NYPD Shooting Incident

M. Marrakchi

4/10/2022

First, installing the following packages the "tidyverse", "lubridate" and "formattable" and then run the the libraries related to those packages

```
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   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

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

```
library (lubridate)
```

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

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

```
library (formattable)
```

Background and Objectives

The historical information regarding the shooting incidents in NYC is shared on the website DATA.ORG and could be accessed at this link https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic. The objective of this project is the review and analysis of this dataset. In order to do so the below steps will be followed:
1st step: Describe and import the dataset in a reproducible manner,
2nd step Add to the document a summary of the data and clean up the dataset by changing appropriate variables to factor and date types and getting rid of any columns not needed. And through the summary function check the existence of missing data, describe how this missing data will be handled,
3rd step Add different visualizations and analysis to the project.
4th step Write the conclusion to the project report and include any possible sources of bias.

1st step - describe and import data:

The *NYPD Incident Incident Data (Historic)* is important from the DATA.ORG website

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

```
## # A tibble: 6 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
## # ... 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>
```

This dataset contains 19 variables as shown in the table above

2nd step - adding the summary and cleaning the data:

- summary of the data is displayed in order to read the review the content of the data

```
summary(NYPD)
```

```
##   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
##
```

Possible bias

1- Expecting that the majority of the cases are in the Bronx,
2- The highest number of the incident are committed by the age group 18-24

For this project, we will analyze the following information:
- shooting per BORO (town),
- shooting per perpetual age group, and
- shooting incident vs. death cases

The unnecessary variables will be removed from the data set, i.e., INCIDENT_KEY, OCCUR_TIME, PRECINCT, JURISDICTION_CODE, LOCATION_DESC, PERP_SEX, VIC_AGE_GROUP,PERP_RACE , VIC_SEX, VIC_RACE, X_COORD_CD, Y_COORD_CD, Latitude, Longitude and Lon_Lat.

```
NYPD<- NYPD %>% select (-c(INCIDENT_KEY, OCCUR_TIME,  PRECINCT, JURISDICTION_CODE, LOCATION_DESC, PERP_S
```

the remaining contains four variables, i.e., OCCUR_DATE, BORO, PERP_AGE_GROUP and STATISTICAL_MURDER_FLAG, all the remaining values have a character type, we transform the OCCUR_DATE date type and the BORO to factor type and change the name of the header of the Variable to Date, Town, tat_murder_flag, and Perp_age_group.

