NYPD Shooting Incident Data Report

Robin Lutter

30-8-2021

Introduction

In the following I will analyze the NYPD Shooting Incident Data. Here I listed the questions that I would like to answer and the information I would like to obtain from the analysis:

- How are the number of murders distributed over the different boros?
- How are the number of murders changing over the years?
  - Is some sort of trend or pattern recognizable?
  - If yes, what could be the reasons for that?
- How are the murders spread over different races?
  - Is there something noticable?
  - Can we predict a future trend?

To answer these question, I will perform the following steps during the data analysis:

1. Include necessary libraries
2. Import the dataset and give information about the columns
3. Tidy the dataset
4. Modify the dataset and plotting
   - 4.1 Analyze the number of murders for each boro
   - 4.2 Analyze the number of murders over the years
   - 4.3 Analyze the number of murders for each victims race
   - 4.4 Analyze the number of murders for each victims race over the years
5. Modeling the numbers of black victims
6. Conclusion and talking about bias

1. Include necessary libraries

```
library(tidyverse)
library(lubridate)
```

2. Import the dataset and give information about the columns

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

# show a preview of the data
head(nypd_csv_data)
```

```
## # A tibble: 6 x 19
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO          PRECINCT JURISDICTION_CODE
```

```
##          <dbl> <chr>      <chr>      <chr>            <dbl>             <dbl>
## 1    201575314 08/23/2019 22:10:00   QUEENS             103                 0
## 2    205748546 11/27/2019 15:54:00   BRONX               40                 0
## 3    193118596 02/02/2019 19:40:00   MANHATTAN           23                 0
## 4    204192600 10/24/2019 00:52:00   STATEN ISLAND      121                 0
## 5    201483468 08/22/2019 18:03:00   BRONX               46                 0
## 6    198255460 06/07/2019 17:50:00   BROOKLYN            73                 0
## # ... with 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>
```

Column information

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

Column name	Column Description
INCIDENT_KEY	Randomly generated persistent ID for each incident
OCCUR_DATE	Exact date of the shooting incident
OCCUR_TIME	Exact time of the shooting incident
BORO	Borough where the shooting incident occurred
PRECINCT	Precinct where the shooting incident occurred
JURISDICTION_CODE	Jurisdiction where the shooting incident occurred. Jurisdiction codes 0(Patrol), 1(Transit) and 2(Housing) represent NYPD whilst codes 3 and more represent non NYPD jurisdictions
LOCATION_DESC	Location of the shooting incident
STATISTICAL_MURDER_FLAG	Shooting resulted in the victim's death which would be counted as a murder
PERP_AGE_GROUP	Perpetrator's age within a category
PERP_SEX	Perpetrator's sex description
PERP_RACE	Perpetrator's race description
VIC_AGE_GROUP	Victim's age within a category
VIC_SEX	Victim's sex description
VIC_RACE	Victim's race description
X_COORD_CD	Midblock X-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104)
Y_COORD_CD	Midblock Y-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104)
Latitude	Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
Longitude	Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
Lon_Lat	Longitude and Latitude Coordinates for mapping

3. Tidy the dataset

```
# exclude some columns that are not needed for the analysis
nypd_tidy_data <- nypd_csv_data |>
  select(-c(INCIDENT_KEY, X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon_Lat))

# change datatypes for date and time columns from character to date and time
nypd_tidy_data <- nypd_tidy_data |>
  mutate(OCCUR_DATE = mdy(OCCUR_DATE), OCCUR_TIME = hms(OCCUR_TIME))

# change appropriate columns to factor
nypd_tidy_data <- nypd_tidy_data |>
  mutate(PERP_AGE_GROUP = as.factor(PERP_AGE_GROUP)) |>
  mutate(PERP_SEX = as.factor(PERP_SEX)) |>
  mutate(PERP_RACE = as.factor(PERP_RACE)) |>
  mutate(VIC_AGE_GROUP = as.factor(VIC_AGE_GROUP)) |>
  mutate(VIC_SEX = as.factor(VIC_SEX)) |>
  mutate(VIC_RACE = as.factor(VIC_RACE)) |>
  mutate(BORO = as.factor(BORO))

