NYPD Shooting Incident Data Report

Aaron Li

22/1/2022

The data source

We take the historic dataset that includes a list of every shooting incident occurred in NYC going back to 2006 through the end of the previous calendar year. You may find the same CSV file used in this report as below :

https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8

Lets load up the raw data first :

```
library(tidyverse)
library(lubridate)

url_in <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv"
NYPD_shootings <- read_csv(url_in)
NYPD_shootings
```

```
## # A tibble: 23,585 x 19
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     PRECINCT JURISDICTION_CODE
##           <dbl> <chr>      <time>     <chr>       <dbl>              <dbl>
##  1     24050482 08/27/2006 05:35      BRONX          52                  0
##  2     77673979 03/11/2011 12:03      QUEENS        106                  0
##  3    203350417 10/06/2019 01:09      BROOKLYN       77                  0
##  4     80584527 09/04/2011 03:35      BRONX          40                  0
##  5     90843766 05/27/2013 21:16      QUEENS        100                  0
##  6     92393427 09/01/2013 04:17      BROOKLYN       67                  0
##  7     73057167 06/05/2010 21:16      BROOKLYN       77                  0
##  8    211362213 03/20/2020 21:27      BROOKLYN       81                  0
##  9    137564752 07/04/2014 00:25      QUEENS        101                  0
## 10    147024011 10/18/2015 01:33      QUEENS        106                  0
## # ... with 23,575 more rows, and 13 more variables: LOCATION_DESC <chr>,
## #   STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>, PERP_SEX <chr>,
## #   PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>, VIC_RACE <chr>,
## #   X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>, Longitude <dbl>,
## #   Lon_Lat <chr>
```

Analysis 1 - Perpetrator race distribution

As we see from the data structure, **PERP_RACE** represents the race of perpetrator in each shooting incidence. We would like to do a quick analysis on how it is distributed among various races and plot a pie chart on it.

Cleaning and counting

We will do some cleaning to filter out the incidences that do not record the perpetrator race and based on what's left to arrive at the counts per race in the whole dataset:

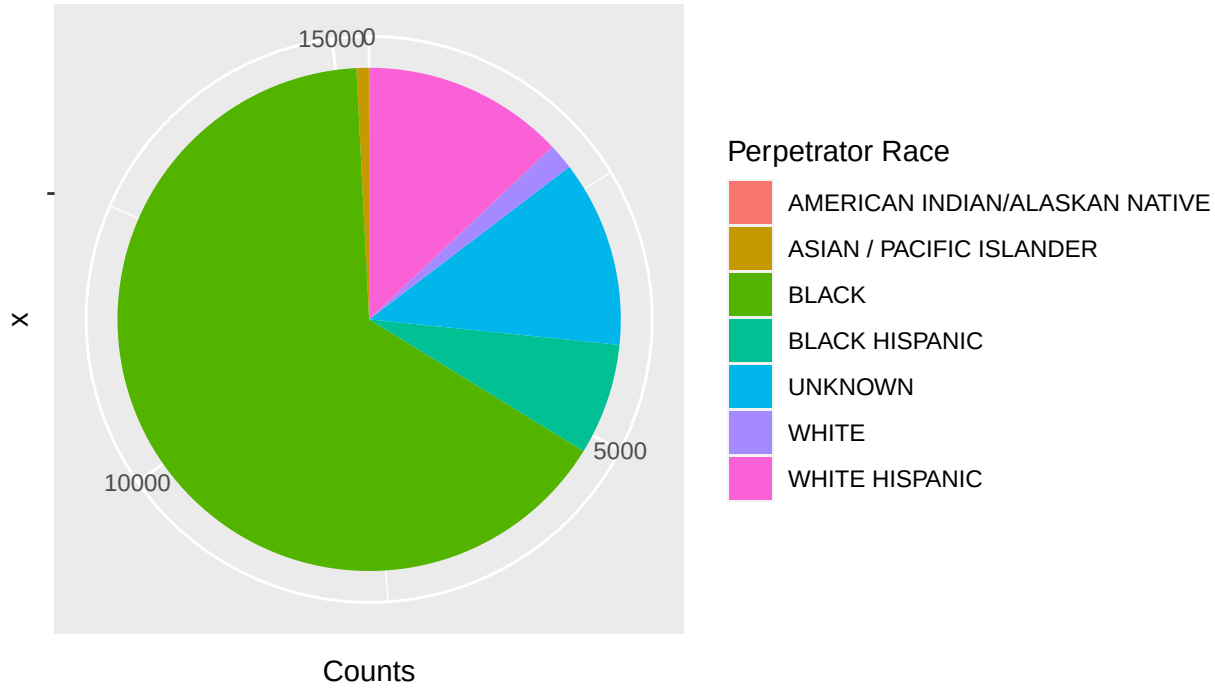
```
NYPD_shootings_perp_races <- NYPD_shootings %>%
    filter(!is.na(PERP_RACE)) %>%
    select(PERP_RACE) %>%
    count(PERP_RACE) %>%
    rename(`Perpetrator Race` = PERP_RACE, Counts = n)
NYPD_shootings_perp_races
```

```
## # A tibble: 7 x 2
##   `Perpetrator Race`             Counts
##   <chr>                           <int>
## 1 AMERICAN INDIAN/ALASKAN NATIVE      2
## 2 ASIAN / PACIFIC ISLANDER          122
## 3 BLACK                           10025
## 4 BLACK HISPANIC                   1096
## 5 UNKNOWN                          1836
## 6 WHITE                             255
## 7 WHITE HISPANIC                   1988
```

