NYCPD

Edouard Robeyns

2022-04-24

Introduction

This file is attempting to explore the NYPD Shooting Incident Data (Historic) available at https://catalog.data.gov/dataset

I will explore three aspects of the data:

1. **geographical**: are there areas more affected than others?
2. **demographical**: are there groups of people more affected or responsible than others?
3. **temporal**: does the frequency of incidents changes over time?

Library preparation

In order to access convenient functions, some packages must be imported.

```
# use tidyverse
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
# use lubridate
library(lubridate)
```

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

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

Import data

First let's import the data. While there is a single file to import, I will use the same method as displayed during the course, because I intend to keep this file as reference for later works. The main dataframe containing all the data will be named *nycpd*.

```
# save url
url_in <- "https://data.cityofnewyork.us/api/views/833y-fsy8/"
# save file name
file_names <- c("rows.csv")
# concate url and file name
urls <- str_c(url_in, file_names)
# download the file content into a dataframe
nycpd <- read_csv(urls[1])
```

```
## Rows: 23585 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): OCCUR_DATE, BORO, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_R...
## 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

I always start with a summary of the data, to get an idea of the number of lines/columns, and to start identifying which columns may be relevant to the chosen areas of inquiry.

```
#display the summary
summary(nycpd)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:23585       Length:23585      Length:23585
##  1st Qu.: 55322804   Class :character   Class1:hms        Class :character
##  Median : 83435362   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :102280741                      Mode  :numeric
##  3rd Qu.:150911774
##  Max.   :230611229
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.000     Length:23585       Mode :logical
##  1st Qu.: 44.00   1st Qu.:0.000     Class :character   FALSE:19085
##  Median : 69.00   Median :0.000     Mode  :character   TRUE :4500
##  Mean   : 66.21   Mean   :0.333
##  3rd Qu.: 81.00   3rd Qu.:0.000
##  Max.   :123.00   Max.   :2.000
##                   NA's   :2
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:23585       Length:23585       Length:23585       Length:23585
```

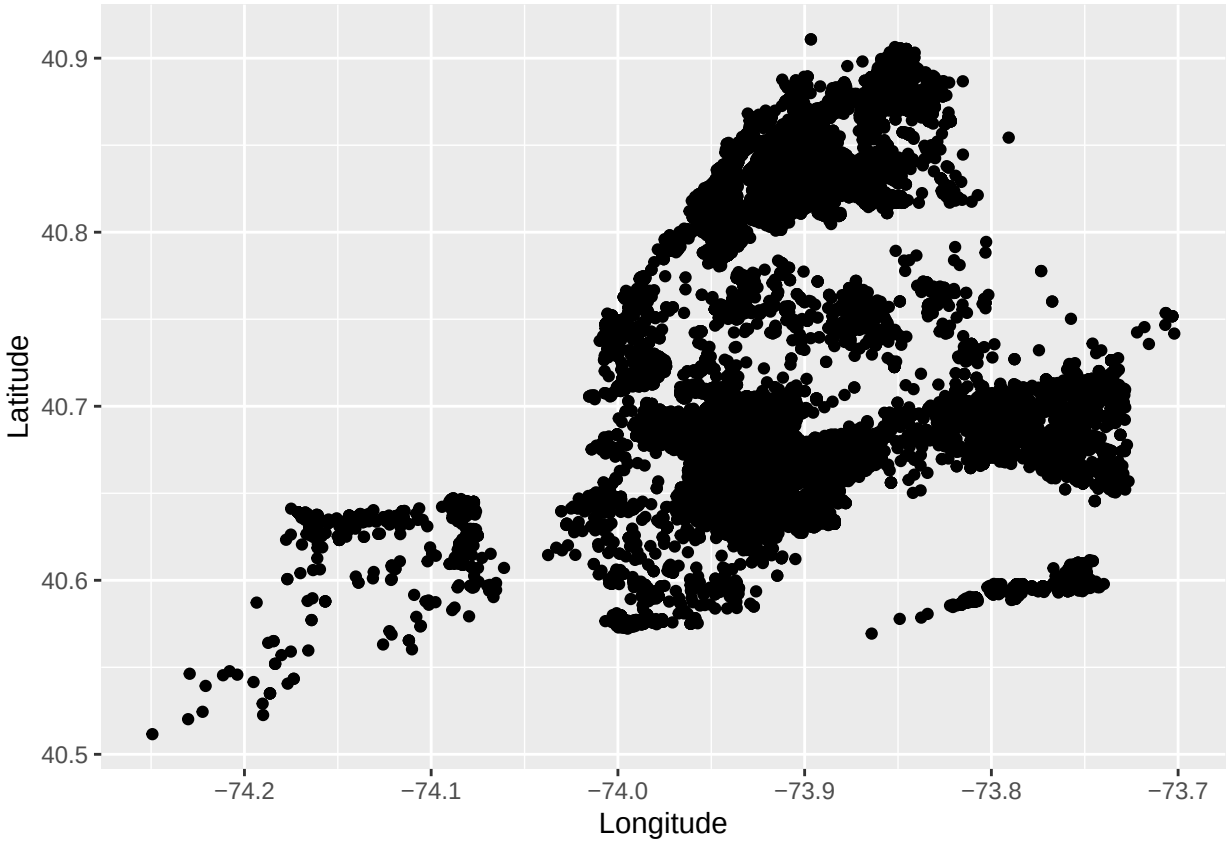
```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_SEX            VIC_RACE           X_COORD_CD        Y_COORD_CD
##  Length:23585       Length:23585       Min.   : 914928   Min.   :125757
##  Class :character   Class :character   1st Qu.: 999925   1st Qu.:182539
##  Mode  :character   Mode  :character   Median :1007654   Median :193470
##                                        Mean   :1009379   Mean   :207300
##                                        3rd Qu.:1016782   3rd Qu.:239163
##                                        Max.   :1066815   Max.   :271128
##
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:23585
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Median :40.70   Median :-73.92   Mode  :character
##  Mean   :40.74   Mean   :-73.91
##  3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :40.91   Max.   :-73.70
##
```

Analysis

Geographical data

First, I will start with the geographical data. Since **Latitude** and **Longitude** are already numbers, there is no need to do any transformation, and a first plot can be done quickly:

