

NYPD Shooting Incidents Report

5/20/2023

Problem Statement:

New York City is one of the biggest cities in the US, it attracts millions of visitors each year, gun violence concerns many of the city visitors, shooting incidents and deaths are increasing at an alarming rate recently. Starting 2020 the gun violence has increased dramatically, what is the reason for this increase? Is it happening at a specific time of the day? Is it happening more in some boroughs or is it happening at the same rate across multiple ones? Is it happening to a certain age group? How are these shootings leading to deaths? I'm going to investigate some of the questions in the analysis below.

Questions of Interest:

- Incidents distribution by year.
- Incidents distribution by time.
- Are these incidents happening to a certain age group.
- Safest hours to go out.
- Borough with the highest shooting incidents.

This following goes through the NYPD Shooting Incidents csv data that is available in <https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD>

Before we start:

Please note that this project uses the package tidyverse, if it's not installed, run the following two commands in R or R-Studio console `install.packages("tidyverse")`. If this is your first time using RStudio please note that you might also need to install tinytex using the following `install.packages("tinytex")`

Step 1: This step involves the following:

- Download the data.
- Import the tidyverse and the lubridate libraries.
- View the internal structure of the data frame.

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.1      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.2      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## 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.

## spc_tbl_ [27,312 x 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ INCIDENT_KEY : num [1:27312] 2.29e+08 1.37e+08 1.48e+08 1.47e+08 5.89e+07 ...
## $ OCCUR_DATE : chr [1:27312] "05/27/2021" "06/27/2014" "11/21/2015" "10/09/2015" ...
## $ OCCUR_TIME : 'hms' num [1:27312] 21:30:00 17:40:00 03:56:00 18:30:00 ...
## ..- attr(*, "units")= chr "secs"
## $ BORO : chr [1:27312] "QUEENS" "BRONX" "QUEENS" "BRONX" ...
## $ LOC_OF_OCCUR_DESC : chr [1:27312] NA NA NA NA ...
## $ PRECINCT : num [1:27312] 105 40 108 44 47 81 114 81 105 101 ...
## $ JURISDICTION_CODE : num [1:27312] 0 0 0 0 0 0 0 0 0 0 ...
## $ LOC_CLASSFCTN_DESC : chr [1:27312] NA NA NA NA ...
## $ LOCATION_DESC : chr [1:27312] NA NA NA NA ...
## $ STATISTICAL_MURDER_FLAG: logi [1:27312] FALSE FALSE TRUE FALSE TRUE TRUE ...
## $ PERP_AGE_GROUP : chr [1:27312] NA NA NA NA ...
## $ PERP_SEX : chr [1:27312] NA NA NA NA ...
## $ PERP_RACE : chr [1:27312] NA NA NA NA ...
## $ VIC_AGE_GROUP : chr [1:27312] "18-24" "18-24" "25-44" "<18" ...
## $ VIC_SEX : chr [1:27312] "M" "M" "M" "M" ...
## $ VIC_RACE : chr [1:27312] "BLACK" "BLACK" "WHITE" "WHITE HISPANIC" ...
## $ X_COORD_CD : num [1:27312] 1058925 1005028 1007668 1006537 1024922 ...
## $ Y_COORD_CD : num [1:27312] 180924 234516 209837 244511 262189 ...
## $ Latitude : num [1:27312] 40.7 40.8 40.7 40.8 40.9 ...
## $ Longitude : num [1:27312] -73.7 -73.9 -73.9 -73.9 -73.9 ...
## $ Lon_Lat : chr [1:27312] "POINT (-73.73083868899994 40.662964620000025)" "POINT (-73.73083868899994 40.662964620000025)" ...
## - attr(*, "spec")=
## .. cols(
## .. INCIDENT_KEY = col_double(),
## .. OCCUR_DATE = col_character(),
## .. OCCUR_TIME = col_time(format = ""),
## .. BORO = col_character(),
## .. LOC_OF_OCCUR_DESC = col_character(),
## .. PRECINCT = col_double(),
## .. JURISDICTION_CODE = col_double(),
## .. LOC_CLASSFCTN_DESC = col_character(),
## .. LOCATION_DESC = col_character(),
## .. STATISTICAL_MURDER_FLAG = col_logical(),
## .. PERP_AGE_GROUP = col_character(),
## .. PERP_SEX = col_character(),
## .. PERP_RACE = col_character(),
## .. VIC_AGE_GROUP = col_character(),
## .. VIC_SEX = col_character(),
## .. VIC_RACE = col_character(),
## .. X_COORD_CD = col_double(),
## .. Y_COORD_CD = col_double(),
## .. Latitude = col_double(),
## .. Longitude = col_double(),
## .. Lon_Lat = col_character()
## .. )
## - attr(*, "problems")=<externalptr>

```

Step 2: This step will tidy and/or transform the data to make it ready for the visualization steps.

