NYPD_Project_RMD

Roman N.

2023-04-21

Introduction

In this Rmd document we're analyzing the shooting project dataset, which can be obtained from the following link: Shooting Project Dataset.

Load Libraries and Import Data

1. Loading libraries

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

2. Reading the dataset from CSV file

```
data <- 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.
```

Data Summary

```
summary(data)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:27312       Length:27312      Length:27312
##  1st Qu.: 63860880   Class :character   Class1:hms        Class :character
##  Median : 90372218   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :120860536                      Mode  :numeric
##  3rd Qu.:188810230
##  Max.   :261190187
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character
##                     Mean   : 65.64   Mean   :0.3269
##                     3rd Qu.: 81.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##                                      NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312       Mode :logical           Length:27312
##  Class :character   FALSE:22046             Class :character
##  Mode  :character   TRUE :5266              Mode  :character
##
##
##
##
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX
##  Length:27312       Length:27312       Length:27312       Length:27312
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
```

```
##  Length:27312       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000029   1st Qu.:182834   1st Qu.:40.67
##  Mode  :character   Median :1007731   Median :194487   Median :40.70
##                     Mean   :1009449   Mean   :208127   Mean   :40.74
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :10
##    Longitude        Lon_Lat
##  Min.   :-74.25   Length:27312
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :10
```

As we can see all of the columns have no correct format. Let's change appropriate variables to factor and date types and getting rid of any columns not needed.

Convert appropriate variables to factor and date types

```
data$OCCUR_DATE <- as.Date(data$OCCUR_DATE, format = "%m/%d/%Y")
data$BORO <- as.factor(data$BORO)
data$PRECINCT <- as.factor(data$PRECINCT)
data$LOCATION_DESC <- as.factor(data$LOCATION_DESC)
data$STATISTICAL_MURDER_FLAG <- as.logical(data$STATISTICAL_MURDER_FLAG)
data$PERP_AGE_GROUP <- as.factor(data$PERP_AGE_GROUP)
data$PERP_SEX <- as.factor(data$PERP_SEX)
data$PERP_RACE <- as.factor(data$PERP_RACE)
data$VIC_AGE_GROUP <- as.factor(data$VIC_AGE_GROUP)
data$VIC_SEX <- as.factor(data$VIC_SEX)
data$VIC_RACE <- as.factor(data$VIC_RACE)
```

Remove unnecessary columns

I'm not going to do any geospatial maps and their analysis so I removed those columns.

```
data <- select(data, -c(OCCUR_TIME, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon
```

Check for missing data by columns

```
colSums(is.na(data))
```

```
##            INCIDENT_KEY             OCCUR_DATE                    BORO
##                       0                      0                       0
##      LOC_OF_OCCUR_DESC               PRECINCT     LOC_CLASSFCTN_DESC
```

```
##                   25596                       0                   25596
##           LOCATION_DESC STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP
##                   14977                       0                    9344
##                PERP_SEX               PERP_RACE           VIC_AGE_GROUP
##                    9310                    9310                       0
##                 VIC_SEX                VIC_RACE
##                       0                       0
```

Here we have a problem with missing data. It's obvious that NYPD has no enough information about some crime places or even criminals. All missing values we'll replace with "unknown" or its equivalent.

Replace missing values

```
data$LOCATION_DESC <- replace(data$LOCATION_DESC, is.na(data$LOCATION_DESC), "NONE")
data <- data %>%
  mutate(PERP_AGE_GROUP = recode(PERP_AGE_GROUP, "1020" = "UNKNOWN", "224" = "UNKNOWN", "940" = "UNKNOW
data$PERP_AGE_GROUP <- replace(data$PERP_AGE_GROUP, is.na(data$PERP_AGE_GROUP), "UNKNOWN")
data$PERP_SEX <- replace(data$PERP_SEX, is.na(data$PERP_SEX), "U")
data$PERP_RACE <- replace(data$PERP_RACE, is.na(data$PERP_RACE), "UNKNOWN")
```