```
# display the incident by longitude and latitude
ggplot(data = nycpd, aes(x = Longitude, y = Latitude)) + geom_point()
```



When displaying this graph side by side with a map of New York city, I can clearly identify some regions like *Manhattan* or *Staten Island*. It seems to me *Bronx* and *Brooklyn* have the higher concentration of incidents.

Looking back at the summary, I see district names in the column named **BORO** (short for borough). Doing an aggregation first and then a descending sort confirms that there have been more incidents in *Brooklyn* and *Bronx*, while *Staten Island* has the least.

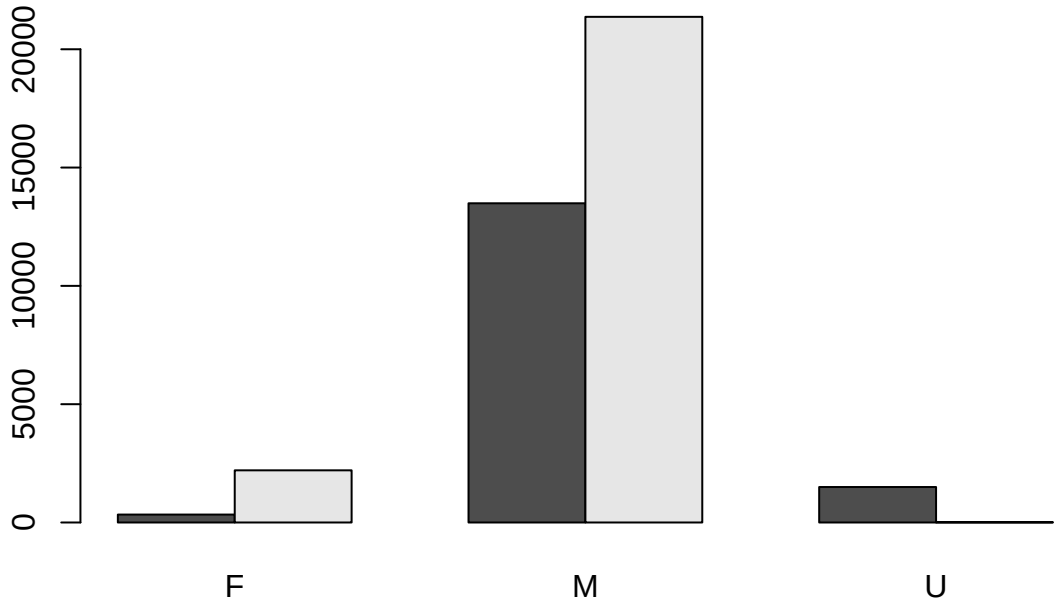
```
# aggregate borough data
boro_data <- as.data.frame(table(nycpd["BORO"]))
# sort borough data by descending frequency
boro_data[order(-boro_data$Freq),]
```