```
## # A tibble: 17 x 2
##   year incidents
##   <dbl>     <int>
## 1  2006      2055
## 2  2007      1887
## 3  2008      1959
## 4  2009      1828
## 5  2010      1912
## 6  2011      1939
## 7  2012      1717
## 8  2013      1339
## 9  2014      1464
##10  2015      1434
##11  2016      1208
##12  2017       970
##13  2018       958
##14  2019       967
##15  2020      1948
##16  2021      2011
##17  2022      1716
```

Step 3: Let's graph the data now:

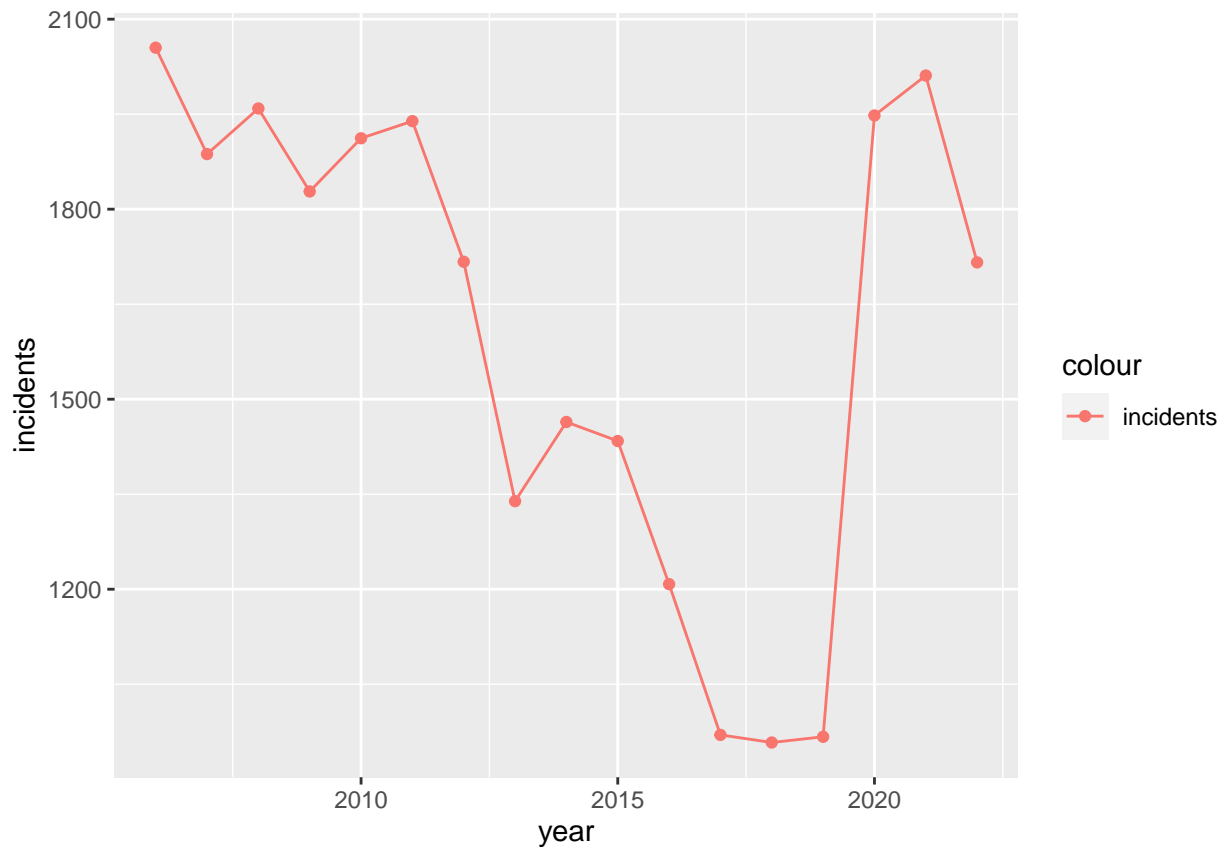
- As you can see below the shooting incidents have been dropping since 2011, then there is a big spike that starts in 2020.