Check wrangling result

```
colSums(is.na(data))
```

```
##            INCIDENT_KEY               OCCUR_DATE                     BORO
##                       0                        0                        0
##       LOC_OF_OCCUR_DESC                 PRECINCT      LOC_CLASSFCTN_DESC
##                   25596                        0                    25596
##           LOCATION_DESC STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP
##                       0                        0                        0
##                PERP_SEX               PERP_RACE           VIC_AGE_GROUP
##                       0                        0                        0
##                 VIC_SEX                VIC_RACE
##                       0                        0
```

```
summary(data)
```

```
##   INCIDENT_KEY         OCCUR_DATE                     BORO
##  Min.   :  9953245   Min.   :2006-01-01   BRONX        : 7937
##  1st Qu.: 63860880   1st Qu.:2009-07-18   BROOKLYN     :10933
##  Median : 90372218   Median :2013-04-29   MANHATTAN    : 3572
##  Mean   :120860536   Mean   :2014-01-06   QUEENS       : 4094
##  3rd Qu.:188810230   3rd Qu.:2018-10-15   STATEN ISLAND:  776
##  Max.   :261190187   Max.   :2022-12-31
##
##  LOC_OF_OCCUR_DESC     PRECINCT     LOC_CLASSFCTN_DESC
##  Length:27312       75     : 1557   Length:27312
```

```
##  Class :character   73     : 1452   Class :character
##  Mode  :character   67     : 1216   Mode  :character
##                     44     : 1020
##                     79     : 1012
##                     47     :  953
##                     (Other):20102
##                    LOCATION_DESC   STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  NONE                      :15152   Mode :logical           (null) :  640
##  MULTI DWELL - PUBLIC HOUS: 4832   FALSE:22046             <18    : 1591
##  MULTI DWELL - APT BUILD  : 2835   TRUE :5266              UNKNOWN:12495
##  (null)                   :  977                           18-24  : 6222
##  PVT HOUSE                :  951                           25-44  : 5687
##  GROCERY/BODEGA           :  694                           45-64  :  617
##  (Other)                  : 1871                           65+    :   60
##    PERP_SEX              PERP_RACE     VIC_AGE_GROUP   VIC_SEX
##  (null):  640   BLACK         :11432   <18    : 2839   F: 2615
##  F     :  424   UNKNOWN       :11146   1022   :    1   M:24686
##  M     :15439   WHITE HISPANIC: 2341   18-24  :10086   U:   11
##  U     :10809   BLACK HISPANIC: 1314   25-44  :12281
##                 (null)        :  640   45-64  : 1863
##                 WHITE         :  283   65+    :  181
##                 (Other)       :  156   UNKNOWN:   61
##                            VIC_RACE
##  AMERICAN INDIAN/ALASKAN NATIVE:   10
##  ASIAN / PACIFIC ISLANDER      :  404
##  BLACK                         :19439
##  BLACK HISPANIC                : 2646
##  UNKNOWN                       :   66
##  WHITE                         :  698
##  WHITE HISPANIC                : 4049
```

Finally we have a cleaned-up dataset to work with.

Basic visualizations, analysis, linear model

First of all we create several dataframes for analysis.

```
shooting_by_race_total <- data %>%
    group_by(VIC_RACE) %>%
    summarize(TOTAL = n()) %>%
    ungroup()
shooting_by_age_group_total <- data %>%
    group_by(VIC_AGE_GROUP) %>%
    summarize(TOTAL = n()) %>%
    ungroup()
shooting_in_boro_by_race <- data %>%
    group_by(BORO, VIC_RACE) %>%
    summarize(COUNT = n()) %>%
    ungroup() %>%
    left_join(shooting_by_race_total, by = "VIC_RACE") %>%
    mutate(PERCENTAGE = COUNT / TOTAL) %>%
    select(BORO, VIC_RACE, COUNT, PERCENTAGE) %>%
    ungroup()
```

```
## `summarise()` has grouped output by 'BORO'. You can override using the
## `.groups` argument.
```