```
NYPD$BORO <- factor(NYPD$BORO )
NYPD$OCCUR_DATE = as.Date(NYPD$OCCUR_DATE, format = "%m/%d/%Y")
names (NYPD) <- c("Date", "Town", "Stat_murder_flag" , "Perp_age_group")
```

The cleaned dataset contains 23585 shooting incidents occurred between 2006-01-01 and 2020-12-31
to keep the number of the incident updated that function was used "r nrow (NYPD)" and for the dates those functions were used r min and max of the Date column.

There is missing data in "Perp_age_group" of, 35.17%.
to keep the percentages updated that functions was used "r round (100 sum(is.na(NYPD$tPerp_age_group))/nrow (NYPD),2)"*
As the missing data represent around 35%, the analysis of the number of shooting age groups will be carried out based on the remaining 65%.

3nd step - Visualization and Analysis of the shooting incident

A- Number of shooting per town

- ***We will analyse the shooting and death occurrence per Town.***
  Shooting incidents

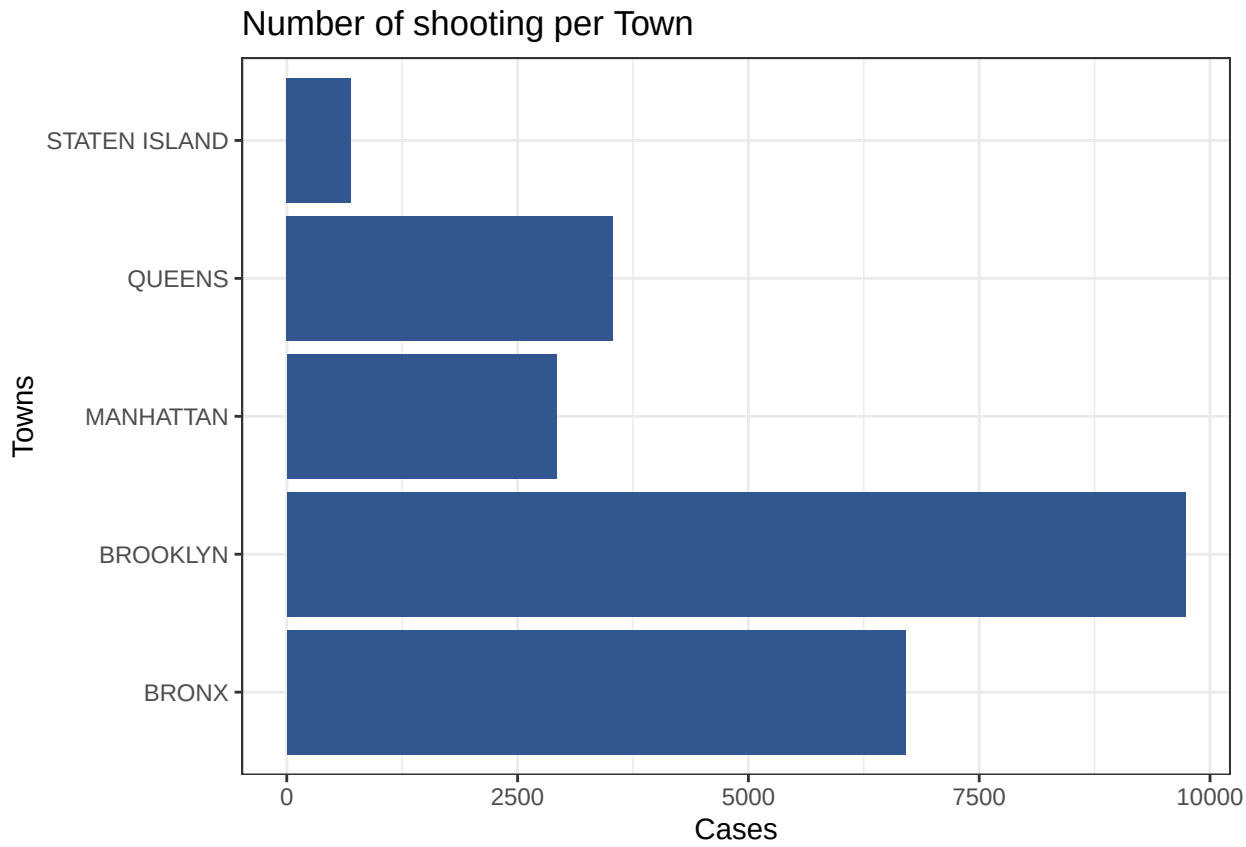
```
# we will group by Town
NYPD_by_town <- NYPD %>% group_by(Town) %>% summarize(Cases = n(), Deaths = sum (Stat_murder_flag)) %>%
NYPD_by_town
```

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

```
# get the towns with the highest and lowest number of shootings
NYPD_by_town %>%  slice_max(Cases, n=1)
```

```
## # A tibble: 1 x 3
##   Town     Cases Deaths
##   <fct>    <int>  <int>
## 1 BROOKLYN  9734   1898
```