```
##            Var1 Freq
## 2      BROOKLYN 9734
## 1         BRONX 6701
## 4        QUEENS 3532
## 3     MANHATTAN 2922
## 5 STATEN ISLAND  696
```

Demographical data

Looking back at the summary, two columns contain the word *sex*: **PERP_SEX** and **VIC_SEX**. I first isolate them to aggregate the quantities separately, then concatenate the results to display them in a bar chart.

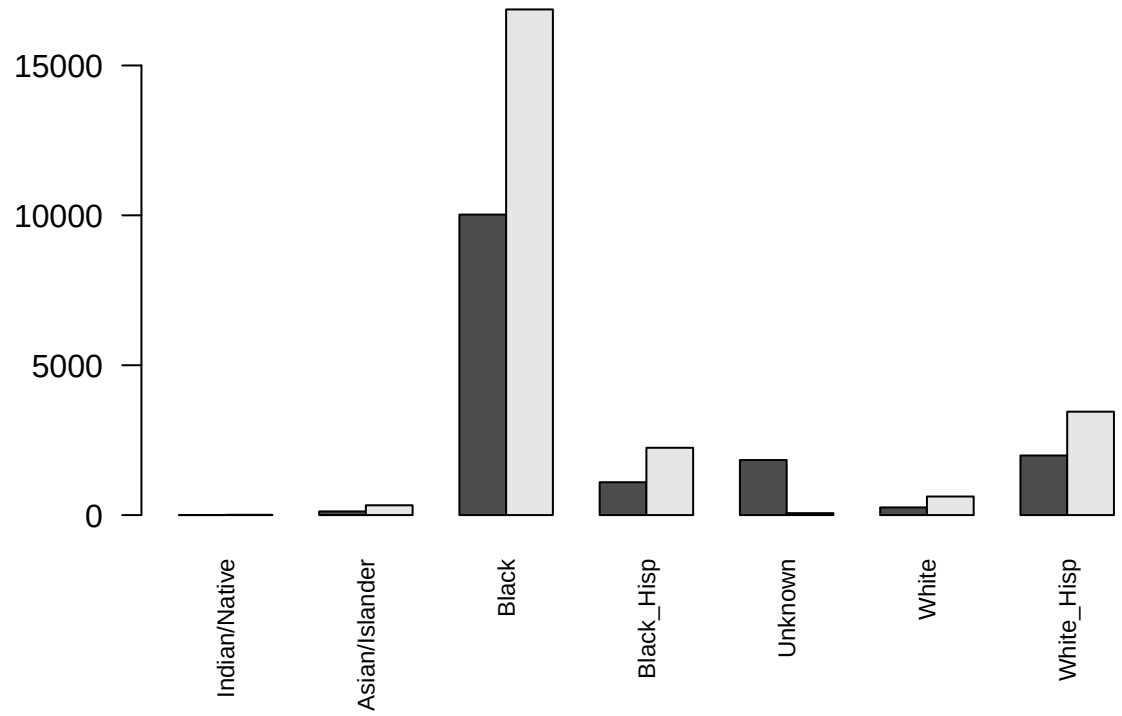
```r
# aggregate perpetrator sex data
perp_sex_data <- table(nycpd["PERP_SEX"])
# aggregate victim sex data
vic_sex_data <- table(nycpd["VIC_SEX"])
# concatenate both perpetrator and victim
sex_data <- rbind(perp_sex_data, vic_sex_data)
# display as a bar chart
barplot(sex_data, beside = TRUE)
```



The result shows that *males* both perpetuate and suffer more from shooting incidents than the other two categories combined (*female* and *unknown*).

Using the same methodology, create a similar chart for the *race*, using the columns **PERP_RACE** and **VIC_RACE**.

```r
# aggregate the perpetrator race data
perp_race_data <- table(nycpd["PERP_RACE"])
# aggregate the victim race data
vic_race_data <- table(nycpd["VIC_RACE"])
# concatenate perpetrator and victim
race_data <- rbind(perp_race_data, vic_race_data)
# rename columns to shorter version
colnames(race_data) <- c('Indian/Native','Asian/Islander','Black', 'Black_Hisp', 'Unknown', 'White', 'W
# display as a bar chart, turn the column names for readability
barplot(race_data, beside = TRUE, las=2, cex.names=.75)
```



Here, we can see that **Black** is the most represented category for both *perpetrator* and *victim* groups.

Finally, I will display a bar chart for the *age group*, using the columns **PERP_AGE_GROUP** and **VIC_AGE_GROUP**.

Here it was necessary to clean the data first, otherwise I could not recombine *perpetrator* and *victim* data back together. First, I replaced all occurrences of **<NA>** to **UNKNOWN**. Then I changes the special cases for **1020**, **940** and **224** to **UNKNOWN** as well. While those were probably typos, I decided better to add them to the **UNKNOWN** category rather than guessing what the original data was. Given the low number, they probably wouldn't affect the outcome in any significant way regardless.