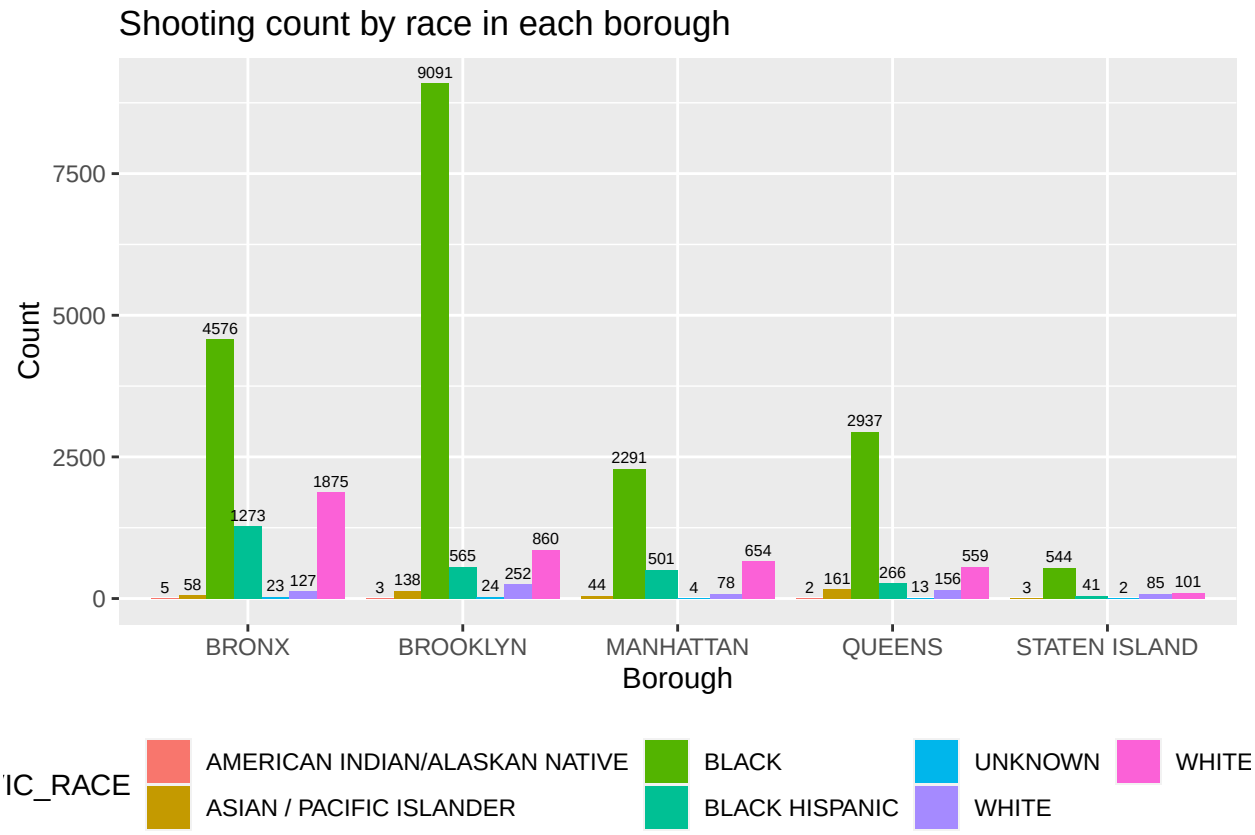
```
shooting_in_boro_by_age_group <- data %>%
    group_by(BORO, VIC_AGE_GROUP) %>%
    summarize(COUNT = n()) %>%
    ungroup() %>%
    left_join(shooting_by_age_group_total, by = "VIC_AGE_GROUP") %>%
    mutate(PERCENTAGE = COUNT / TOTAL) %>%
    select(BORO, VIC_AGE_GROUP, COUNT, PERCENTAGE) %>%
    ungroup()
```

```
## `summarise()` has grouped output by 'BORO'. You can override using the
## `.groups` argument.
```