# show a summary of the tidy data
summary(nypd_tidy_data)
```

```
##    OCCUR_DATE           OCCUR_TIME                                BORO
##  Min.   :2006-01-01   Min.   :0S                          BRONX        :6700
##  1st Qu.:2008-12-30   1st Qu.:3H 20M 0S                   BROOKLYN     :9722
##  Median :2012-02-26   Median :15H 0M 0S                   MANHATTAN    :2921
##  Mean   :2012-10-03   Mean   :12H 32M 59.1318737270849S   QUEENS       :3527
##  3rd Qu.:2016-02-28   3rd Qu.:20H 44M 15S                 STATEN ISLAND: 698
##  Max.   :2020-12-31   Max.   :23H 59M 0S
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.0000    Length:23568       Mode :logical
##  1st Qu.: 44.00   1st Qu.:0.0000    Class :character   FALSE:19080
##  Median : 69.00   Median :0.0000    Mode  :character   TRUE :4488
##  Mean   : 66.21   Mean   :0.3323
##  3rd Qu.: 81.00   3rd Qu.:0.0000
##  Max.   :123.00   Max.   :2.0000
##                   NA's   :2
##  PERP_AGE_GROUP PERP_SEX              PERP_RACE    VIC_AGE_GROUP   VIC_SEX
##  18-24  :5448   F   :  334   BLACK         :9855   <18    : 2525   F: 2195
##  25-44  :4613   M   :13305   WHITE HISPANIC:1961   18-24  : 9000   M:21353
##  UNKNOWN:3156   U   : 1504   UNKNOWN       :1869   25-44  :10287   U:   20
##  <18    :1354   NA's: 8425   BLACK HISPANIC:1081   45-64  : 1536
##  45-64  : 481                WHITE         : 255   65+    :  155
##  (Other):  57                (Other)       : 122   UNKNOWN:   65
##  NA's   :8459                NA's          :8425
##                            VIC_RACE
##  AMERICAN INDIAN/ALASKAN NATIVE:    9
##  ASIAN / PACIFIC ISLANDER      :  320
##  BLACK                         :16846
##  BLACK HISPANIC                : 2244
##  UNKNOWN                       :  102
##  WHITE                         :  615
```

```
##  WHITE HISPANIC                : 3432
```

As we can see, there is missing data in some of the columns. For the analysis that I want to perform, this won't be a problem, because I will focus on the number of murders with respect to the victims race and the boro where the incident occured. For these cases, all the necessary data is available.

In case we wanted to evaluate data from columns with missing values, we would have to handle them in one of many possible ways. Since we don't know much more about the data, the easiest way would be either to ignore those rows for further analysis or replace the missing value with an calculated average value instead (if this makes sense for the specific column).

4. Modify the dataset and plotting

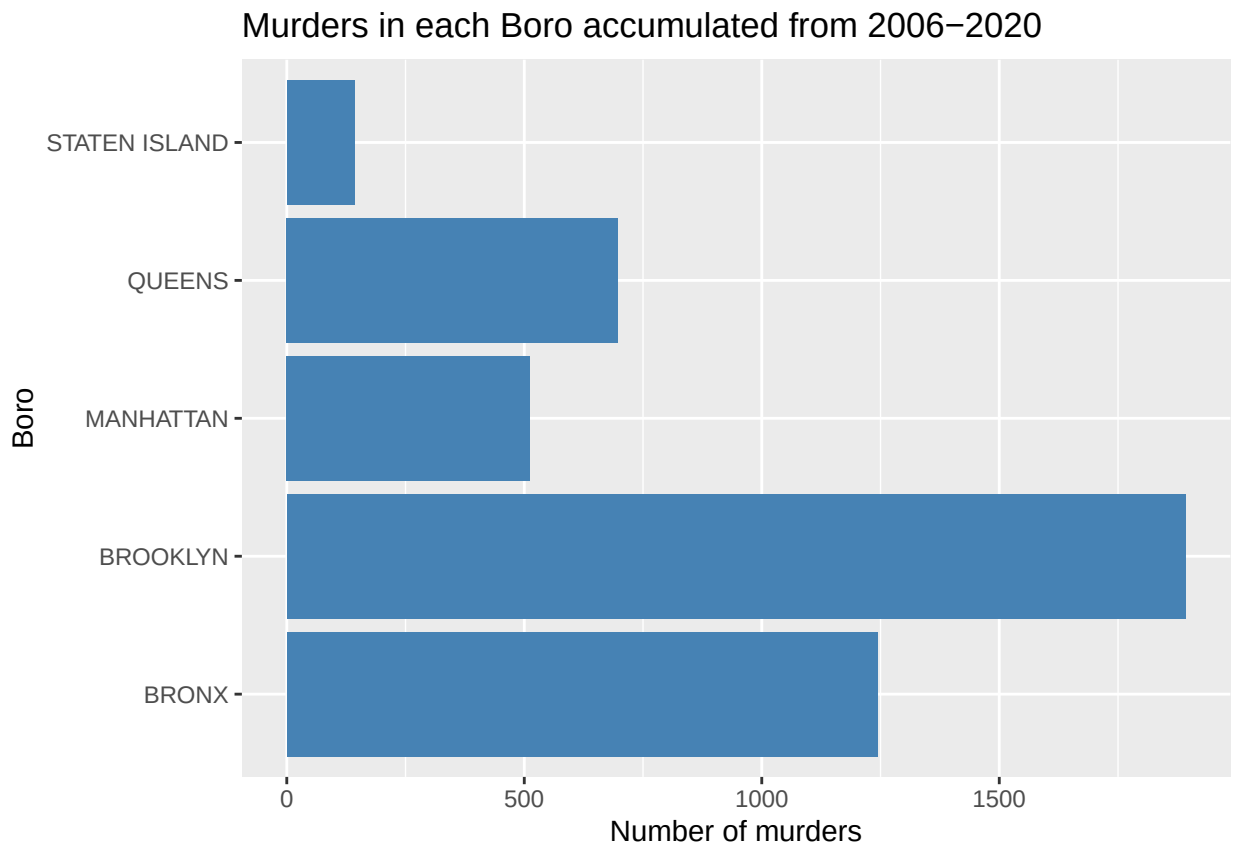
4.1 Analyze the number of murders for each boro