```
data <- data %>%
  mutate(date=mdy(OCCUR_DATE)) %>%
  mutate(hour=as.numeric( format(strptime(data$OCCUR_TIME,"%H:%M:%S"),'%H') )) %>%
  mutate(shot = 1) %>% mutate(Dead = ifelse(STATISTICAL_MURDER_FLAG=="TRUE", "Yes", "No") )

data <- data %>% mutate(date=mdy(OCCUR_DATE)) %>% mutate(year=year(date) )

group_by_year <- data %>% group_by(year) %>% summarize(incidents = n())

group_by_year %>% ggplot(aes(x = year, y = incidents)) +
  geom_line(aes(y =incidents, color = "incidents")) +
  geom_point(alpha = 1, aes(color = "incidents")) #+scale_y_log10()
```



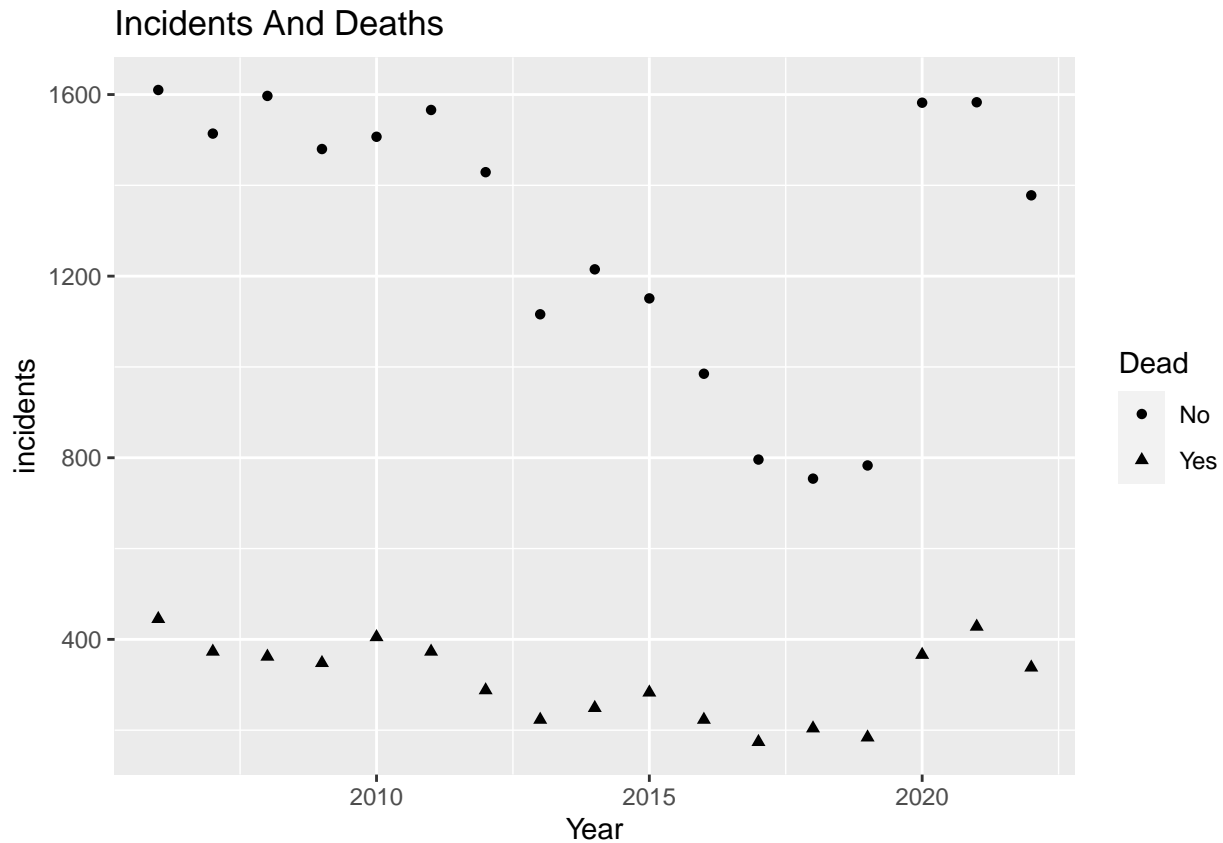
- Another way to look into the data is by graphing the data a bit differently as shown below. We do need to transform the data a bit to make it work.