```
# display list of unique perpetrator age group
unique(nycpd["PERP_AGE_GROUP"])
```

```
## # A tibble: 10 x 1
##    PERP_AGE_GROUP
##    <chr>
##  1 <NA>
##  2 18-24
##  3 UNKNOWN
##  4 25-44
##  5 <18
##  6 45-64
##  7 65+
##  8 1020
##  9 940
## 10 224
```

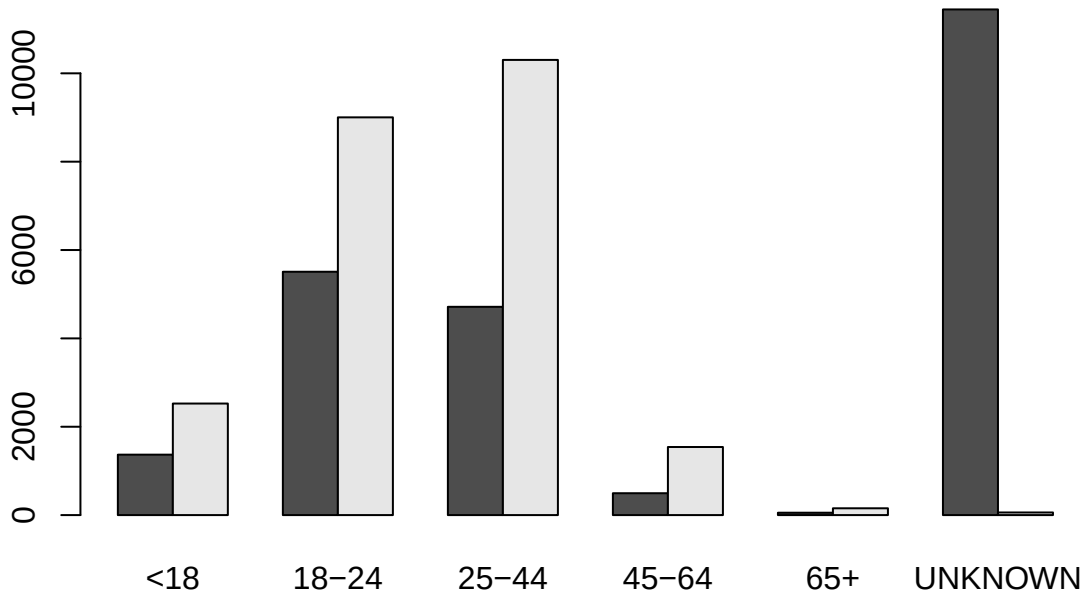
```r
# display list of unqiue victim age group
unique(nycpd["VIC_AGE_GROUP"])
```

```
## # A tibble: 6 x 1
##   VIC_AGE_GROUP
##   <chr>
## 1 25-44
## 2 65+
## 3 18-24
## 4 <18
## 5 45-64
## 6 UNKNOWN
```

```r
# the perpetrator has four more age group than the victim, preventing concatenation
# I will manually fix that below

# isolate perpetrator age group
perp_age_group_list <- nycpd["PERP_AGE_GROUP"]
# replace <NA> by UNKNOWN
perp_age_group_list[is.na(perp_age_group_list)] <- "UNKNOWN"
# replace 1020 by UNKNOWN
perp_age_group_list$PERP_AGE_GROUP[perp_age_group_list$PERP_AGE_GROUP == "1020"] <- "UNKNOWN"
# replace 940 by UNKNOWN
perp_age_group_list$PERP_AGE_GROUP[perp_age_group_list$PERP_AGE_GROUP == "940"] <- "UNKNOWN"
# replace 224 by UNKNOWN
perp_age_group_list$PERP_AGE_GROUP[perp_age_group_list$PERP_AGE_GROUP == "224"] <- "UNKNOWN"