```
# get all murders for each boro over all the years
nypd_murders_by_boro <- nypd_tidy_data |>
  group_by(BORO) |>
  summarize(MURDERS = sum(STATISTICAL_MURDER_FLAG == TRUE)) |>
  select(BORO, MURDERS) |>
  ungroup()

head(nypd_murders_by_boro)
```

```
## # A tibble: 5 x 2
##   BORO          MURDERS
##   <fct>           <int>
## 1 BRONX            1244
## 2 BROOKLYN         1892
## 3 MANHATTAN         512
## 4 QUEENS            697
## 5 STATEN ISLAND     143
```

```
# plot murders in each boro
nypd_murders_by_boro |>
  ggplot(aes(x = MURDERS, y = BORO)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Murders in each Boro accumulated from 2006-2020",
       x = "Number of murders", y = "Boro")
```



The plot shows, that most of the murders occur in Brooklyn and Bronx, whereas Staten Island has a low murder rate. If we wanted to perform a deeper interpretation of this distribution, we would need more data. For example, if we knew the number of people living in each boro, we could check if one has a higher or lower crime rate than the others with respect to their total population.

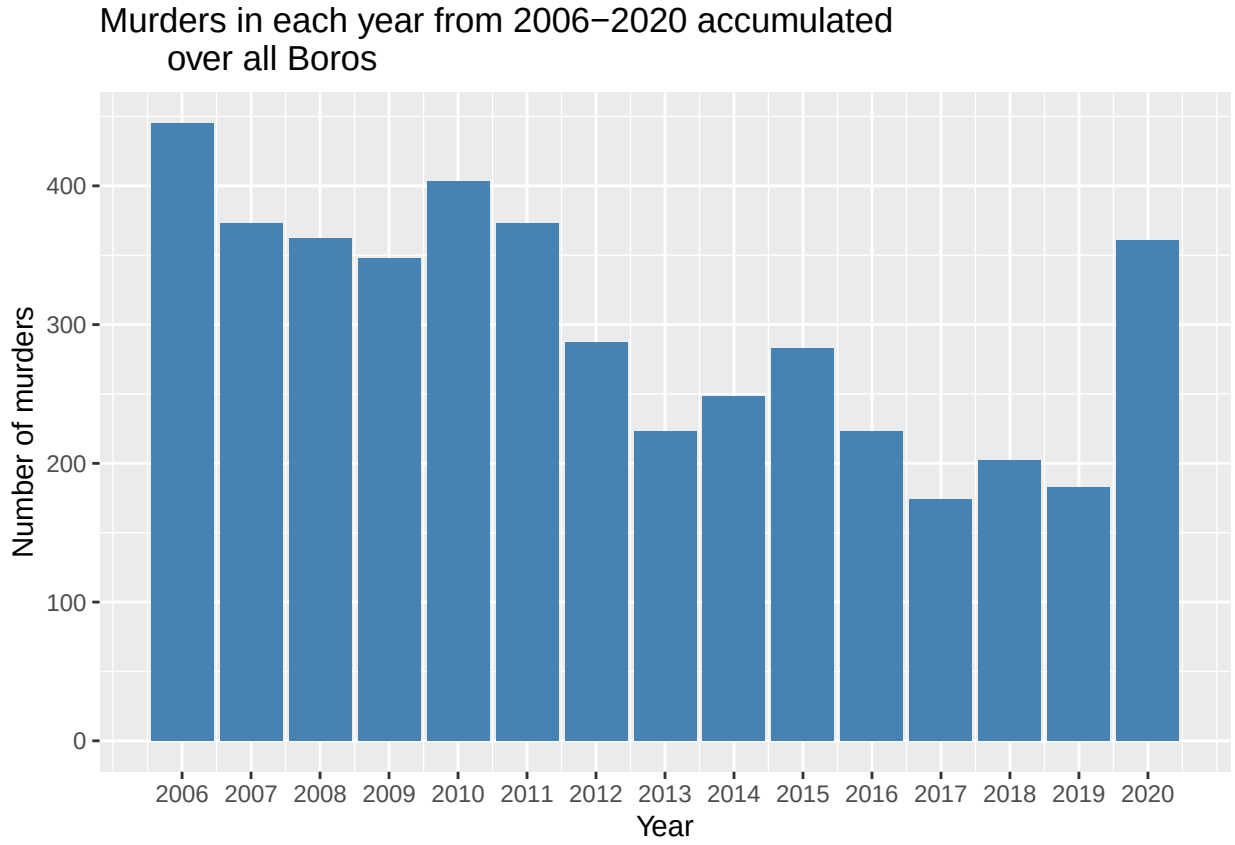
4.2 Analyze the number of murders over the years