Plotting

To have a clearer view on how perpetrator races are distributed, we will plot the count data as a pie chart:

```
ggplot(NYPD_shootings_perp_races, aes(x="", y=Counts, fill=`Perpetrator Race`)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0)
```



Analysis 2 - Shooting deaths by year

STATISTICAL_MURDER_FLAG being TRUE indicates victim was shot dead in that particular incidence. We would like to count the shot death and number of the incidences year by year and find the trend and later on will work on a model to find out if these two values are somehow correlated.

Cleaning and counting

STATISTICAL_MURDER_FLAG and **OCCUR_DATE** are crucial to be retained while we drop off the rest of the unnecessary columns. We would also need to extract years out of **OCCUR_DATE** for grouping purpose later:

```
NYPD_shooting_deaths <- NYPD_shootings %>%
    select(c(INCIDENT_KEY,OCCUR_DATE,STATISTICAL_MURDER_FLAG)) %>%
    mutate(month = month(mdy(OCCUR_DATE)), year = year(mdy(OCCUR_DATE)))
NYPD_shooting_deaths
```

```
## # A tibble: 23,585 x 5
##    INCIDENT_KEY OCCUR_DATE STATISTICAL_MURDER_FLAG month  year
##           <dbl> <chr>      <lgl>                   <dbl> <dbl>
##  1     24050482 08/27/2006 TRUE                        8  2006
##  2     77673979 03/11/2011 FALSE                       3  2011
##  3    203350417 10/06/2019 FALSE                      10  2019
##  4     80584527 09/04/2011 FALSE                       9  2011
```

```
##  5     90843766 05/27/2013 FALSE                      5  2013
##  6     92393427 09/01/2013 FALSE                      9  2013
##  7     73057167 06/05/2010 FALSE                      6  2010
##  8    211362213 03/20/2020 FALSE                      3  2020
##  9    137564752 07/04/2014 FALSE                      7  2014
## 10    147024011 10/18/2015 FALSE                     10  2015
## # ... with 23,575 more rows
```

Now we count number of deaths and incidences by year:

