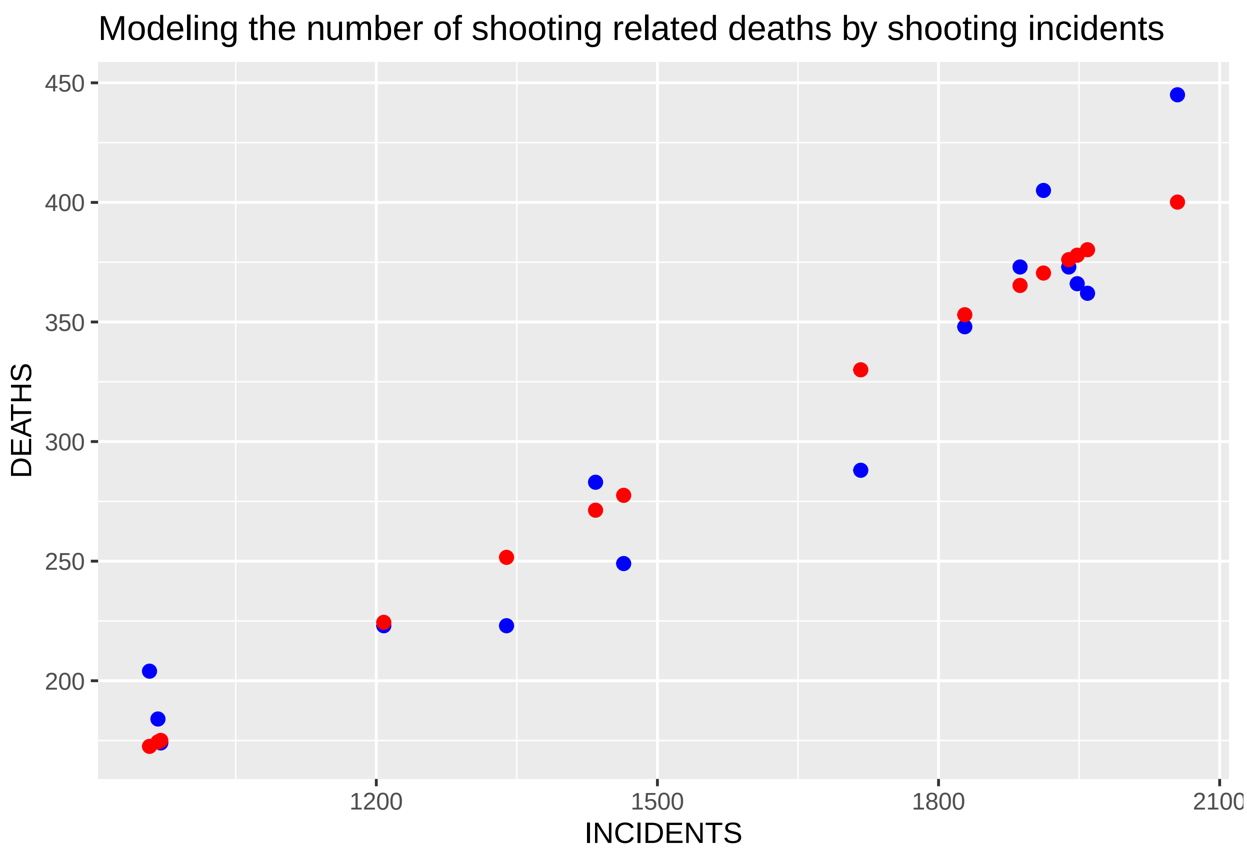
Previously our dataset had the number of shooting incidents and deaths. Let's see if we can use the # of shooting incidents to predict how many deaths we will have in NYC per year. We will use a linear model:

```
model <- lm(DEATHS ~ INCIDENTS, data = shooting_data_year)
shooting_data_year_final <- shooting_data_year %>%
    mutate(PRED = predict(model))
shooting_data_year_final
```

```
## # A tibble: 15 x 4
##     YEAR INCIDENTS DEATHS  PRED
##    <dbl>     <dbl>  <int> <dbl>
##  1  2006      2055    445  400.
##  2  2007      1887    373  365.
##  3  2008      1959    362  380.
##  4  2009      1828    348  353.
##  5  2010      1912    405  370.
##  6  2011      1939    373  376.
##  7  2012      1717    288  330.
##  8  2013      1339    223  252.
##  9  2014      1464    249  278.
## 10  2015      1434    283  271.
## 11  2016      1208    223  224.
## 12  2017       970    174  175.
## 13  2018       958    204  173.
## 14  2019       967    184  174.
## 15  2020      1948    366  378.
```

We have a model that predicts the number of deaths, let's visualize the predictions vs the actual death related count

```
shooting_data_year_final %>%
    ggplot() + geom_point(aes(x = INCIDENTS, y = DEATHS, colour = "DEATHS"),
    color = "blue", size = 2, show.legend = TRUE) + geom_point(aes(x = INCIDENTS,
    y = PRED, colour = "PRED"), color = "red", size = 2, show.legend = TRUE) +
    ggtitle("Modeling the number of shooting related deaths by shooting incidents")
```



It looks like we are able to create a fairly accurate model to represent the number of gun related deaths using the # of gun shooting incidents.