```
# get all murders for each year
nypd_murders_by_year <- nypd_tidy_data |>
  mutate(YEAR = year(nypd_tidy_data$OCCUR_DATE)) |>
  group_by(YEAR) |>
  summarize(MURDERS = sum(STATISTICAL_MURDER_FLAG == TRUE)) |>
  select(YEAR, MURDERS) |>
  ungroup()

head(nypd_murders_by_year)
```

```
## # A tibble: 6 x 2
##    YEAR MURDERS
##   <dbl>   <int>
## 1  2006     445
## 2  2007     373
## 3  2008     362
## 4  2009     348
## 5  2010     403
## 6  2011     373
```

```
# plot murders in each year
nypd_murders_by_year |>
  ggplot(aes(x = YEAR, y = MURDERS)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Murders in each year from 2006-2020 accumulated
       over all Boros", x = "Year", y = "Number of murders") +
  scale_x_continuous(breaks=c(2006:2020), labels=c(2006:2020))
```



We can see, that the number of murders had a decreasing trend over the years, whereas in 2020 there was a significant increase. To understand the reason for this, again we would need more data. One of the influencing factors could be the COVID-19 pandemic. Many people lost their jobs, were frustrated and had not much of an engangement. This could have lead to an increasing level of crime. But with the existing base of data we can't be entirely sure about that.

4.3 Analyze the number of murders for each victims race

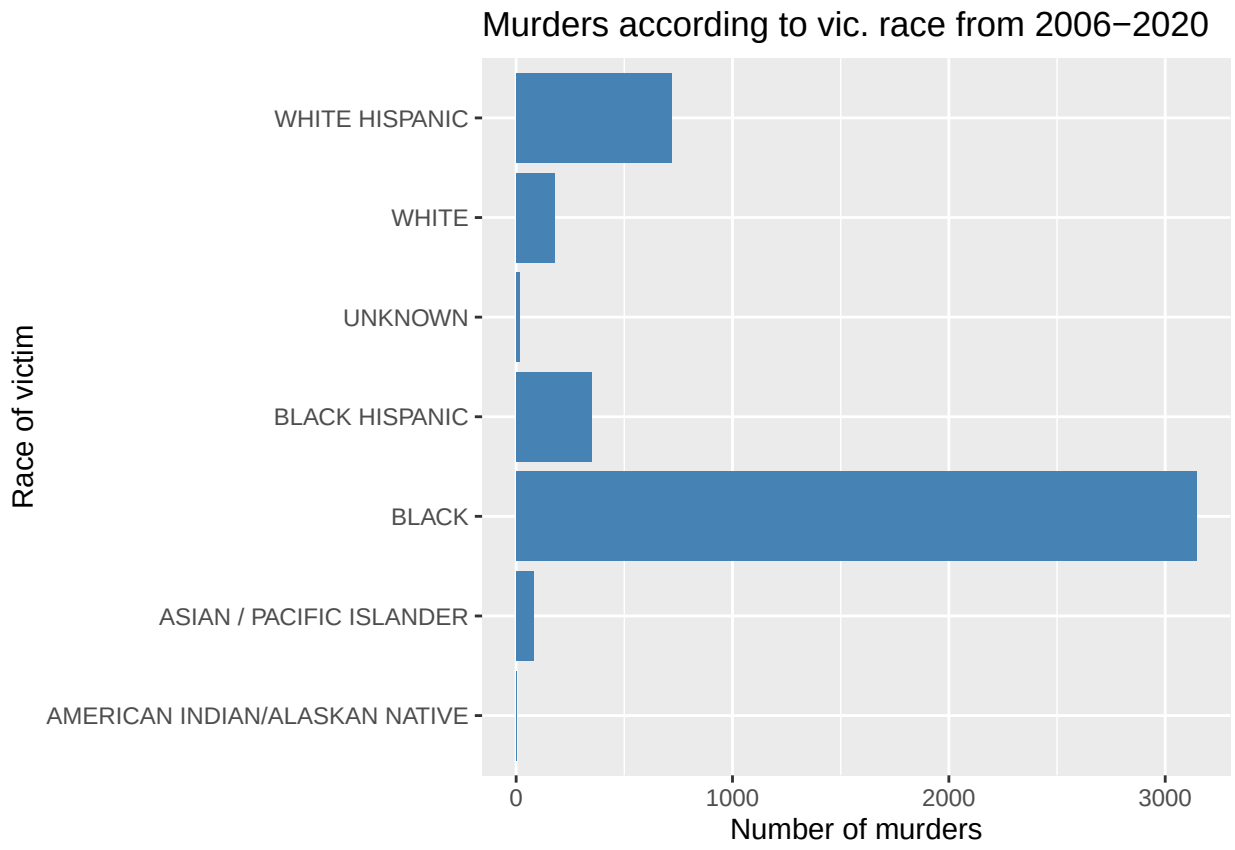
```
# get all murders listed for each victims race
nypd_murders_by_vic_race <- nypd_tidy_data |>
  group_by(VIC_RACE) |>
  summarize(MURDERS = sum(STATISTICAL_MURDER_FLAG ==TRUE)) |>
  select(VIC_RACE, MURDERS) |>
  ungroup()