```
# group_by_year_date_and_death <- data %>% group_by(year, STATISTICAL_MURDER_FLAG) %>% summarize(incid
group_by_year_date_and_death <- data %>% group_by(year, Dead) %>% summarize(incidents = n())

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

- What I have done here is look at the data from another lens, where we group the data based on the incidents and deaths, then graph it.

```
group_by_year_date_and_death %>%
  ggplot(aes(x = year, y = incidents, shape = Dead)) + geom_point() +
  labs(x = "Year", y = "incidents", title='Incidents And Deaths')
```



- Another way to look at the data is to view when these incidents occur during the day, as you can see it increases in the evening and starts dropping around 5AM.

```
count_shooting_by_hour <- data %>%
  select(INCIDENT_KEY, hour, OCCUR_TIME, PERP_SEX, VIC_SEX, PRECINCT) %>%
  group_by(hour) %>% summarize(incidents = n())

count_shooting_by_BORO <- data %>% group_by(BORO) %>% summarize(incidents = n())

count_shooting_by_PRECINCT <- data %>% group_by(PRECINCT) %>% summarize(incidents = n())

count_shooting_by_PERP_SEX <- data %>% group_by(PERP_SEX) %>% summarize(incidents = n())

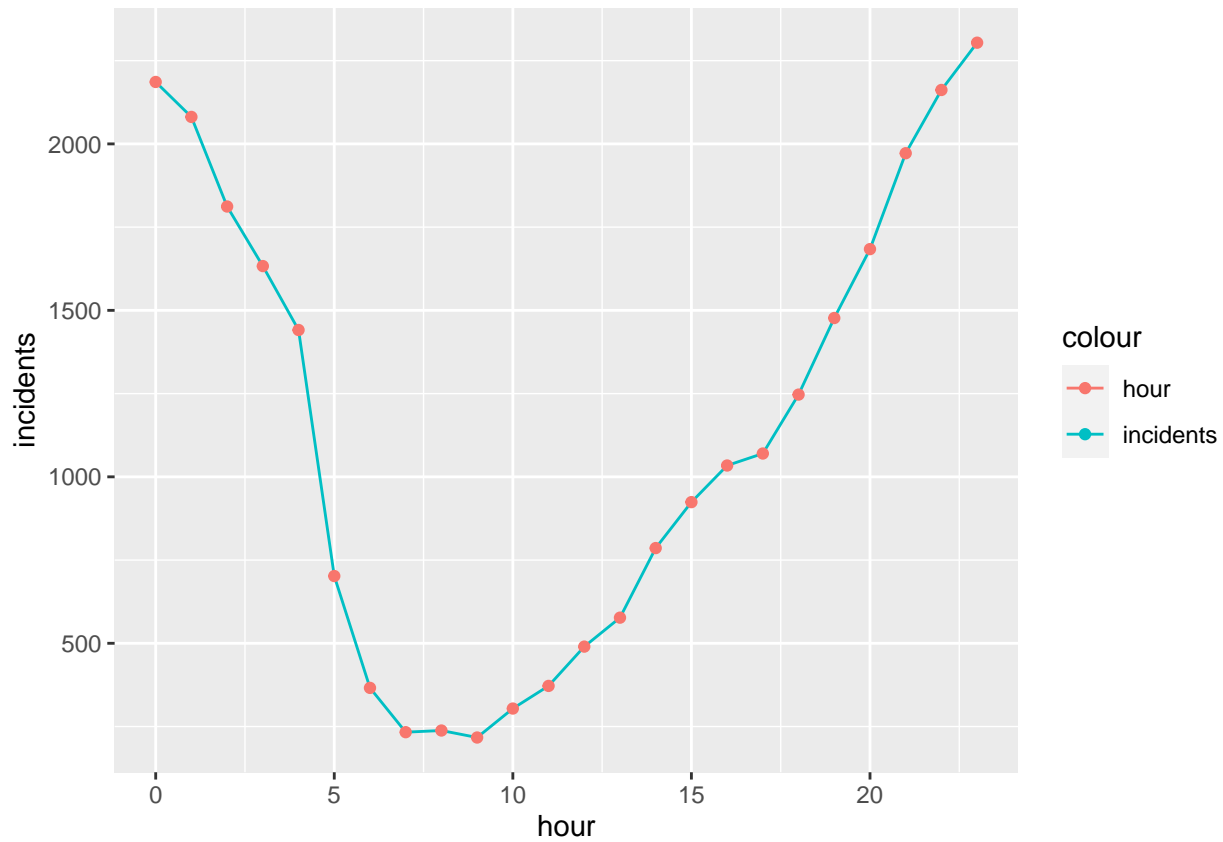
count_shooting_by_hour
```

```
## # A tibble: 24 x 2
##   hour incidents
##   <dbl>     <int>
## 1     0     2186
## 2     1     2081
## 3     2     1812
## 4     3     1633
## 5     4     1441
## 6     5      702
## 7     6      366
## 8     7      233
## 9     8      238
```

```
## 10      9      217
## # i 14 more rows

# view(count_shooting_by_hour)

### Shooting incidents seem to start increasing at night and stop goes down in the morning/afternoon
count_shooting_by_hour %>% ggplot(aes(x = hour, y = incidents)) +
  geom_line(aes(y = incidents, color = "incidents")) +
  geom_point(alpha = 1, aes(color = "hour")) #+scale_y_log10()
```