Let's also try a special Forecasting Library called Prophet that will allow us to predict the number of future gun related incidents in NYC. We first have to manipulate the data so that Prophet is able to read it.

```
# making prophet fit
shooting_data_year_prophet <- shooting_data_year_final %>%
    select(c(YEAR, INCIDENTS))
shooting_data_year_prophet$ds <- as.Date(paste(as.character(shooting_data_year_prophet$YEAR),
    "01", "01", sep = "-"))
shooting_data_year_prophet <- shooting_data_year_prophet %>%
    select(c(ds, INCIDENTS)) %>%
    rename(y = "INCIDENTS")

shooting_data_year_prophet
```

```
## # A tibble: 15 x 2
##    ds             y
##    <date>     <dbl>
##  1 2006-01-01  2055
##  2 2007-01-01  1887
##  3 2008-01-01  1959
##  4 2009-01-01  1828
##  5 2010-01-01  1912
##  6 2011-01-01  1939
```

```
##   7 2012-01-01  1717
##   8 2013-01-01  1339
##   9 2014-01-01  1464
##  10 2015-01-01  1434
##  11 2016-01-01  1208
##  12 2017-01-01   970
##  13 2018-01-01   958
##  14 2019-01-01   967
##  15 2020-01-01  1948
```

Now that we have it in the format that Prophet requires, let's format use it to forecast out 10 years into the future to see what will happen with the # of shooting incidents.

```
# making prophet fit
m <- prophet(shooting_data_year_prophet)
```

```
## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to override this.
```

```
## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
```

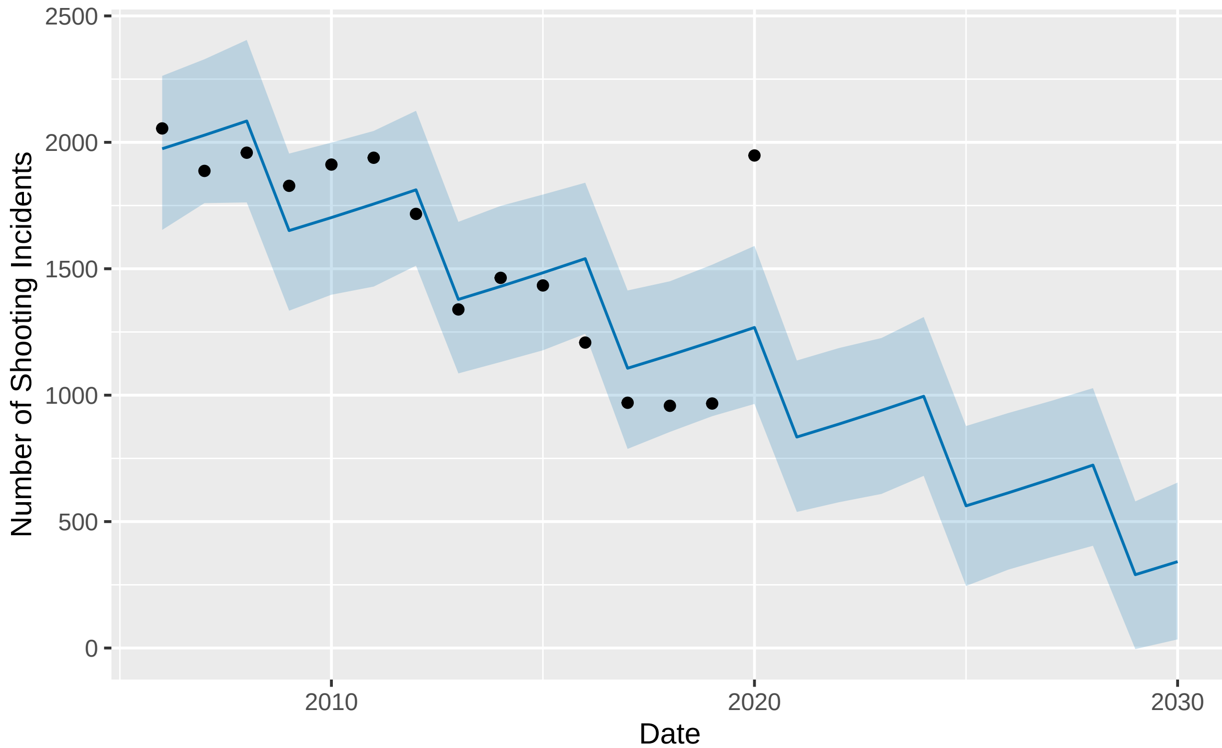
```
## n.changepoints greater than number of observations. Using 11
```

```
future <- make_future_dataframe(m, periods = 10, freq = "year")

# forecasting out the future
forecast <- predict(m, future)
tail(forecast[c("ds", "yhat", "yhat_lower", "yhat_upper")])
```

```
##            ds     yhat yhat_lower yhat_upper
## 20 2025-01-01 562.2234 245.287458   877.9162
## 21 2026-01-01 613.7614 310.104911   929.4536
## 22 2027-01-01 667.5319 358.392376   976.6673
## 23 2028-01-01 723.3876 404.351491  1027.8263
## 24 2029-01-01 290.0544  -4.208589   579.9408
## 25 2030-01-01 341.5924  34.027242   654.5122
```

```
plot(m, forecast, main = "Main titles", xlab = "Date", ylab = "Number of Shooting Incidents",
    sub = "Sub-title")
```