head(nypd_murders_by_vic_race)
```

```
## # A tibble: 6 x 2
```

```
##   VIC_RACE                          MURDERS
##   <fct>                               <int>
## 1 AMERICAN INDIAN/ALASKAN NATIVE          0
## 2 ASIAN / PACIFIC ISLANDER               81
## 3 BLACK                                3144
## 4 BLACK HISPANIC                        351
## 5 UNKNOWN                                17
## 6 WHITE                                 177
```

```
# plot murders according to victims race
nypd_murders_by_vic_race |>
  ggplot(aes(x = MURDERS, y = VIC_RACE)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Murders according to vic. race from 2006-2020",
       x = "Number of murders", y = "Race of victim")
```



The plot shows that most of the victims were black, whereas no american indian/alaskan native person was murdered in the last 15 years. Again we would need more information about the proportion of each race in the whole population of New York to get a better understanding of those numbers and their relationships.

4.4 Analyze the number of murders for each victims race over the years

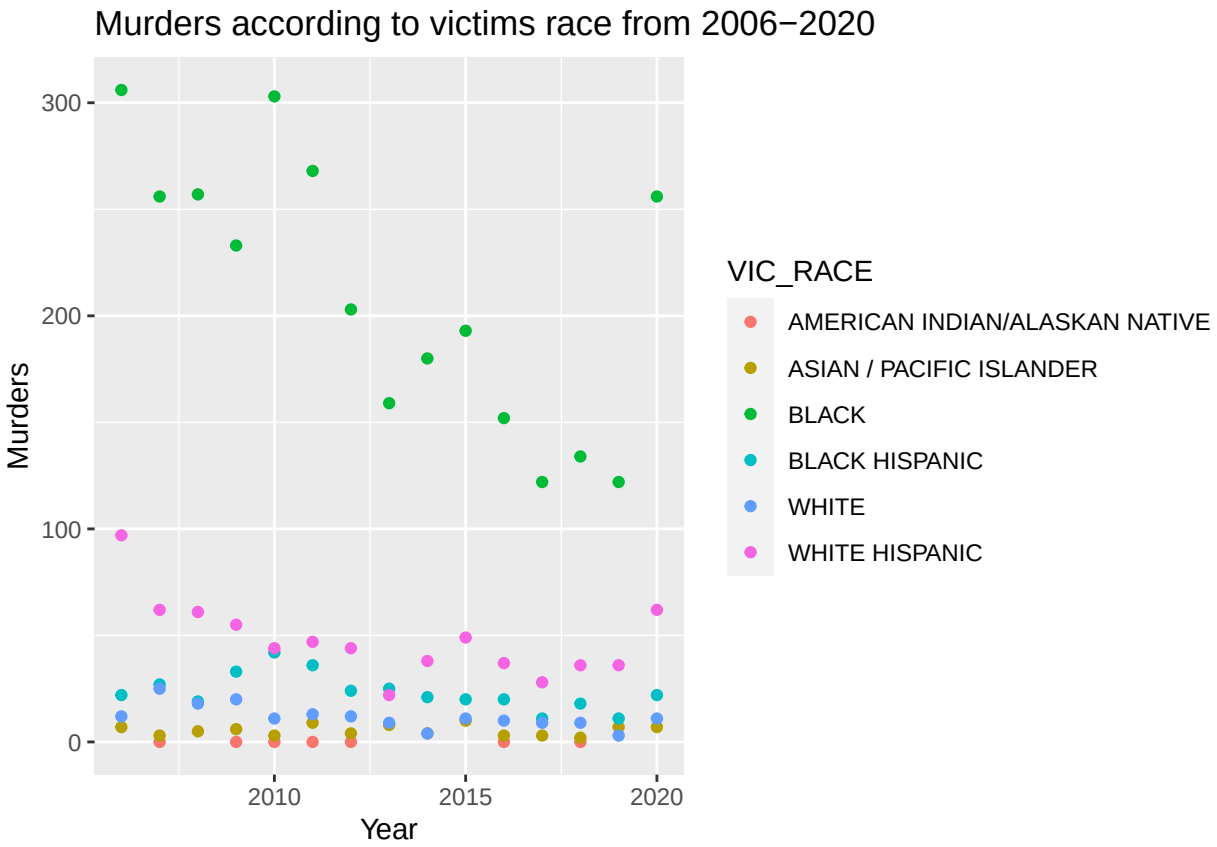
```
# get murders for each year and victims race
nypd_murder_by_year_race <- nypd_tidy_data |>
  group_by(YEAR = year(nypd_tidy_data$OCCUR_DATE), VIC_RACE) |>
  summarize(MURDERS = sum(STATISTICAL_MURDER_FLAG == TRUE),
```

