NYPD Shooting Incident Data Report

Alden Lin Azhi

2024-03-02

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
```

What is this?

This document uses NYPD Shooting Incident Data (Historic) posted on below site to do some data analysis work, as part of CU Boulder's Data Science as A Field course project work.

https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

Step 1 - Important the data in a reproducible manner

```
url = "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
nypd_shooting = read.csv(url)
```

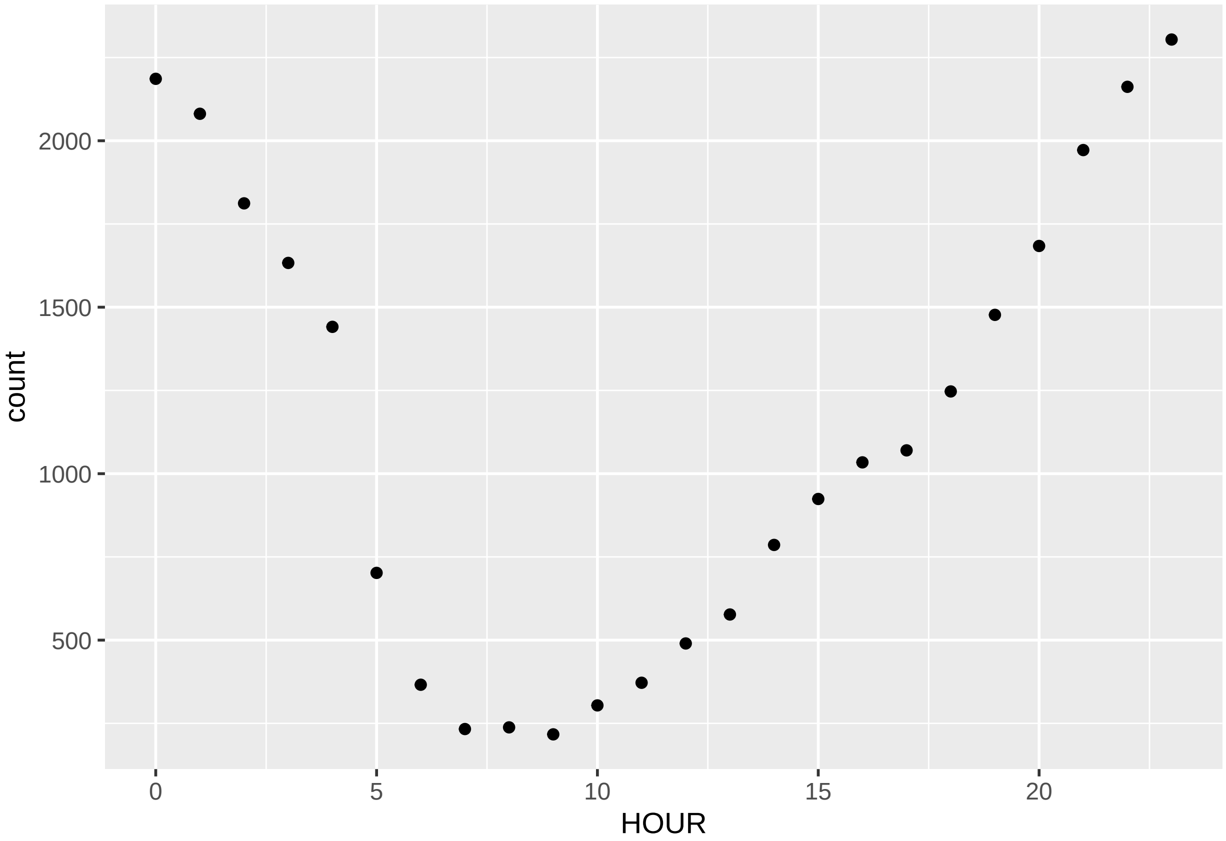
Step 2 - Cleaning, Analysis and Visualization

Most Dangeous Hours and Least Dangeous Hours

I would like to gain some insights which hours the city is safer and which hours less. The analysis below indicates that mid-night is the the most dangerous while early in the morning safest.