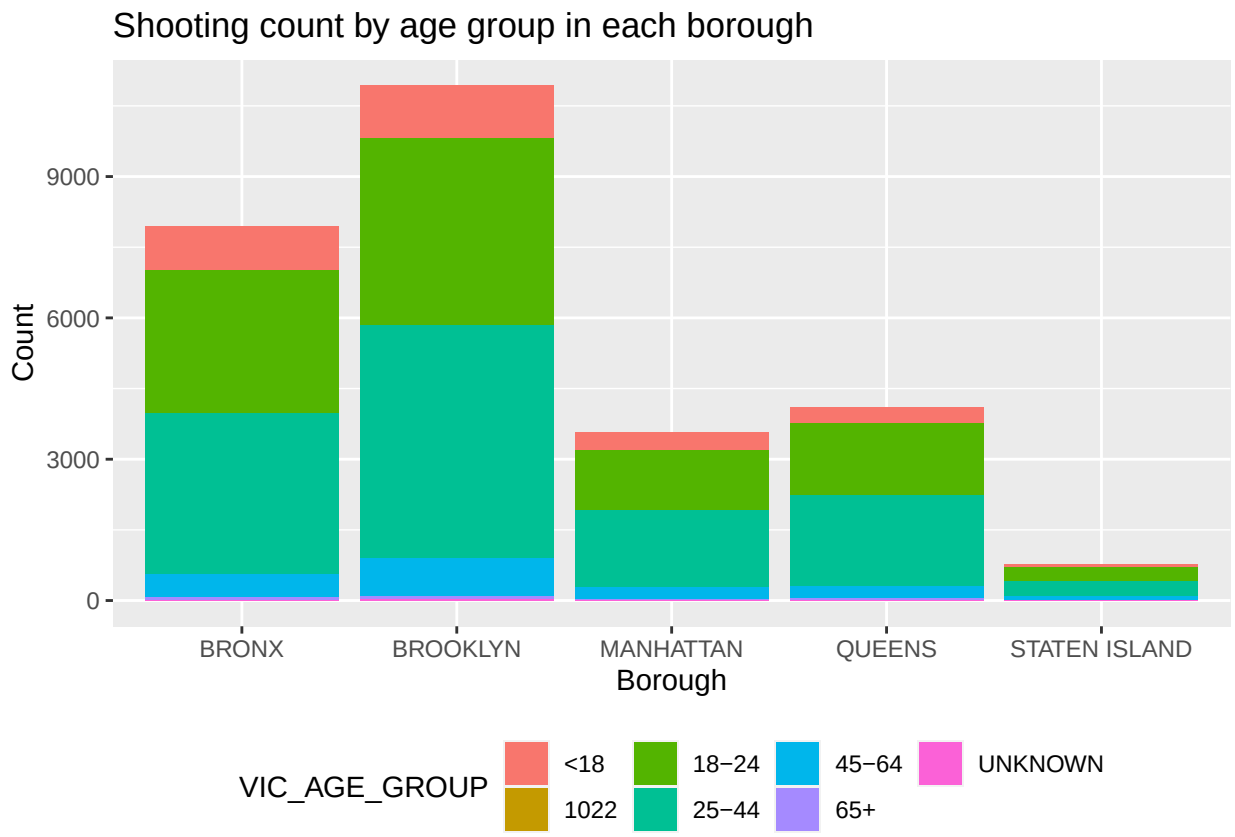
```
incidents_and_murders_by_boro <- data %>%
    group_by(BORO) %>%
    summarize(MURDERS = sum(STATISTICAL_MURDER_FLAG,na.rm='TRUE'),
              INCIDENTS = n()) %>%
    ungroup()
```

Next we visualize our data. To visualize the data with ggplot, we can use different types of plots depending on the purpose of our analysis. I chose Bar plot, Stacked bar plot and Grouped bar plot.