```
            CASES = sum(STATISTICAL_MURDER_FLAG == TRUE |
                          STATISTICAL_MURDER_FLAG == FALSE)) |>
  mutate(DEATH_PERCENTAGE = MURDERS / CASES) |>
  select(YEAR, VIC_RACE, MURDERS, CASES, DEATH_PERCENTAGE) |>
  ungroup()

head(nypd_murder_by_year_race)
```

```
## # A tibble: 6 x 5
##    YEAR VIC_RACE                 MURDERS CASES DEATH_PERCENTAGE
##   <dbl> <fct>                      <int> <int>            <dbl>
## 1  2006 ASIAN / PACIFIC ISLANDER       7    26            0.269
## 2  2006 BLACK                        306  1422            0.215
## 3  2006 BLACK HISPANIC                22   114            0.193
## 4  2006 UNKNOWN                       1     2            0.5
## 5  2006 WHITE                        12    46            0.261
## 6  2006 WHITE HISPANIC               97   445            0.218
```

```
# plot murders according to victims race over the years
nypd_murder_by_year_race |>
  filter(VIC_RACE != "UNKNOWN") |>
  ggplot(aes(x = YEAR, y = MURDERS, color = VIC_RACE)) +
  geom_point() +
  labs(title = "Murders according to victims race from 2006-2020",
       x = "Year", y = "Murders")
```



Here we can see the distribution of murders for each race over the last 15 years. As we have seen before,

most of the victims were black people. At this point we can get another insight. The black victims are the only ones where the number of murders significantly decreased over the years (excluding the year 2020). For all other races there was a nearly constant amount of murders each year. This means that the overall decrease in murders that we have seen in the plot of chapter 4.2 is caused mainly by the decrease of murders of black people.

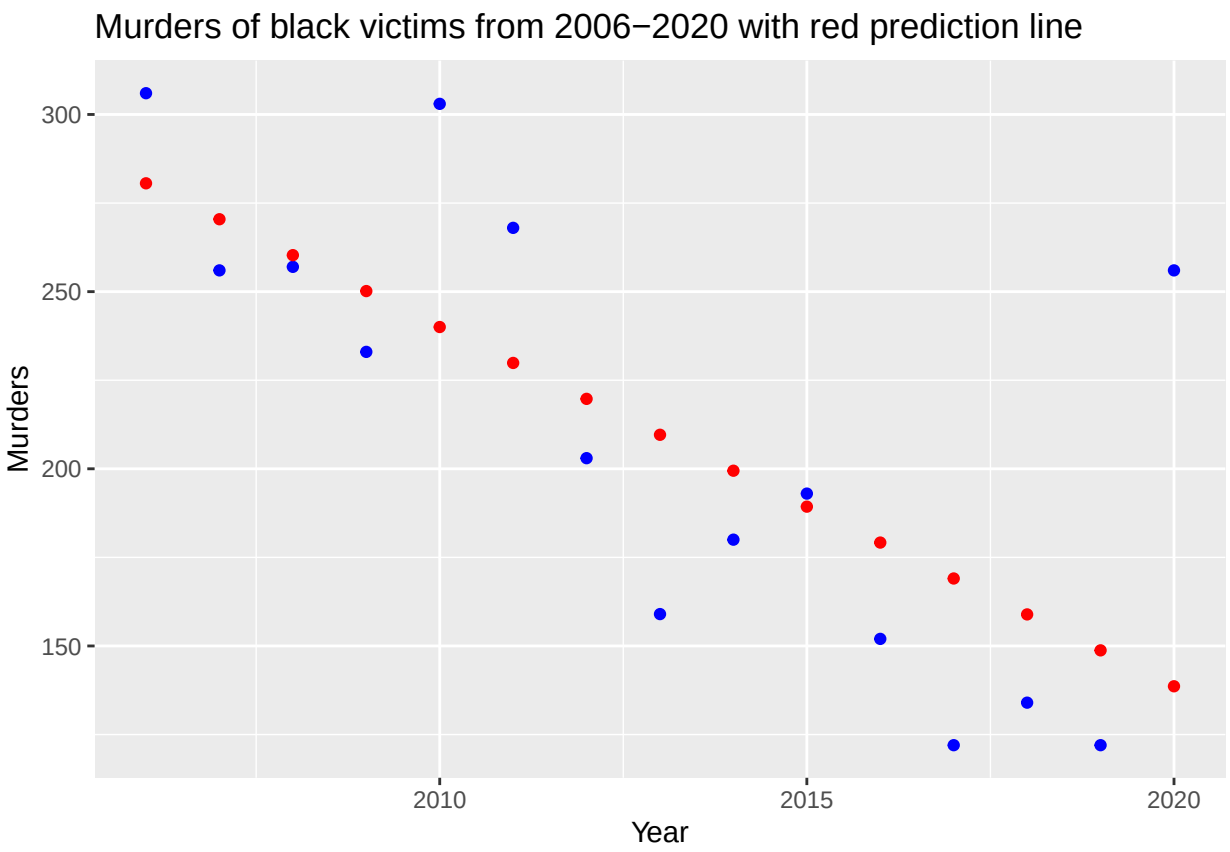
5. Modeling the numbers of black victims