```
#Creating a histogram for the number of shootings per weekday.
 NYPD_by_town %>% ggplot(aes(x= Town, y = Cases)) + geom_col (fill = "#325790") + coord_flip()  +  theme
```

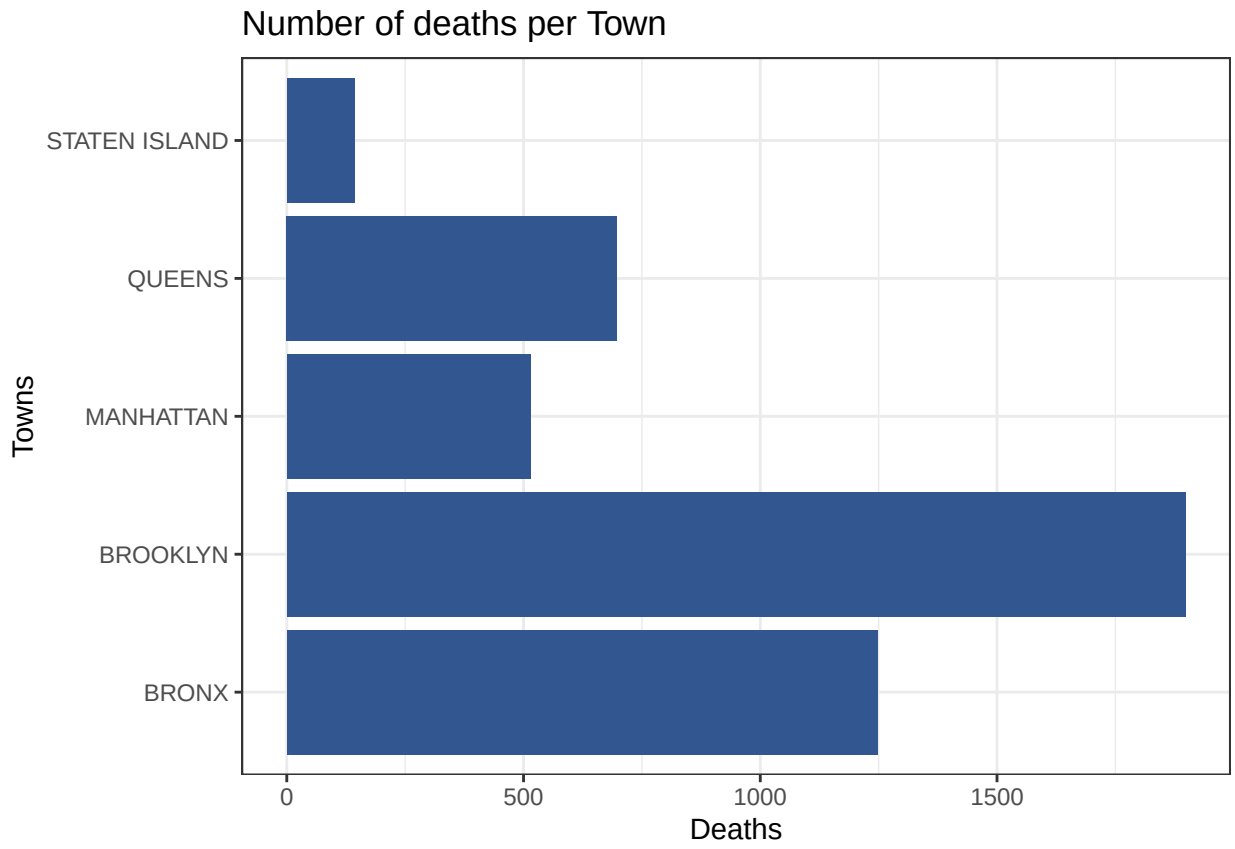


Deaths

```
# get the towns with the highest and lowest number of shootings
NYPD_by_town %>%  slice_max(Deaths, n=1)
```

```
## # A tibble: 1 x 3
##   Town     Cases Deaths
##   <fct>    <int>  <int>
## 1 BROOKLYN  9734   1898
```