# aggregate perpetrator age group
perp_age_group_data <- table(perp_age_group_list)
# aggregate victim age group
vic_age_group_data <- table(nycpd["VIC_AGE_GROUP"])
# concatenate perpetrator and victim
age_group_data <- rbind(perp_age_group_data, vic_age_group_data)
# display as a bar chart
barplot(age_group_data, beside = TRUE)
```



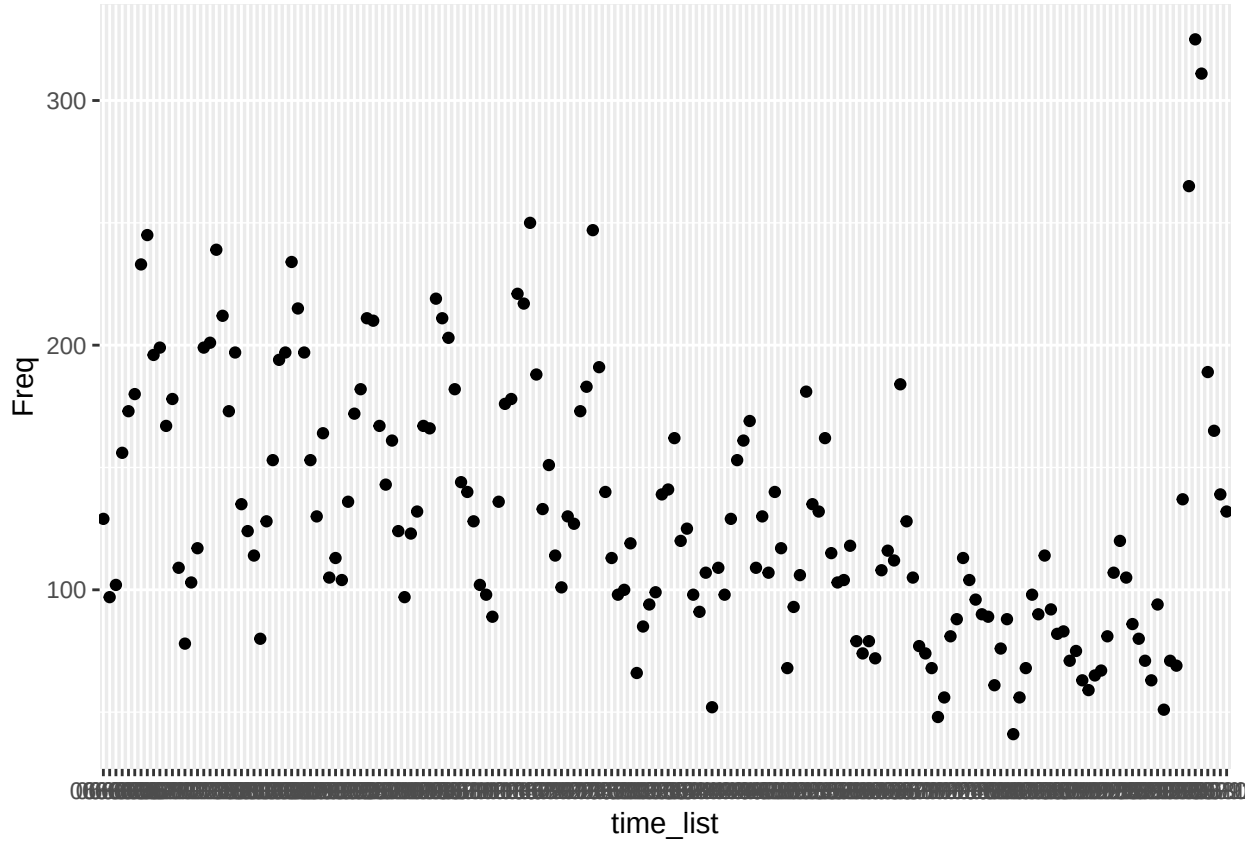
As opposed to the last two graphs, the data for perpetrator and victim group do not match. The **UNKNOWN** in the perpetrator group nearly equals all other categories combined, while the data for the victim group present a more regular bell curve.

Also of note, the average perpetrator is younger (18-24 category) while the victim is older (25-44 category).

Temporal data

In order to display a readable graph about more than 23 thousands incidents, I believe it makes sense to aggreate the numbers by month. Below, I isolate the **OCCUR_DATE** column, then change the format to *year-month*, then do the aggregation.

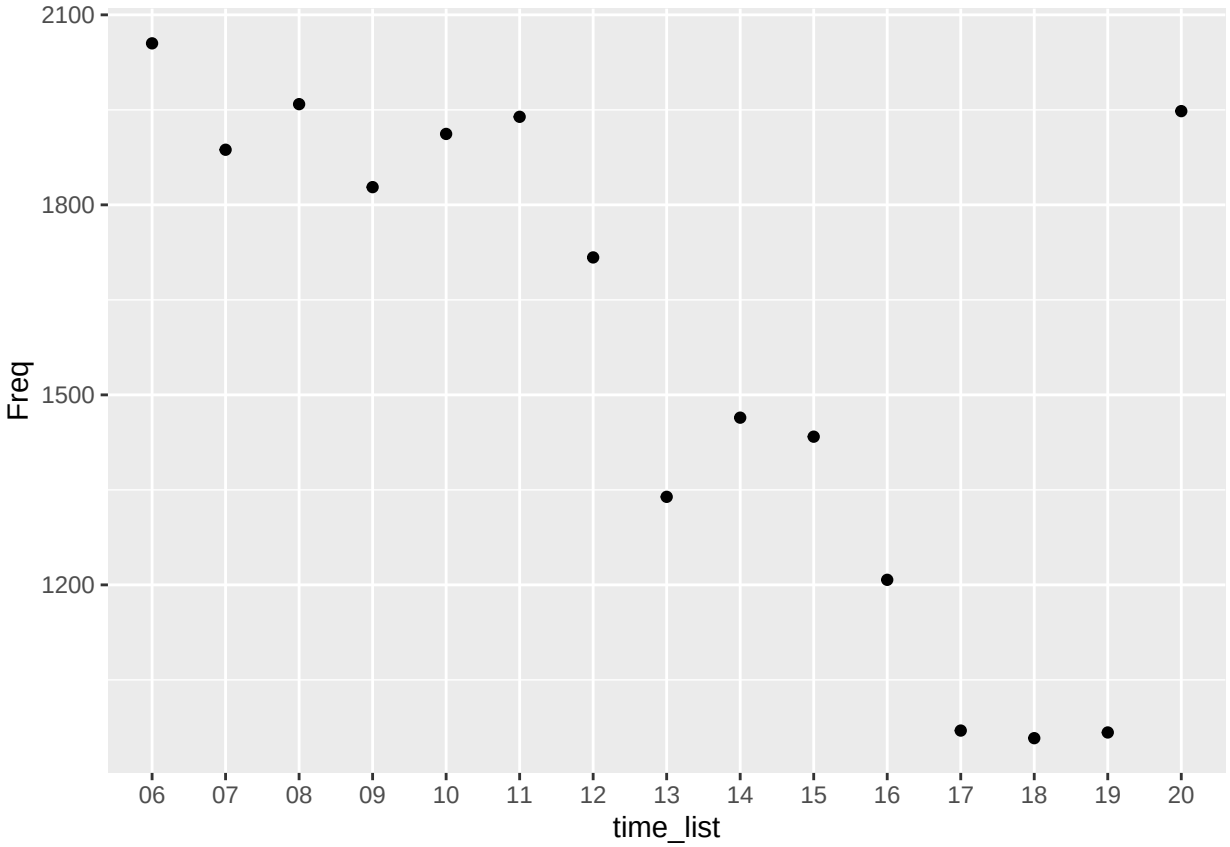
```
# isolate dates of incidents
time_list <- nycpd["OCCUR_DATE"]
# change format from character to date
time_list <- time_list %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE))
# remove the day to reduce granularity
time_list$OCCUR_DATE <- format(time_list$OCCUR_DATE,'%y-%m')
# aggregate incidents by month
time_by_month_data <- as.data.frame(table(time_list))
# display number of incidents by month
ggplot(data = time_by_month_data, aes(x = time_list, y = Freq)) + geom_point()
```



While a tendency (down) can be identified, I wanted to see if it would be more defined if the aggregation was by year.

The below graph uses the same sequence of data transformation but aggregate by year.