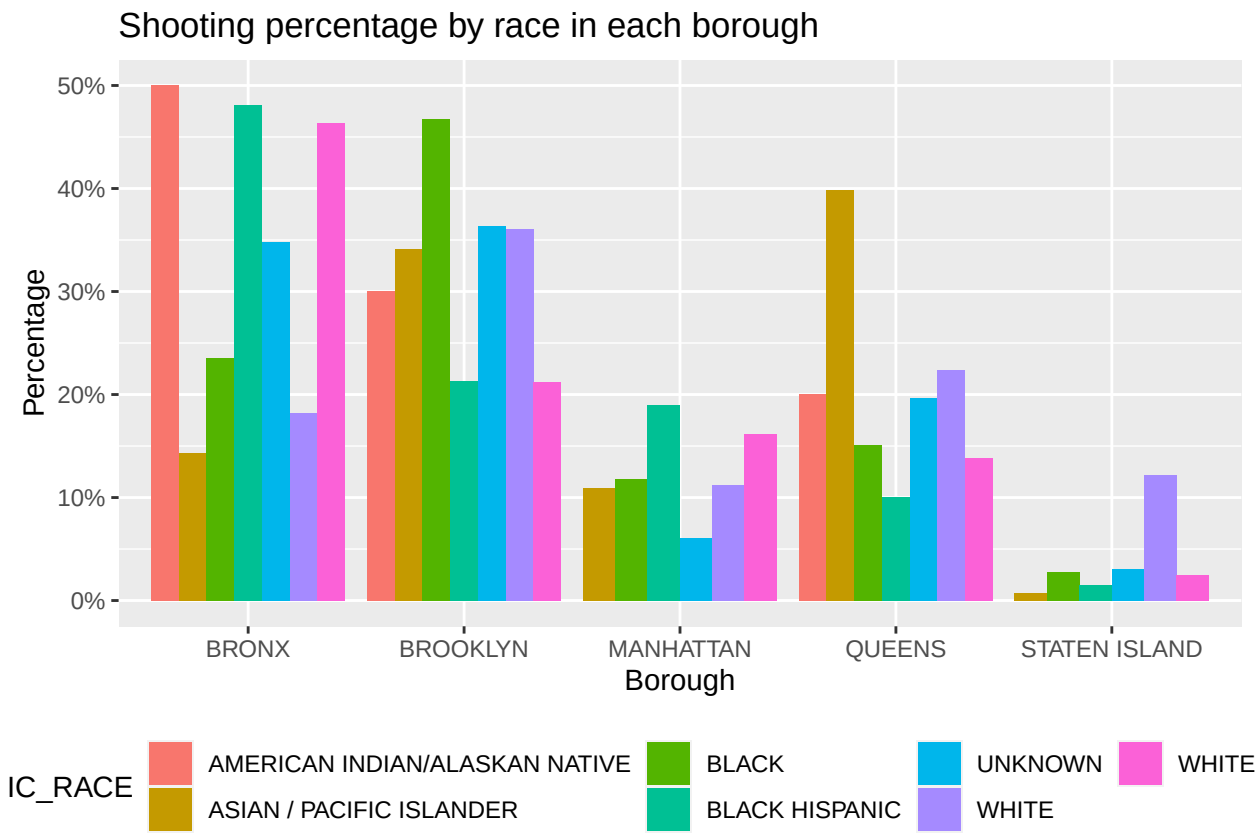
```
ggplot(shooting_in_boro_by_race, aes(x = BORO, y = COUNT, fill = VIC_RACE)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = COUNT), position = position_dodge(width = 0.9), vjust = -0.5, size = 2) +
  labs(title = "Shooting count by race in each borough", x = "Borough", y = "Count") +