```
#Creating a histogram for the number of shootings per weekday.
 NYPD_by_town %>% ggplot(aes(x= Town, y = Deaths)) + geom_col (fill = "#325790") + coord_flip()  +  ther
```

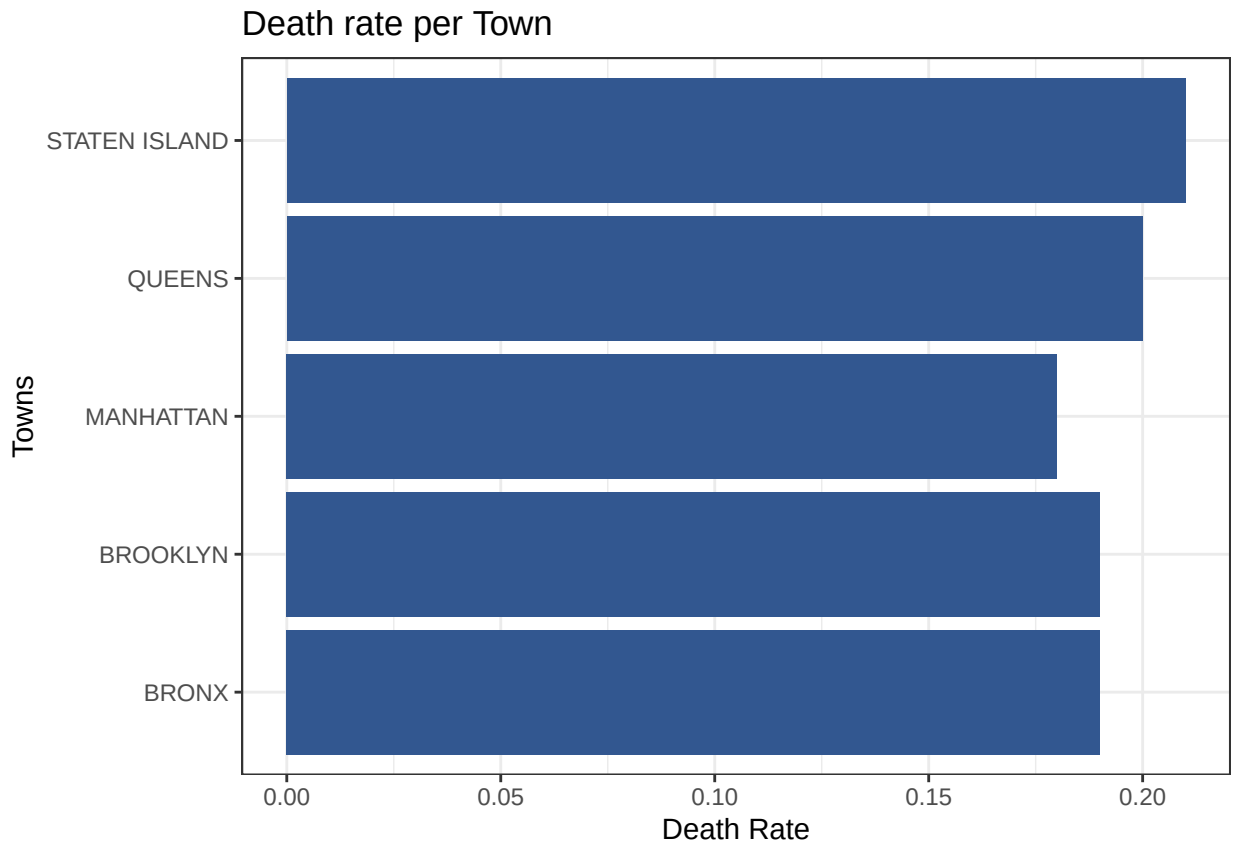



```
#we will group by town and calculate the number of cases, deaths and ration Death / Cases
NYPD_death_rate_by_town <- NYPD %>% group_by(Town) %>% summarize (Cases = n(), Deaths = sum (Stat_murde
NYPD_death_rate_by_town
```

- We will analyze the death rate per shooting per town

```
## # A tibble: 5 x 4
##   Town          Cases Deaths Death_Rate
##   <fct>         <int>  <int>      <dbl>
## 1 BRONX          6701   1247       0.19
## 2 BROOKLYN       9734   1898       0.19
## 3 MANHATTAN      2922    515       0.18
## 4 QUEENS         3532    697       0.2
## 5 STATEN ISLAND   696    143       0.21
```

```
#Creating a histogram  for the number death rate per town.
NYPD_death_rate_by_town %>% ggplot(aes(x= Town, y = Death_Rate)) + geom_col (fill = "#325790") + coord_
```



Conclusion on the shooting and the rate by town.

The highest number of shootings occurred was in Brooklyn and the lowest one was on Staten Island. The dataset was missing the population per town; therefore, it was not possible to compare the shooting rate per capita which would provide more information about the level of shooting risks in each of the five towns.

The second comparison was based on the death rate per shooting for each town, it shows that the rates are around 20 %, the highest one was in Staten Island with a rate of 21% and the lowest was in Manhattan with a rate of 18%.

B- Number of shooting per perpetual age group