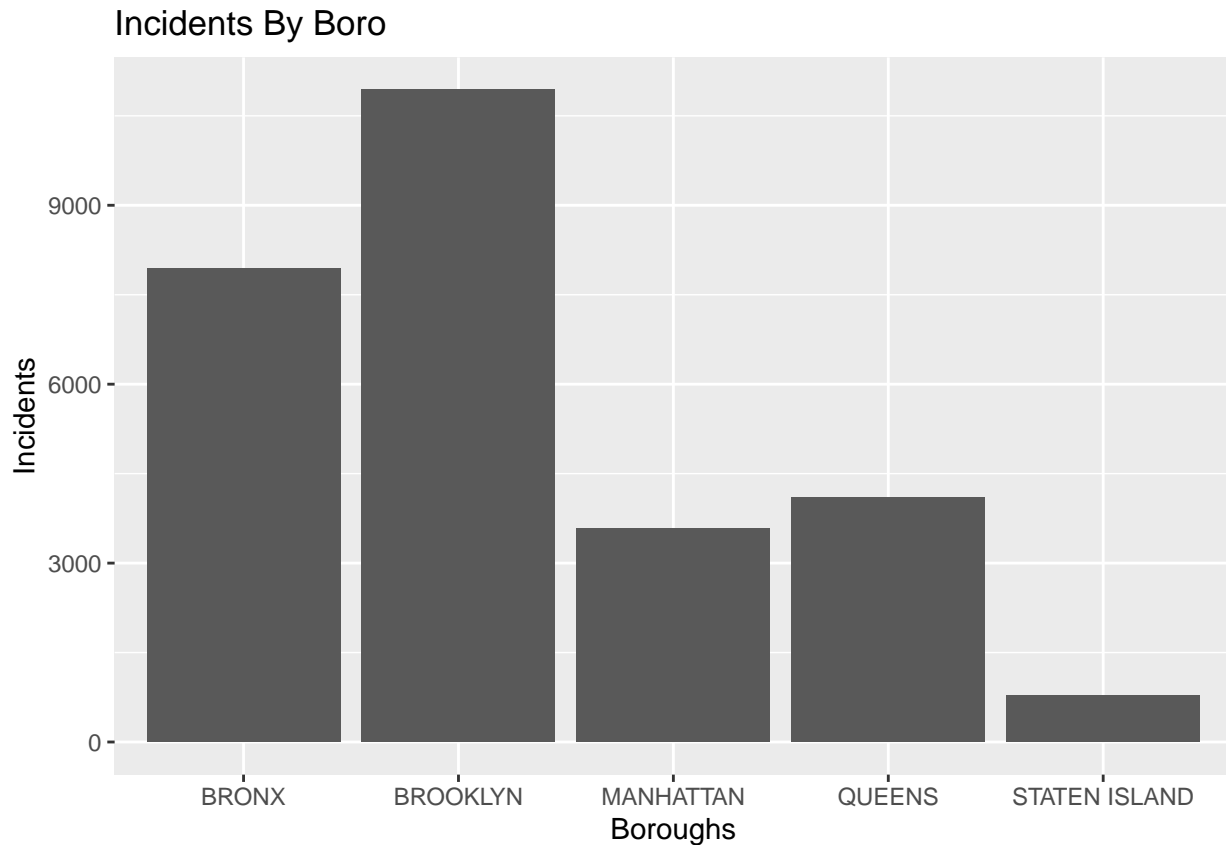


data

- Another graph shows the shooting incidents by Borough.

```
grouped_by_boro <- data %>%
  group_by(BORO) %>%
  count() %>%
  ungroup()

grouped_by_boro %>%
  ggplot(aes(x = BORO, y = n)) +
  geom_bar(stat='identity') +
  labs(title = "Incidents By Boro", x = "Boroughs", y = "Incidents")
```



- Another graph shows the shooting incidents by age group, 25 to 44 is the Highest.