  theme(legend.position = "bottom")
```



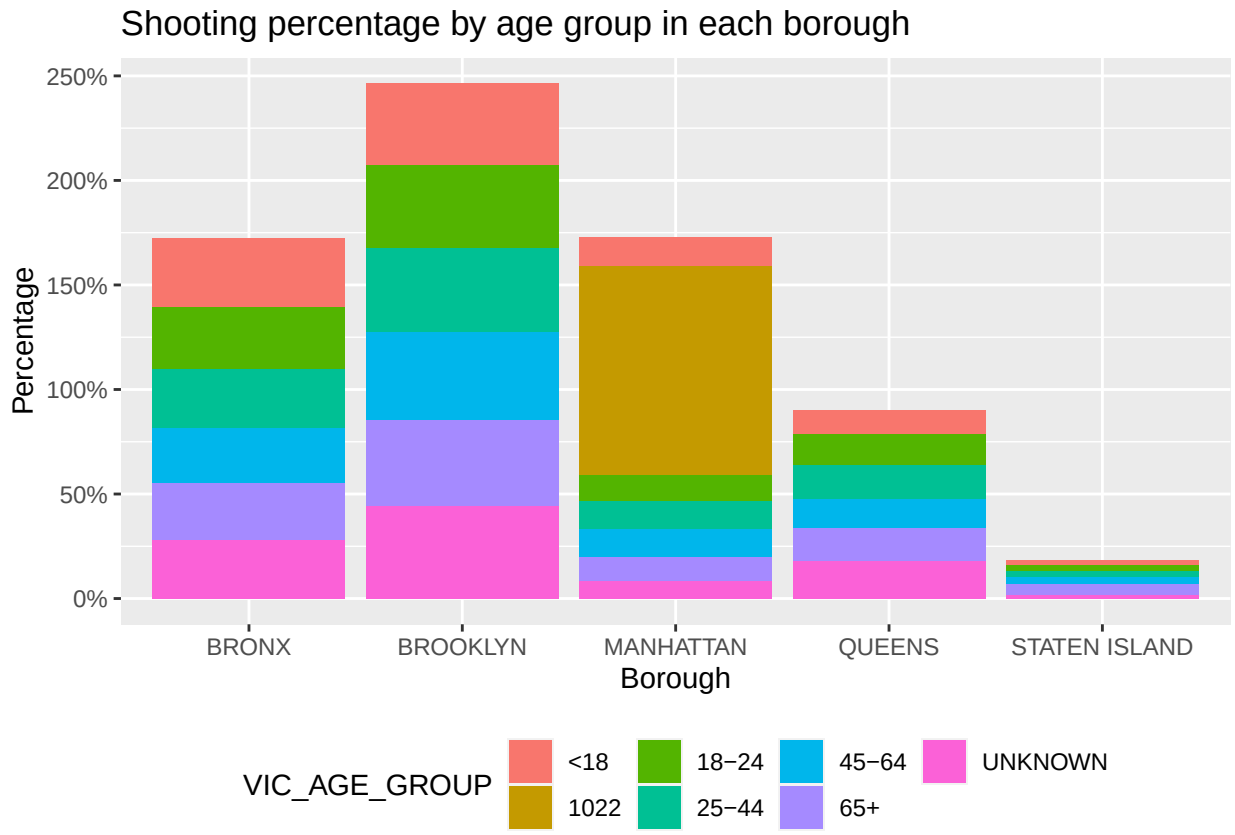

```
ggplot(shooting_in_boro_by_age_group, aes(x = BORO, y = COUNT, fill = VIC_AGE_GROUP)) +
  geom_bar(stat = "identity") +
  labs(title = "Shooting count by age group in each borough", x = "Borough", y = "Count") +
  theme(legend.position = "bottom")
```