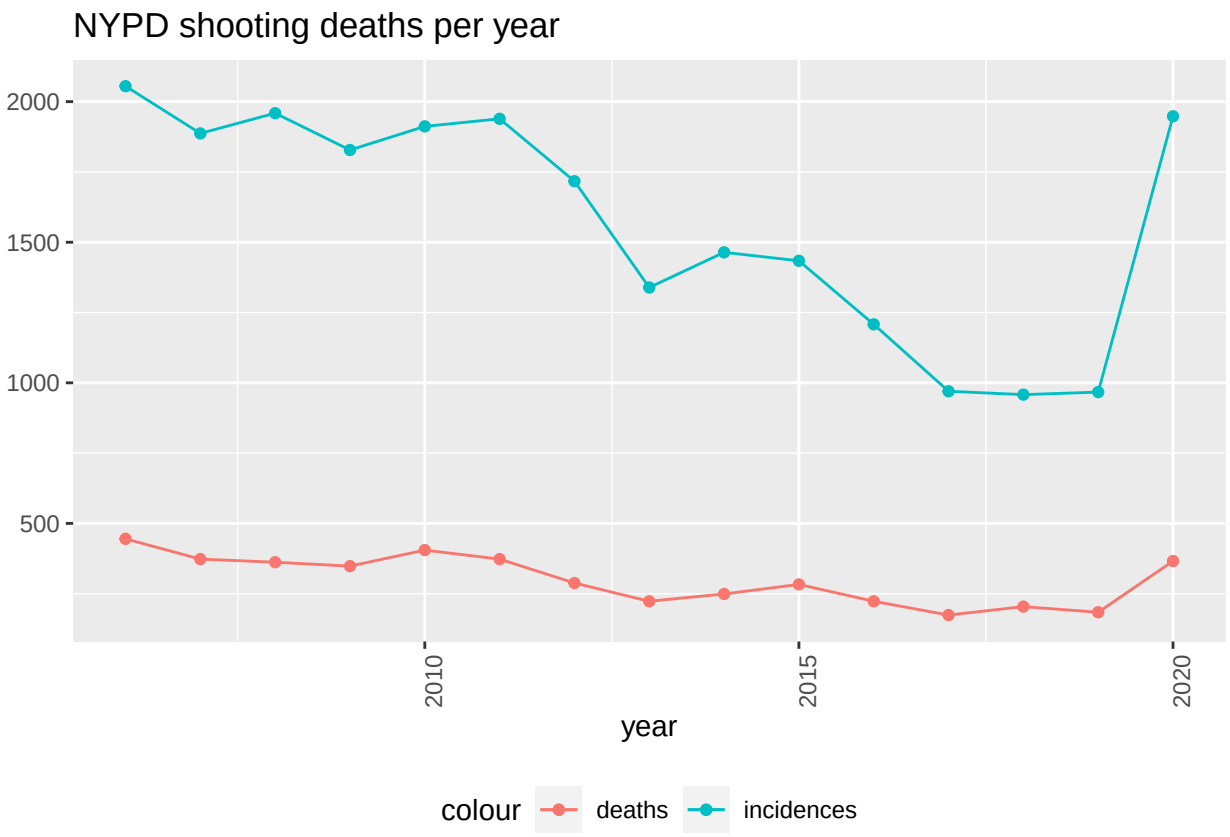
```
NYPD_shooting_deaths_per_yr <- NYPD_shooting_deaths %>%
    group_by(year) %>%
    summarize(incidences = n(), deaths = sum(STATISTICAL_MURDER_FLAG == TRUE))
NYPD_shooting_deaths_per_yr
```

```
## # A tibble: 15 x 3
##     year incidences deaths
##    <dbl>      <int>  <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
```

Plotting

The plot reflects the change of shooting incidences and deaths over years: Now we count number of deaths and incidences by year:

```
NYPD_shooting_deaths_per_yr %>%
     ggplot(aes(x = year, y = incidences)) +
     geom_line(aes(color = "incidences"))  +
     geom_point(aes(color = "incidences")) +
     geom_line(aes(y = deaths, color = "deaths")) +
     geom_point(aes(y= deaths, color = "deaths")) +
     theme(legend.position="bottom",
           axis.text.x = element_text(angle = 90)) +
     labs(title = "NYPD shooting deaths per year", y= NULL)
```



We also find number of incidences and deaths to some extent related.

Modelling and plotting the predicition

Deaths per year is found changing along with incidences, so we assume it is linear to incidences. A model can be built as below:

```
mod <- lm(deaths ~ incidences, data = NYPD_shooting_deaths_per_yr)
summary(mod)
```