```
# linear model
model_data <- nypd_murder_by_year_race |>
  filter(VIC_RACE == "BLACK")

model_murders <- lm(MURDERS ~ YEAR, data = model_data)

# add a prediction column
model_data <- model_data |>
  mutate(MURDERS_PRED = predict(model_murders))

# visualize the prediction of murders
model_data |> ggplot() +
  geom_point(aes(x = YEAR, y = MURDERS), color = "blue") +
  geom_point(aes(x = YEAR, y = MURDERS_PRED), color = "red") +
  labs(title = "Murders of black victims from 2006-2020 with red prediction line",
       x = "Year", y = "Murders")
```



The blue points represent the number murdered black people over the years. The red line shows the linear prediction. As we have already seen, the numbers are decreasing overall with a remarkable exception in 2020 that was discussed in chapter 4.2 and 4.4. Of course, outliers like the one in 2020 should be investigated further to draw more insight from it, but for this project, I will stop the analysis at this point.

6. Conclusion and talking about bias

I hope this report gave you a solid overview over the NYPD Shooting Incident Data. Of course not all aspects of the data set were covered in this report. For example the features like age and sex of the perpetrator and victims were not covered at all in this analysis. This information surely would allow a deeper understanding of those incidents and murders.

As with all data analysis, there is always some sort of bias. On the one hand, the data itself can be biased in such a way, that some incidents are not covered in the data set at all, due to certain (unknown) circumstances. On the other hand, the one who analysis the data can be biased. One example could be a bias in the presentation of the results, which can lead to a kind of controlled focus of the reader of the report. For example, if I would have made some of the plots visually much more noticeable than others, I can make the reader focus on facts that I want the reader to focus on. The avoid such bias, I tried to make all plots as equally appealing as possible.
Another source of bias can be the choice of the model. Since I already assumed a linearly decreasing number of murders of the years according to my plots, my first choice for a model was also a linear one. At this point, one should dive in deeper and try out other types of models to see, if the initial assumption was appropriate.

Session Info

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.1 (2021-08-10)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  German_Germany.1252
##  ctype    German_Germany.1252
##  tz       Europe/Berlin
##  date     2021-09-04
##
## - Packages -------------------------------------------------------------------
##  package     * version date       lib source
##  assertthat    0.2.1   2019-03-21 [1] CRAN (R 4.1.0)
##  backports     1.2.1   2020-12-09 [1] CRAN (R 4.1.0)
##  bit           4.0.4   2020-08-04 [1] CRAN (R 4.1.0)
##  bit64         4.0.5   2020-08-30 [1] CRAN (R 4.1.0)
##  broom         0.7.9   2021-07-27 [1] CRAN (R 4.1.0)
##  cachem        1.0.5   2021-05-15 [1] CRAN (R 4.1.0)
##  callr         3.7.0   2021-04-20 [1] CRAN (R 4.1.0)
##  cellranger    1.1.0   2016-07-27 [1] CRAN (R 4.1.0)
##  cli           3.0.1   2021-07-17 [1] CRAN (R 4.1.0)
##  colorspace    2.0-2   2021-06-24 [1] CRAN (R 4.1.0)
##  crayon        1.4.1   2021-02-08 [1] CRAN (R 4.1.0)
##  curl          4.3.2   2021-06-23 [1] CRAN (R 4.1.0)
```

```