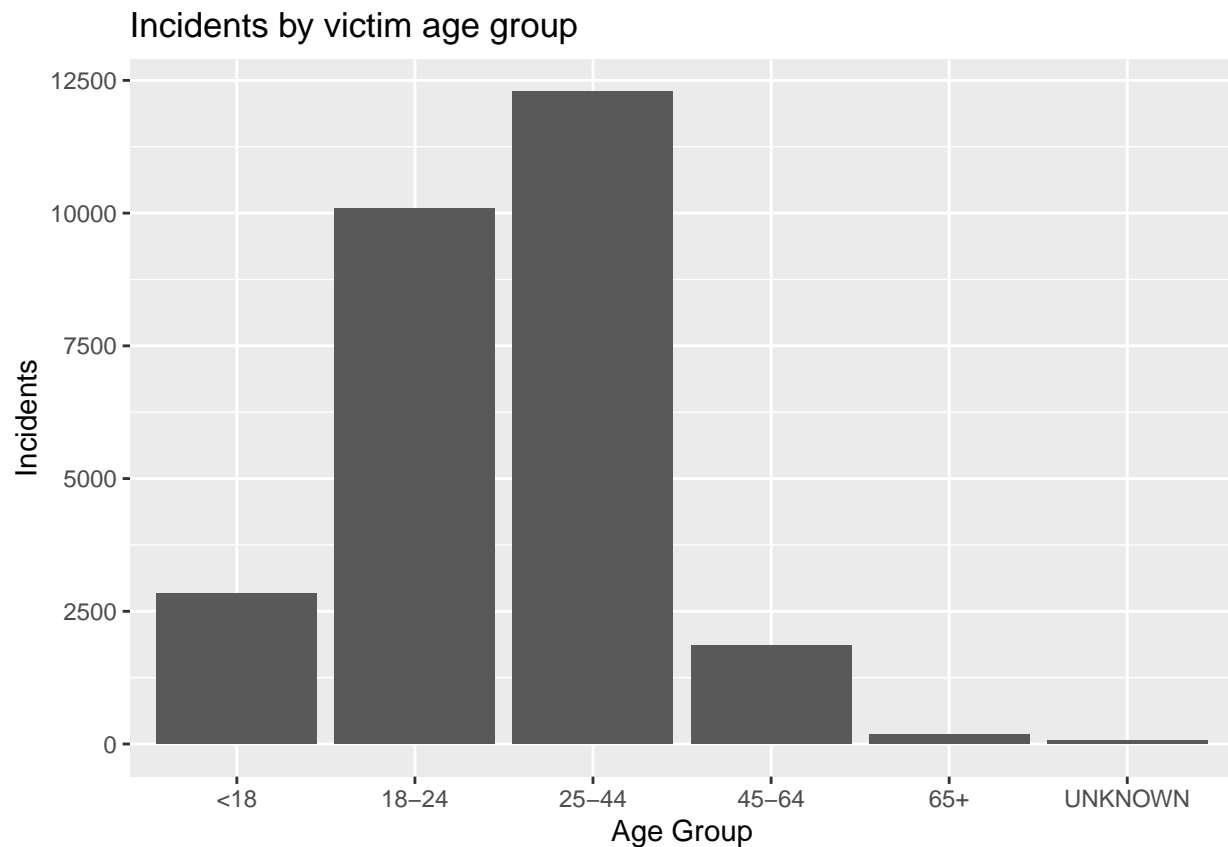
There is a group names 1122 that seems to be a typo, it has a value of 1, I'm filtering it out before

```
grouped_by_age_group <- data %>%
  filter(VIC_AGE_GROUP != "1022") %>%
  group_by(VIC_AGE_GROUP) %>%
  count() %>%
  ungroup()
```

```
summary(grouped_by_age_group)
```

```
## VIC_AGE_GROUP      n
## Length:6          Min.   :  61.0
## Class :character  1st Qu.: 601.5
## Mode  :character  Median :2351.0
##                               Mean  :4551.8
##                               3rd Qu.:8274.2
##                               Max.  :12281.0
```

```
grouped_by_age_group %>%
  ggplot(aes(x = VIC_AGE_GROUP, y = n)) +
  geom_bar(stat='identity') +
  labs(title = "Incidents by victim age group", x = "Age Group", y = "Incidents")
```