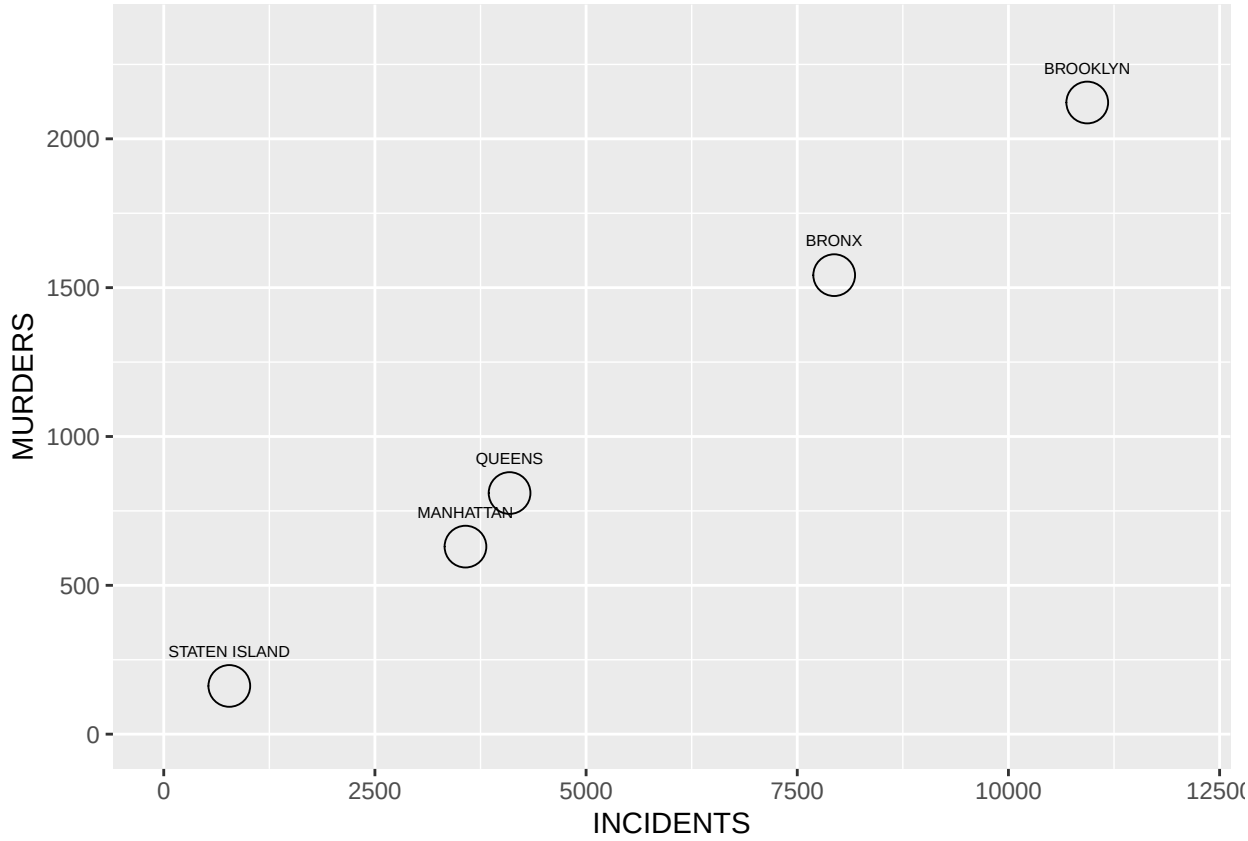

```
ggplot(shooting_in_boro_by_race, aes(x = BORO, y = PERCENTAGE, fill = VIC_RACE)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Shooting percentage by race in each borough", x = "Borough", y = "Percentage") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme(legend.position = "bottom")
```




```
ggplot(shooting_in_boro_by_age_group, aes(x = BORO, y = PERCENTAGE, fill = VIC_AGE_GROUP)) +
  geom_bar(stat = "identity") +
  labs(title = "Shooting percentage by age group in each borough", x = "Borough", y = "Percentage") +
  scale_y_continuous(labels = scales::percent_format()) +
  theme(legend.position = "bottom")
```




```
ggplot(incidents_and_murders_by_boro, aes(x = INCIDENTS, y = MURDERS, label = BORO)) +
  geom_point(size = 7, shape = 21) +
  geom_text(size = 2, vjust = -2.5, hjust = 0.5) +
  xlim(0, max(incidents_and_murders_by_boro$INCIDENTS) * 1.1) +
  ylim(0, max(incidents_and_murders_by_boro$MURDERS) * 1.1)
```