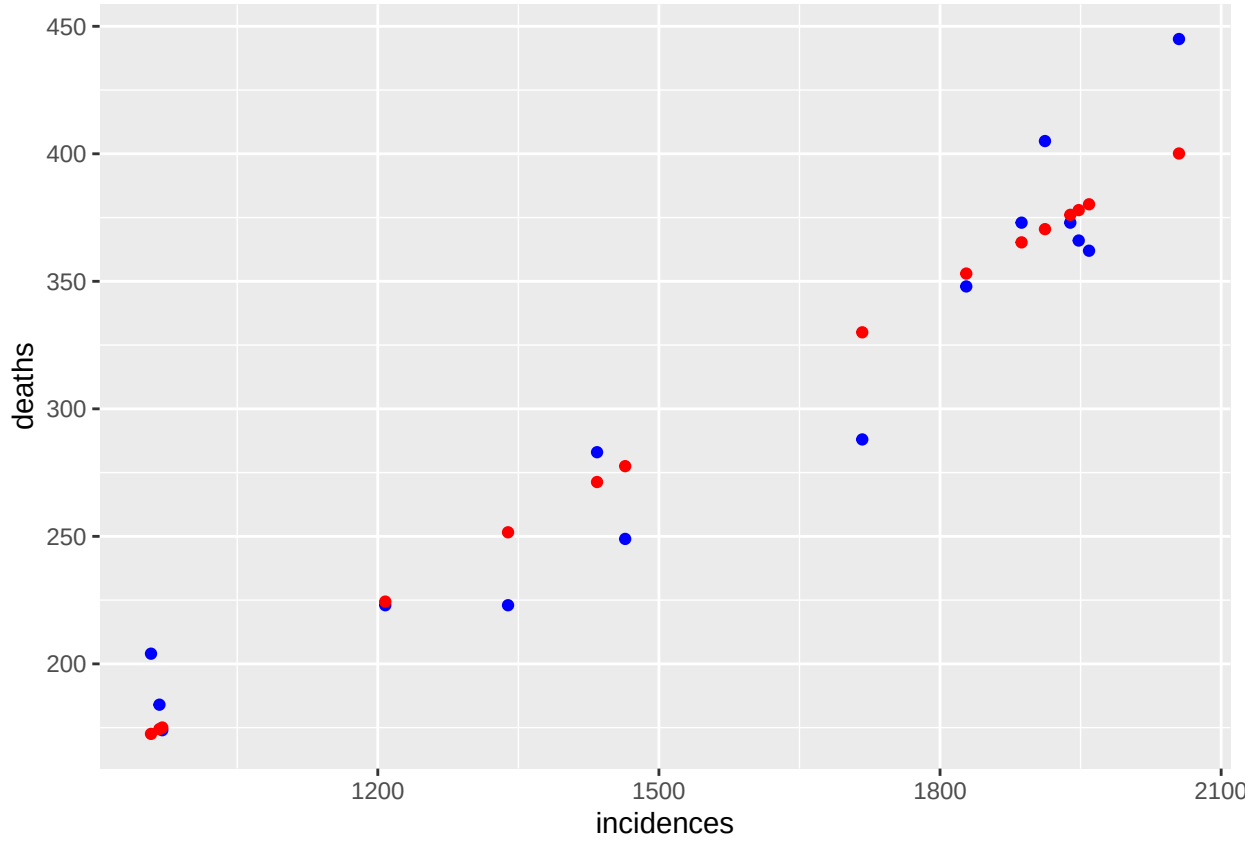
```
##
## Call:
## lm(formula = deaths ~ incidences, data = NYPD_shooting_deaths_per_yr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.010 -15.070  -1.422  10.634  44.875
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -26.16775   27.23773  -0.961    0.354
## incidences    0.20744    0.01681  12.338  1.5e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.41 on 13 degrees of freedom
```

```
## Multiple R-squared:  0.9213, Adjusted R-squared:  0.9153
## F-statistic: 152.2 on 1 and 13 DF,  p-value: 1.497e-08
```

Lets fit the our data into the model and examine how close what's predicted to the actuals