```
#we will group by perp age group and calculate the cases
NYPD_death_per_perp_age <- NYPD %>% filter (Perp_age_group == "<18" | Perp_age_group == "18-24" | Perp_a
NYPD_death_per_perp_age <- NYPD_death_per_perp_age %>% mutate (Percentage = percent (Cases / sum(NYPD_d
NYPD_death_per_perp_age
```

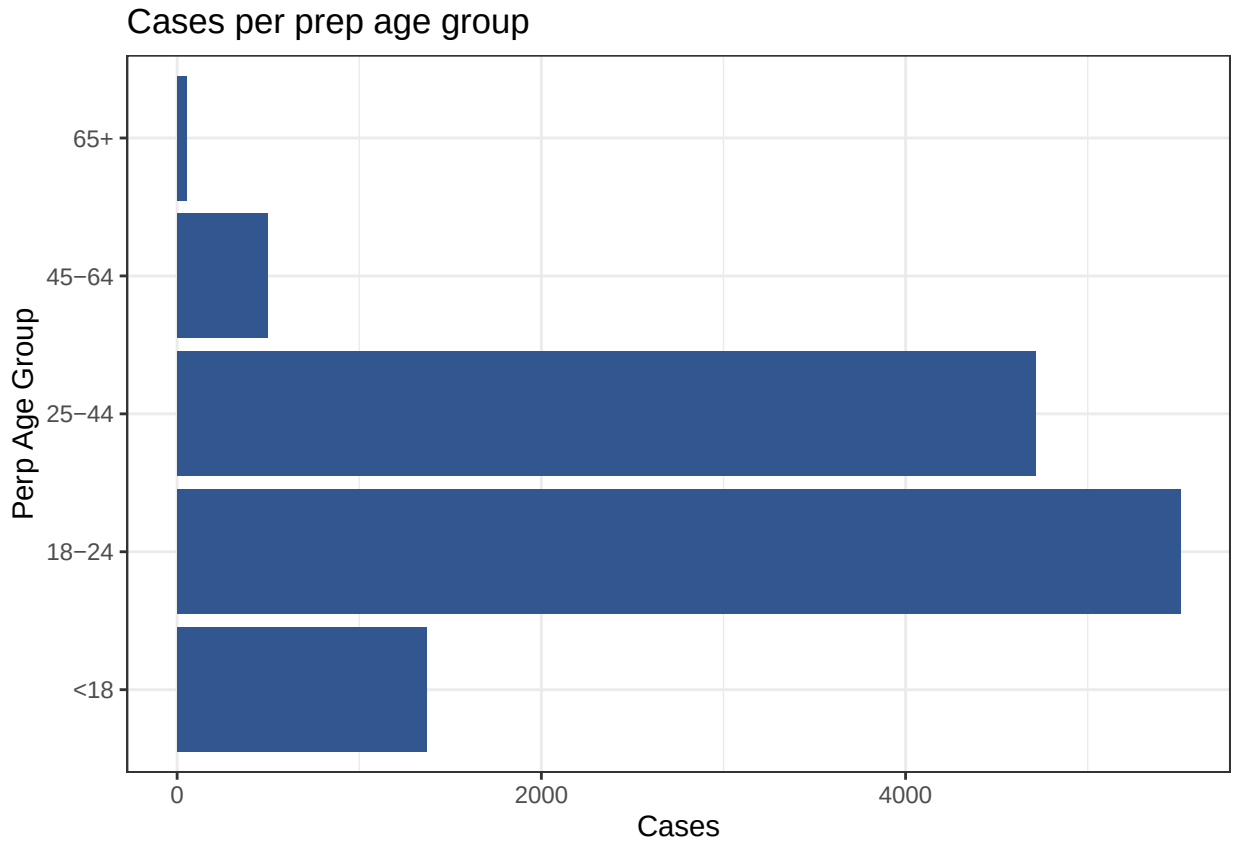
- We will analyse the number of shooting per perpetual age group

```
## # A tibble: 5 x 3
##   Perp_age_group Cases Percentage
```

```
##   <chr>          <int> <formttbl>
## 1 <18             1368 11%
## 2 18-24           5508 45%
## 3 25-44           4714 39%
## 4 45-64            495 4%
## 5 65+               54 0%
```

```
#Creating a histogram  for the number of cases per prep age group.