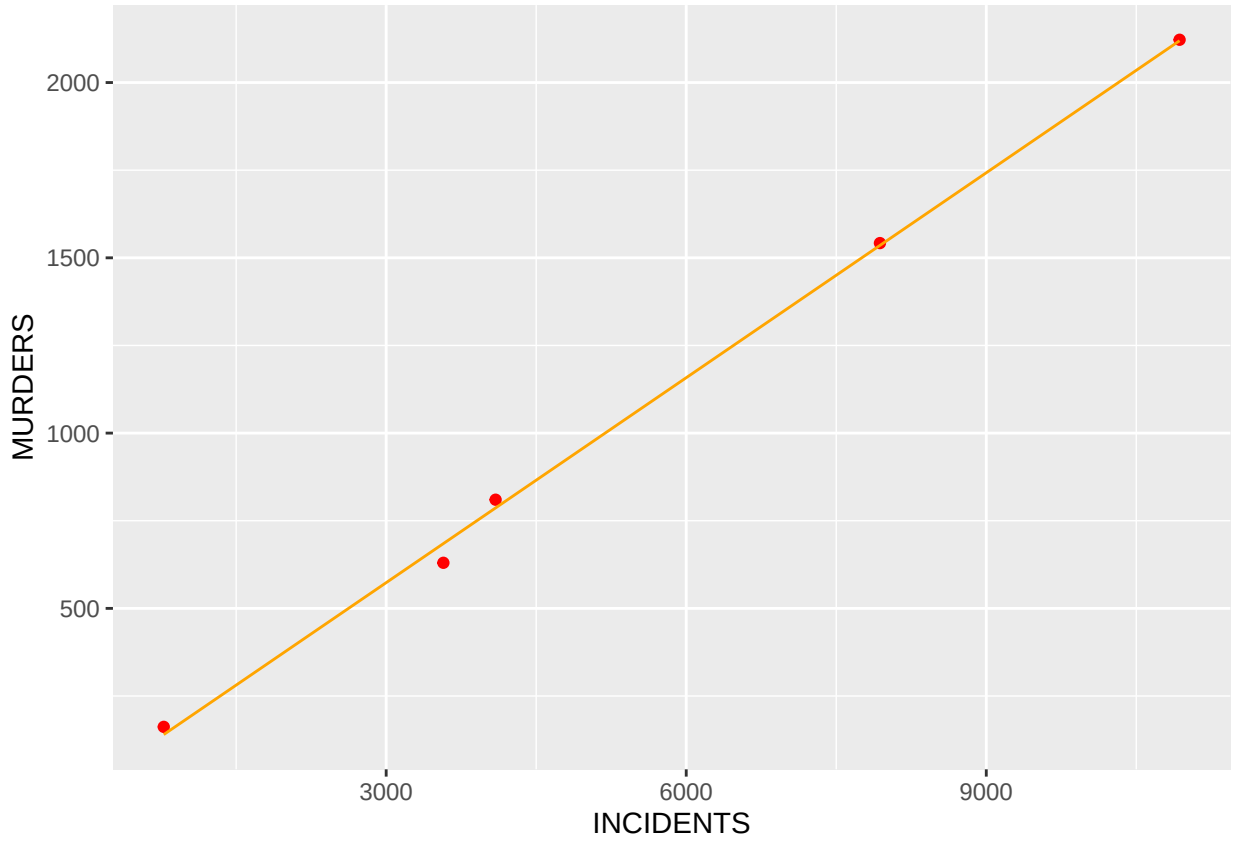


There is obvious strong linear relation between the shown variables. Here we try to build a linear model between INCIDENTS and MURDERS.

```
linear_model <- lm(MURDERS ~ INCIDENTS, data = incidents_and_murders_by_boro)
summary(linear_model)
```

```
##
## Call:
## lm(formula = MURDERS ~ INCIDENTS, data = incidents_and_murders_by_boro)
##
## Residuals:
##       1       2       3       4       5
##   6.593   2.785 -54.832  23.450  22.004
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.216935  30.233994  -0.371    0.735
## INCIDENTS     0.194863   0.004636  42.032 2.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.93 on 3 degrees of freedom
## Multiple R-squared:  0.9983, Adjusted R-squared:  0.9977
## F-statistic:  1767 on 1 and 3 DF,  p-value: 2.964e-05
```

```
prediction <-incidents_and_murders_by_boro%>%mutate(pred = predict(linear_model))
ggplot(prediction)+geom_point(aes(x=INCIDENTS,
y = MURDERS),color ="red")+
geom_line(aes(x=INCIDENTS,y=pred),color = "orange")
```



Our analysis shows us that the most criminal borough in 2006-2021 in New York City is Brooklyn. Unfortunately the most vulnerable race is Black and then Hispanic. People of age 25-44 and 45-64 have been killed more than others. Staten Island is the safest borough.

Obviously there are some reasons for such deviation like tough neighbourhood, tax policy, etc. which can also have impact. Of course for complex analysis it should be investigated carefully.

Conclusion

There are several sources of bias. For example we can have here Observer bias when those who collected the information could being biased toward certain situations. Or even Recall bias when they asked someone who didn't remember all of the important things.

I had personal bias as overconfidence for this dataset in R. Previously I worked with Python a lot. So I had to open some extra manuals to do that.

Session info

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggplot2_3.4.2 dplyr_1.1.1   readr_2.1.4
##
## loaded via a namespace (and not attached):
##  [1] highr_0.10       pillar_1.9.0     compiler_4.2.2   tools_4.2.2
##  [5] digest_0.6.31    bit_4.0.5        evaluate_0.20    lifecycle_1.0.3
##  [9] tibble_3.2.1     gtable_0.3.3     pkgconfig_2.0.3  rlang_1.1.0
## [13] cli_3.6.1        rstudioapi_0.14  curl_5.0.0       yaml_2.3.7
## [17] parallel_4.2.2   xfun_0.38        fastmap_1.1.1    withr_2.5.0
## [21] knitr_1.42       generics_0.1.3   vctrs_0.6.1      hms_1.1.3
## [25] bit64_4.0.5      grid_4.2.2       tidyselect_1.2.0 glue_1.6.2
## [29] R6_2.5.1         fansi_1.0.4      vroom_1.6.1      rmarkdown_2.21
## [33] farver_2.1.1     tzdb_0.3.0       magrittr_2.0.3   scales_1.2.1
## [37] htmltools_0.5.5  colorspace_2.1-0 labeling_0.4.2   utf8_1.2.3
## [41] munsell_0.5.0    crayon_1.5.2
```