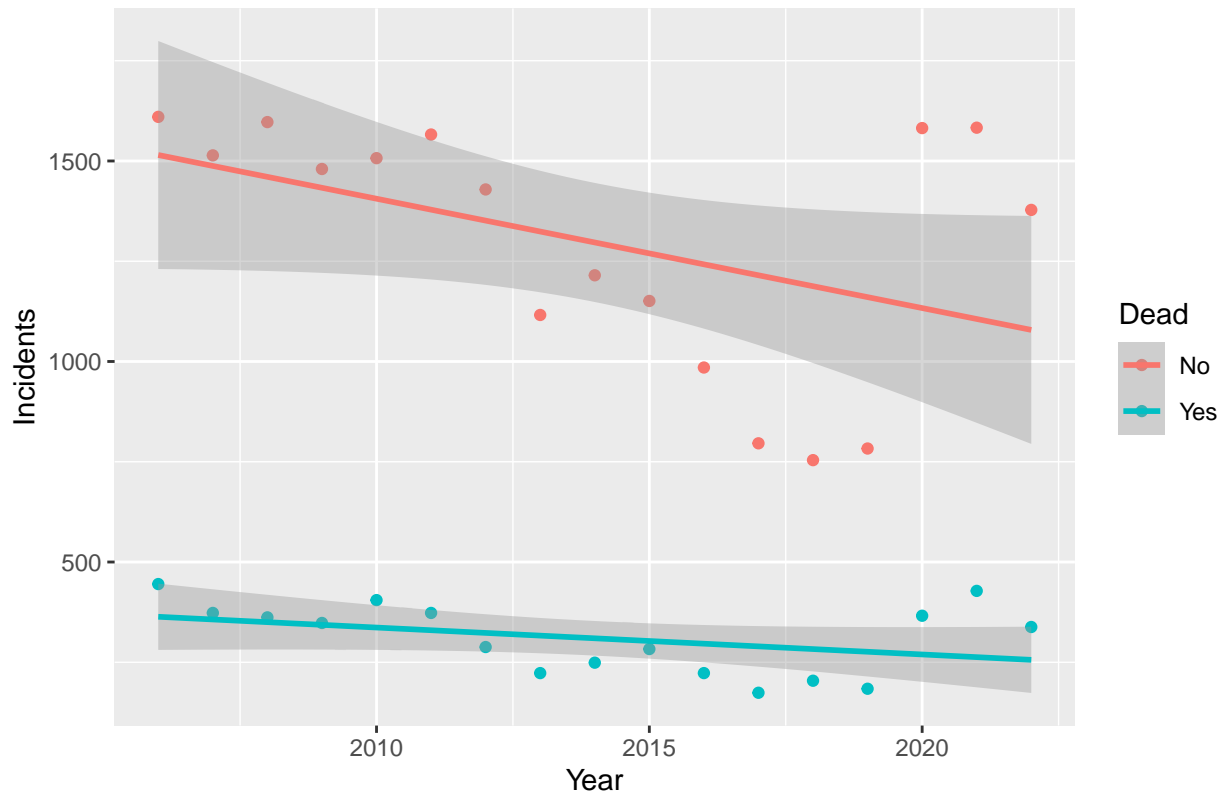
Step 4: Apply a linear model

- Finally I'll be using the same graph but this time with a linear model, the linear model is applied on both outcomes, the incidents that led to deaths and the ones that didn't lead to deaths.

```
group_by_year_date_and_death %>% ggplot(aes(x = year, y = incidents, color = Dead)) +  
  geom_point() +  
  labs(x = "Year", y = "Incidents", title='Incidents And Deaths') +  
  geom_smooth(method = "lm")
```