```
# isolate dates of incidents
time_list <- nycpd["OCCUR_DATE"]
# change format from character to date
time_list <- time_list %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE))
# remove the month and the day to reduce granularity furhter
time_list$OCCUR_DATE <- format(time_list$OCCUR_DATE,'%y')
# aggregate incidents by year
time_by_year_data <- as.data.frame(table(time_list))
# display number of incidents by year
ggplot(data = time_by_year_data, aes(x = time_list, y = Freq)) + geom_point()
```



Here the down tendency is more clearly visible.

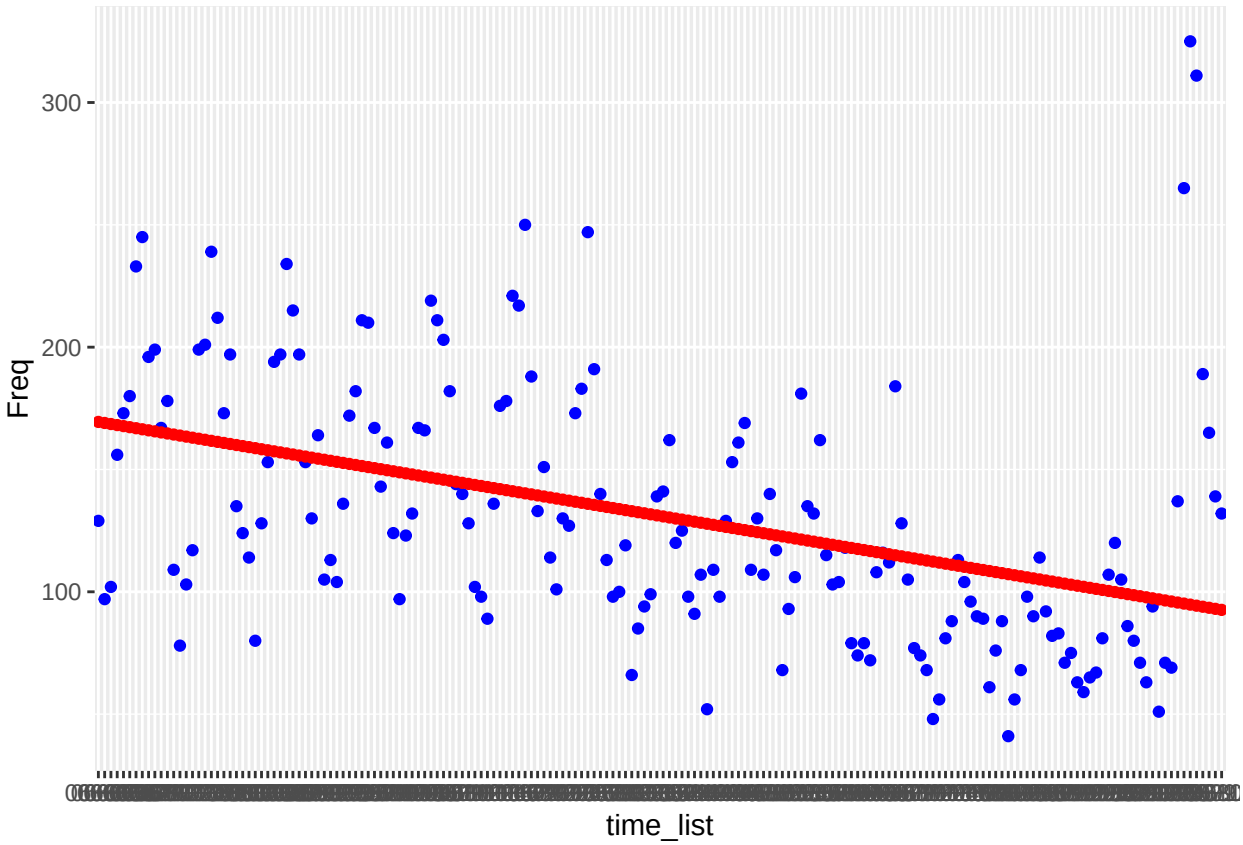
Of note is the last year of the graph, nearly as high as the first year and not following the overall trend of the graph. This is a clear outsider.

Model

Model by month prediction

Using a linear model regression, we can easily update the incident per graph to display a simple prediction.