##  DBI           1.1.1   2021-01-15 [1] CRAN (R 4.1.0)
##  dbplyr        2.1.1   2021-04-06 [1] CRAN (R 4.1.0)
##  desc          1.3.0   2021-03-05 [1] CRAN (R 4.1.0)
##  devtools      2.4.2   2021-06-07 [1] CRAN (R 4.1.0)
##  digest        0.6.27  2020-10-24 [1] CRAN (R 4.1.0)
##  dplyr       * 1.0.7   2021-06-18 [1] CRAN (R 4.1.0)
##  ellipsis      0.3.2   2021-04-29 [1] CRAN (R 4.1.0)
##  evaluate      0.14    2019-05-28 [1] CRAN (R 4.1.0)
##  fansi         0.5.0   2021-05-25 [1] CRAN (R 4.1.0)
##  farver        2.1.0   2021-02-28 [1] CRAN (R 4.1.0)
##  fastmap       1.1.0   2021-01-25 [1] CRAN (R 4.1.0)
##  forcats     * 0.5.1   2021-01-27 [1] CRAN (R 4.1.0)
##  fs            1.5.0   2020-07-31 [1] CRAN (R 4.1.0)
##  generics      0.1.0   2020-10-31 [1] CRAN (R 4.1.0)
##  ggplot2     * 3.3.5   2021-06-25 [1] CRAN (R 4.1.0)
##  glue          1.4.2   2020-08-27 [1] CRAN (R 4.1.0)
##  gtable        0.3.0   2019-03-25 [1] CRAN (R 4.1.0)
##  haven         2.4.3   2021-08-04 [1] CRAN (R 4.1.0)
##  highr         0.9     2021-04-16 [1] CRAN (R 4.1.0)
##  hms           1.1.0   2021-05-17 [1] CRAN (R 4.1.0)
##  htmltools     0.5.1.1 2021-01-22 [1] CRAN (R 4.1.0)
##  httr          1.4.2   2020-07-20 [1] CRAN (R 4.1.0)
##  jsonlite      1.7.2   2020-12-09 [1] CRAN (R 4.1.0)
##  knitr         1.33    2021-04-24 [1] CRAN (R 4.1.0)
##  labeling      0.4.2   2020-10-20 [1] CRAN (R 4.1.0)
##  lifecycle     1.0.0   2021-02-15 [1] CRAN (R 4.1.0)
##  lubridate   * 1.7.10  2021-02-26 [1] CRAN (R 4.1.0)
##  magrittr      2.0.1   2020-11-17 [1] CRAN (R 4.1.0)
##  memoise       2.0.0   2021-01-26 [1] CRAN (R 4.1.0)
##  modelr        0.1.8   2020-05-19 [1] CRAN (R 4.1.0)
##  munsell       0.5.0   2018-06-12 [1] CRAN (R 4.1.0)
##  pillar        1.6.2   2021-07-29 [1] CRAN (R 4.1.0)
##  pkgbuild      1.2.0   2020-12-15 [1] CRAN (R 4.1.0)
##  pkgconfig     2.0.3   2019-09-22 [1] CRAN (R 4.1.0)
##  pkgload       1.2.1   2021-04-06 [1] CRAN (R 4.1.0)
##  prettyunits   1.1.1   2020-01-24 [1] CRAN (R 4.1.0)
##  processx      3.5.2   2021-04-30 [1] CRAN (R 4.1.0)
##  ps            1.6.0   2021-02-28 [1] CRAN (R 4.1.0)
##  purrr       * 0.3.4   2020-04-17 [1] CRAN (R 4.1.0)
##  R6            2.5.1   2021-08-19 [1] CRAN (R 4.1.1)
##  Rcpp          1.0.7   2021-07-07 [1] CRAN (R 4.1.0)
##  readr       * 2.0.1   2021-08-10 [1] CRAN (R 4.1.1)
##  readxl        1.3.1   2019-03-13 [1] CRAN (R 4.1.0)
##  remotes       2.4.0   2021-06-02 [1] CRAN (R 4.1.0)
##  reprex        2.0.1   2021-08-05 [1] CRAN (R 4.1.0)
##  rlang         0.4.11  2021-04-30 [1] CRAN (R 4.1.0)
##  rmarkdown     2.10    2021-08-06 [1] CRAN (R 4.1.1)
##  rprojroot     2.0.2   2020-11-15 [1] CRAN (R 4.1.0)
##  rstudioapi    0.13    2020-11-12 [1] CRAN (R 4.1.0)
##  rvest         1.0.1   2021-07-26 [1] CRAN (R 4.1.0)
##  scales        1.1.1   2020-05-11 [1] CRAN (R 4.1.0)
##  sessioninfo   1.1.1   2018-11-05 [1] CRAN (R 4.1.0)
##  stringi       1.7.3   2021-07-16 [1] CRAN (R 4.1.0)
##  stringr     * 1.4.0   2019-02-10 [1] CRAN (R 4.1.0)
```

```
##  testthat      3.0.4   2021-07-01 [1] CRAN (R 4.1.0)
##  tibble      * 3.1.3   2021-07-23 [1] CRAN (R 4.1.0)
##  tidyr       * 1.1.3   2021-03-03 [1] CRAN (R 4.1.0)
##  tidyselect    1.1.1   2021-04-30 [1] CRAN (R 4.1.0)
##  tidyverse   * 1.3.1   2021-04-15 [1] CRAN (R 4.1.0)
##  tzdb          0.1.2   2021-07-20 [1] CRAN (R 4.1.0)
##  usethis       2.0.1   2021-02-10 [1] CRAN (R 4.1.0)
##  utf8          1.2.2   2021-07-24 [1] CRAN (R 4.1.0)
##  vctrs         0.3.8   2021-04-29 [1] CRAN (R 4.1.0)
##  vroom         1.5.4   2021-08-05 [1] CRAN (R 4.1.0)
##  withr         2.4.2   2021-04-18 [1] CRAN (R 4.1.0)
##  xfun          0.25    2021-08-06 [1] CRAN (R 4.1.0)
##  xml2          1.3.2   2020-04-23 [1] CRAN (R 4.1.0)
##  yaml          2.2.1   2020-02-01 [1] CRAN (R 4.1.0)
##
## [1] C:/Users/Robin/Documents/R/win-library/4.1
## [2] C:/Program Files/R/R-4.1.1/library
```