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


Incidents And Deaths



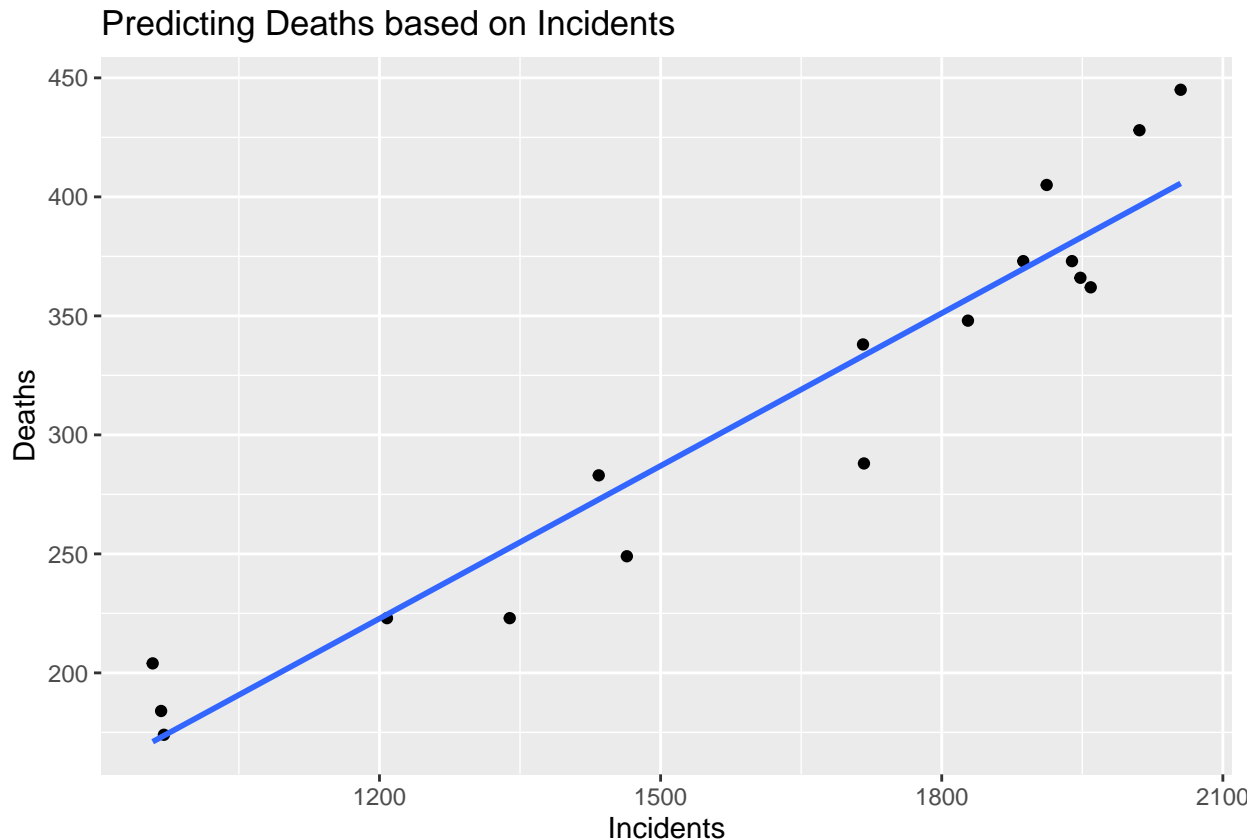
```
grouped_incidents_and_deaths_by_date <- data %>%
  group_by(year) %>%
  summarize(deaths = sum(Dead == "Yes"),
            incidents = n())
) %>%
  ungroup()
grouped_incidents_and_deaths_by_date
```

```
## # A tibble: 17 x 3
##   year deaths incidents
##   <dbl> <int>    <int>
## 1 2006     445     2055
## 2 2007     373     1887
## 3 2008     362     1959
## 4 2009     348     1828
## 5 2010     405     1912
## 6 2011     373     1939
## 7 2012     288     1717
## 8 2013     223     1339
## 9 2014     249     1464
## 10 2015     283     1434
## 11 2016     223     1208
## 12 2017     174      970
## 13 2018     204      958
## 14 2019     184      967
## 15 2020     366     1948
## 16 2021     428     2011
```

```
## 17 2022 338 1716
```

```
grouped_incidents_and_deaths_by_date %>% ggplot(aes(x = incidents, y = deaths)) +  
  geom_point() +  
  labs(x = "Incidents", y = "Deaths", title='Predicting Deaths based on Incidents') +  
  geom_smooth(method = "lm", se=FALSE)
```

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



Conclusion

- The number of incidents increased significantly around 2020.
- Safe hours to be out in the boroughs according to this data is between ~5AM and 6PM.
- Highest shooting incidents are in Brooklyn.
- Highest shooting incidents based on age are between 25-44.

Bias

We need to be careful when we analyze such data/reports, many biases can be present here, for example, who is collecting the data? Is there any data compliance that these reports go through or follow? What about the data entry, are these accurate? When these data are being entered, is it the time of the shooting? or after a few days?

Another thing I was looking at that we need to be careful about is the age group, as noted above 25 to 44 seems to have the highest number of incidents, but I think that makes sense since maybe this group is the one that has big representation, this age group is simply out more than other age groups.