```
# create model using linear model function
# using as.numeric to transform time_list is absolutely essential
mod <- lm(Freq ~ as.numeric(time_list), data=time_by_month_data)
# generate the prediction
time_by_month_with_model <- time_by_month_data %>% mutate(pred = predict(mod))
# plot the data with the prediction, original is blue, prediction is red
time_by_month_with_model %>% ggplot() + geom_point(aes(x = time_list, y = Freq), color = "blue") + geom
```

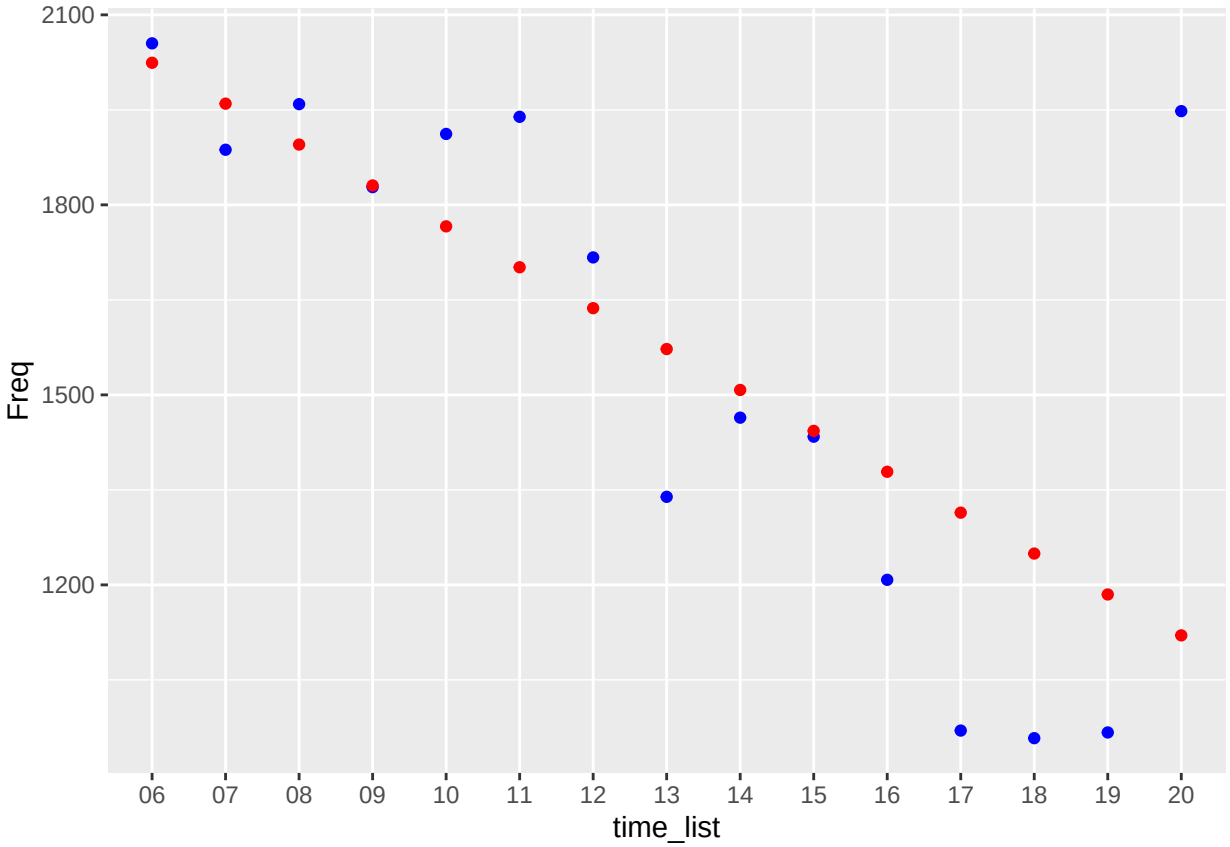


As expected, the overall tendency is down, making the last months clear outliers.

Model by year prediction

Likewise, we update the graph of the yearly count to add a simple linear model.