NYPD_death_per_perp_age %>% ggplot(aes(Perp_age_group , Cases)) + geom_col(fill = "#325790") + coord_fli
```



- Conclusion the number of shooting per perpetual age group

As predicted the highest number of incident per perpetual age group was for those between 18-24, which were 5508 cases representing 45% of the total cases.

C- Modeling

Now we will check if there is a relationship between the number of shooting and the number of death, for this we will build a linear model between those two variables from 2006 to 2020

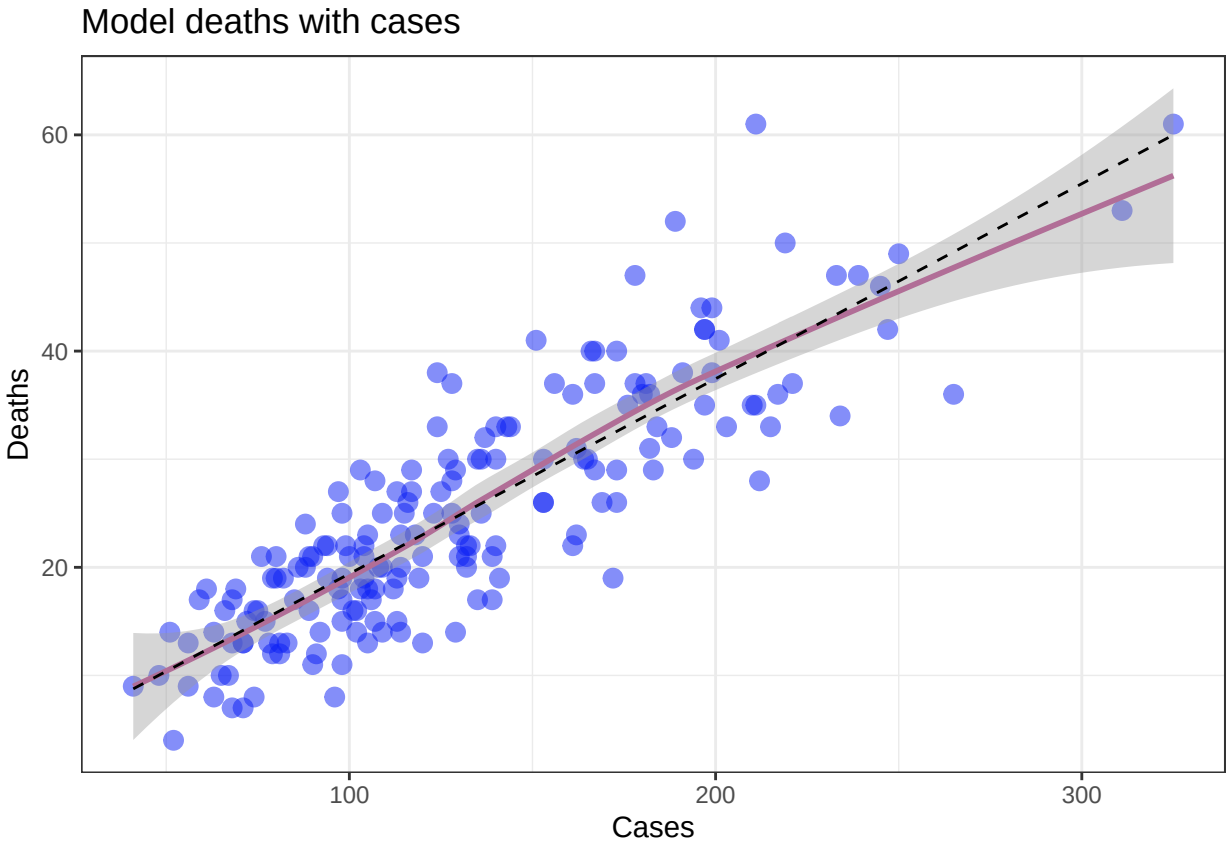
```
#first we need to create a subdata that shows the cases and death per day
NYPD_death_rate_by_month <- NYPD %>% mutate (Month = format (as.Date (NYPD$Date), "%Y/%m")) %>% group_by
# creat the model
```

```
mod <- lm(Deaths ~ Cases , data = NYPD_death_rate_by_month)
summary(mod)
```

```
##
## Call:
## lm(formula = Deaths ~ Cases, data = NYPD_death_rate_by_month)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3916  -3.8531  -0.0315   3.5552  21.5726
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.36199    1.13586   1.199    0.232
## Cases        0.18041    0.00804  22.438   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.697 on 178 degrees of freedom
## Multiple R-squared:  0.7388, Adjusted R-squared:  0.7373
## F-statistic: 503.5 on 1 and 178 DF,  p-value: < 2.2e-16
```

```
# predict number of deaths based on the model
pred <- tibble (pred = predict(mod))
NYPD_death_rate_by_month_pred <- cbind(NYPD_death_rate_by_month,pred)
#Plot the prediction with actual cases and deaths
NYPD_death_rate_by_month_pred %>% ggplot () + geom_point(aes(x=Cases, y=Deaths), size = 3, color = "#0A
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



- ***Conclusion***

  We could assume that there is a linear relationship between the number of cases ad the number of deaths