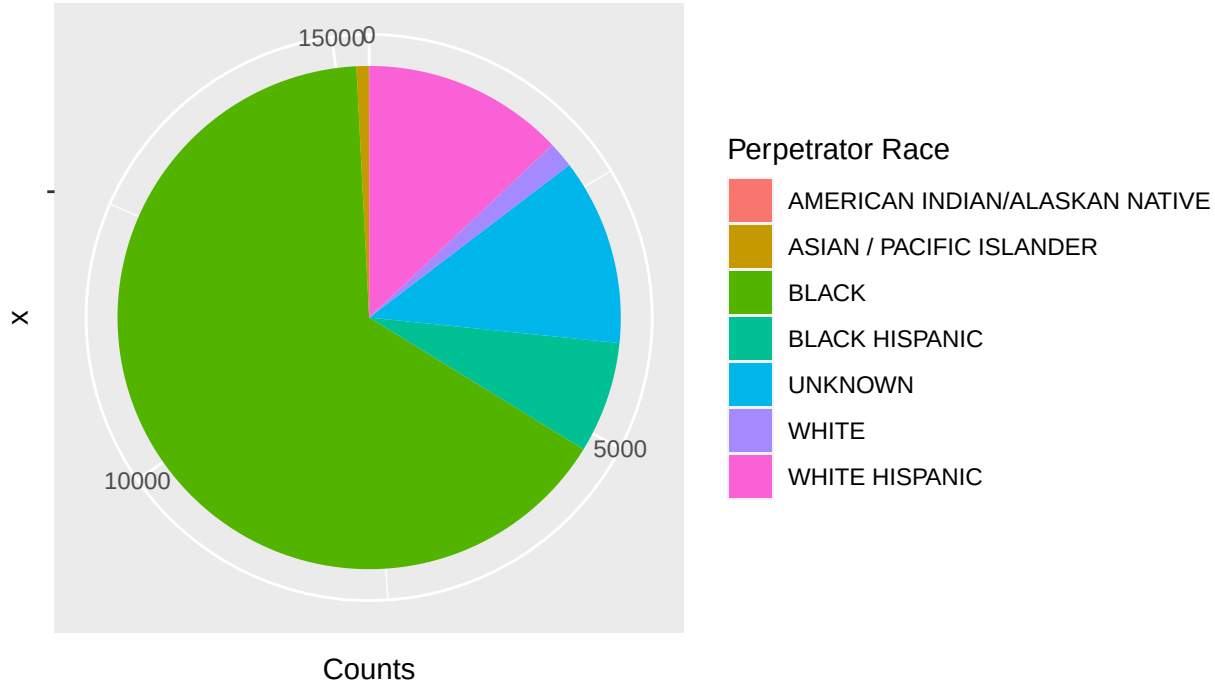
```
NYPD_shooting_deaths_per_yr_pred <- NYPD_shooting_deaths_per_yr %>% mutate(pred = predict(mod))

NYPD_shooting_deaths_per_yr_pred %>% ggplot() +
    geom_point(aes(x=incidences, y=deaths),color = "blue") +
    geom_point(aes(x=incidences, y= pred), color = "red")
```



Above plot does reflect the fact deaths is linear to incidences by year.

Bias analysis in perpetrator race distribution



From what is plotted above, one would conclude dark-skinned people will cause more shooting cases in the world. However, this is considered as one example of overgeneralization bias, as the dataset we are working on is limited to New York which has BLACK as one of the most popular races, so that the same result may not apply to cities like Manila where firarms are also widely available.

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
```

```
##  [1] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     readr_2.1.1     tidyr_1.1.4     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8       assertthat_0.2.1 digest_0.6.29    utf8_1.2.2
##  [5] R6_2.5.1         cellranger_1.1.0 backports_1.4.1  reprex_2.0.1
##  [9] evaluate_0.14    highr_0.9        httr_1.4.2       pillar_1.6.4
## [13] rlang_0.4.12     curl_4.3.2       readxl_1.3.1     rstudioapi_0.13
## [17] rmarkdown_2.11   labeling_0.4.2   bit_4.0.4        munsell_0.5.0
## [21] broom_0.7.11     compiler_4.1.2   modelr_0.1.8     xfun_0.29
## [25] pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.1 fansi_1.0.2
## [29] crayon_1.4.2     tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.3
## [33] grid_4.1.2       jsonlite_1.7.3   gtable_0.3.0     lifecycle_1.0.1
## [37] DBI_1.1.2        magrittr_2.0.1   scales_1.1.1     cli_3.1.0
## [41] stringi_1.7.6    vroom_1.5.7      farver_2.1.0     fs_1.5.2
## [45] xml2_1.3.3       ellipsis_0.3.2   generics_0.1.1   vctrs_0.3.8
## [49] tools_4.1.2      bit64_4.0.5      glue_1.6.0       hms_1.1.1
## [53] parallel_4.1.2   fastmap_1.1.0    yaml_2.2.1       colorspace_2.0-2
## [57] rvest_1.0.2      knitr_1.37       haven_2.4.3
```