```
# create model using linear model function
# using as.numeric to transform time_list is absolutely essential
mod <- lm(Freq ~ as.numeric(time_list), data=time_by_year_data)
# generate the prediction
time_by_year_with_model <- time_by_year_data %>% mutate(pred = predict(mod))
# plot the data with the prediction, original is blue, prediction is red
time_by_year_with_model %>% ggplot() + geom_point(aes(x = time_list, y = Freq), color = "blue") + geom_
```



Again, the down tendency is even more noticeable.

Bias

Potential Bias from the sources

Given the initial data set is a record of *reported* incidents, the first and greatest bias is that not all incidents are reported. Furthermore, it is unlikely for the missing incidents to be uniformly missing. In other words, it is not feasible to expect the missing data to either enhance and diminish the observable tendencies.

The lack of density data pertaining to the geography or demography prevents the making of some conclusions. For example, is there twice as much incident in Brooklyn than Manhattan because there are twice as many people living there? So, while the conclusion comes from the data, it may disproportionately vilify a particular group, even though the ratio of incidents may not be different that any other group.

Potential Bias from the author

One obvious bias from me was to re-categorize **<NA>** as **UNKNOWN**. An alternative would have been to add the **<NA>** category to the *victim* group.

Another bias are the chosen areas of focus of this file: geographical, demographical and temporal.

Closing thoughts

Conclusion

While not in-depth, the following tendencies can be extracted from the analysis made in this file:

1. most incidents appear in Brooklyn or Bronx boroughs
2. the average perpetrator is a young black man, while the average victim is a middle-aged black man
3. while the latest months of the file display an increase of incidents, the overall tendency is a decrease of incidents over the period represented

Session info

Here are the session info.

```
# display session info
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## 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] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.8
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2 xfun_0.30        haven_2.4.3      colorspace_2.0-3
##  [5] vctrs_0.4.0      generics_0.1.2   htmltools_0.5.2  yaml_2.3.5
##  [9] utf8_1.2.2       rlang_1.0.2      pillar_1.7.0     glue_1.6.2
## [13] withr_2.5.0      DBI_1.1.2        bit64_4.0.5      dbplyr_2.1.1
## [17] modelr_0.1.8     readxl_1.4.0     lifecycle_1.0.1  munsell_0.5.0
## [21] gtable_0.3.0     cellranger_1.1.0 rvest_1.0.2      evaluate_0.15
## [25] labeling_0.4.2   knitr_1.38       tzdb_0.3.0       fastmap_1.1.0
## [29] curl_4.3.2       parallel_4.1.3   fansi_1.0.3      highr_0.9
## [33] broom_0.8.0      backports_1.4.1  scales_1.2.0     vroom_1.5.7
## [37] jsonlite_1.8.0   farver_2.1.0     bit_4.0.4        fs_1.5.2
## [41] hms_1.1.1        digest_0.6.29    stringi_1.7.6    grid_4.1.3
## [45] cli_3.2.0        tools_4.1.3      magrittr_2.0.3   crayon_1.5.1
## [49] pkgconfig_2.0.3  ellipsis_0.3.2   xml2_1.3.3       reprex_2.0.1
```

```
## [53] assertthat_0.2.1 rmarkdown_2.13   httr_1.4.2       rstudioapi_0.13
## [57] R6_2.5.1         compiler_4.1.3
```