The black dots are the actual values and the blue line is the prediction. The lower and upper confidence bounds are given by the shaded blue region. It looks like the model took previous years data (before 2020) and continues the pre-COVID trend.

Bias Identification

Personal Bias: As previously mentioned, one of my personal biases was the belief that there would be less shooting related incidents during 2020 because of Covid. I assumed that people would be locked down at home. From examining the data, that assumption was proven to be false. Covid and potentially other factors (2020 was also a hotly contested election year) actually increased the number of shooting incidents.

Information Bias: Another bias that I noticed would potentially be the time the shooting incident was recorded. Since these observations are human dependent, I wanted to remove it entirely from my dataset. I also wanted to remove any factors involving race. There have been several algorithms which ended up being inherently racist. See this MIT article describing police specific algorithms: https://www.technologyreview.com/2020/07/17/1005396/predictive-policing-algorithms-racist-dismantled-machine-learning-bias-criminal-justice/

Conclusion

We cleaned, examined, visualized, analyzed and modeled the data from the NYPD. It looks like the number of shooting incidents has been steadily decreasing year of year, until we hit 2020. Covid (and other factors) seem to have had a negative impact and has lead to an increase in the number of shooting incidents. The Bronx seems to be the most violent borough in NYC with the highest number of shooting incidents /

population. Finally, we showed that we are able to model the number of shooting related deaths by using the # of shooting incidents.

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] prophet_1.0     rlang_0.4.12    Rcpp_1.0.7      formatR_1.11
##  [5] patchwork_1.1.1 scales_1.1.1    lubridate_1.8.0 forcats_0.5.1
##  [9] stringr_1.4.0   purrr_0.3.4     readr_2.0.2     tidyr_1.1.4
## [13] tibble_3.1.5    ggplot2_3.3.5   tidyverse_1.3.1 dplyr_1.0.7
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2          jsonlite_1.7.2       modelr_0.1.8
##  [4] StanHeaders_2.21.0-7 RcppParallel_5.1.4   assertthat_0.2.1
##  [7] highr_0.9           stats4_4.1.2         cellranger_1.1.0
## [10] yaml_2.2.1          pillar_1.6.4         backports_1.3.0
## [13] glue_1.4.2          digest_0.6.28        rvest_1.0.2
## [16] colorspace_2.0-2    htmltools_0.5.2      pkgconfig_2.0.3
## [19] rstan_2.21.2        broom_0.7.10         haven_2.4.3
## [22] processx_3.5.2      tzdb_0.2.0           generics_0.1.1
## [25] farver_2.1.0        ellipsis_0.3.2       withr_2.4.2
## [28] cli_3.1.0           magrittr_2.0.1       crayon_1.4.2
## [31] readxl_1.3.1        evaluate_0.14        ps_1.6.0
## [34] fs_1.5.0            fansi_0.5.0          xml2_1.3.2
## [37] pkgbuild_1.2.1      loo_2.4.1            tools_4.1.2
## [40] prettyunits_1.1.1   hms_1.1.1            matrixStats_0.61.0
## [43] lifecycle_1.0.1     extraDistr_1.9.1     V8_3.6.0
## [46] munsell_0.5.0       reprex_2.0.1         callr_3.7.0
## [49] compiler_4.1.2      grid_4.1.2           rstudioapi_0.13
## [52] labeling_0.4.2      rmarkdown_2.11       gtable_0.3.0
## [55] codetools_0.2-18    inline_0.3.19        DBI_1.1.1
## [58] curl_4.3.2          R6_2.5.1             gridExtra_2.3
## [61] knitr_1.36          fastmap_1.1.0        utf8_1.2.2
## [64] stringi_1.7.5       parallel_4.1.2       vctrs_0.3.8
## [67] dbplyr_2.1.1        tidyselect_1.1.1     xfun_0.27
```