```
#Wrangling of data to get number of cases by hour
nypd_shooting <- nypd_shooting %>% mutate(OCCUR_TIME = hms(OCCUR_TIME))
cases_by_hour <- nypd_shooting %>%
  mutate(HOUR=hour(OCCUR_TIME)) %>%
  group_by(HOUR) %>%
  summarize(count=n())
```

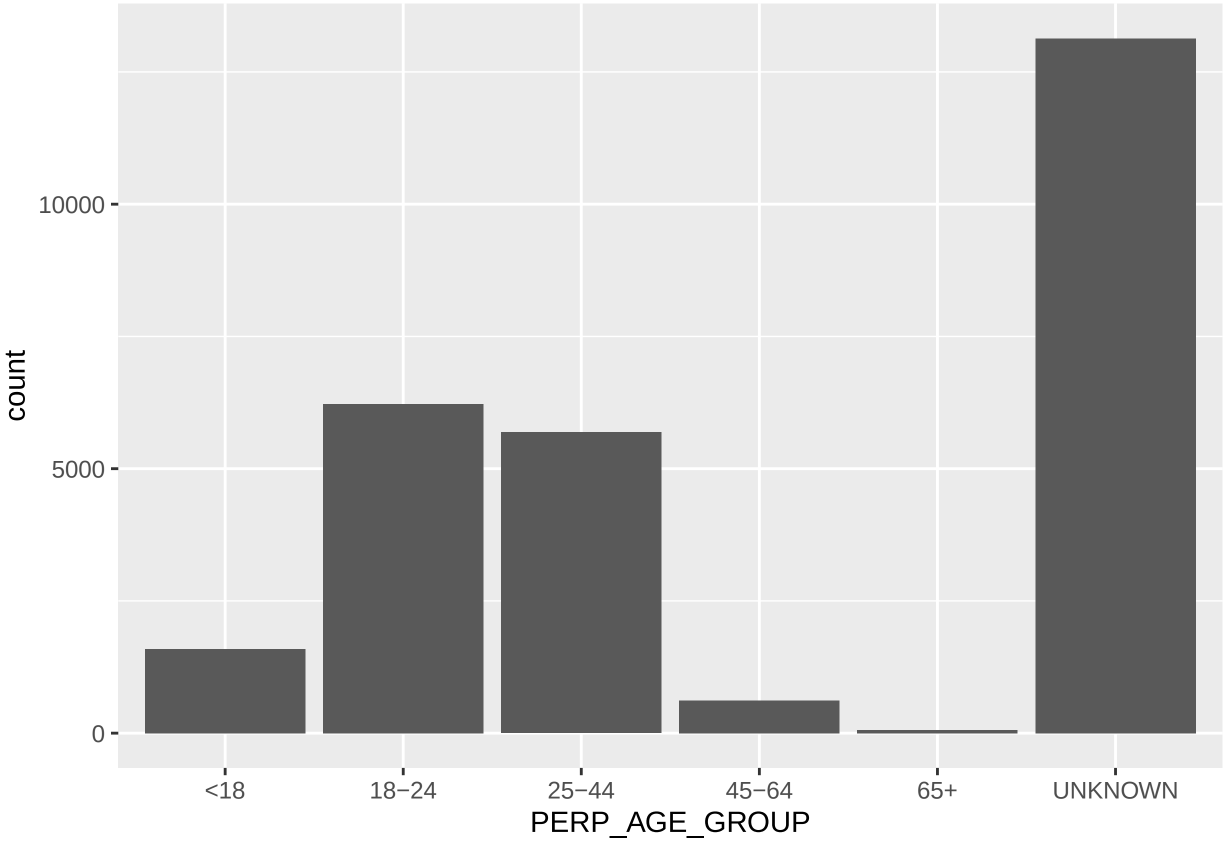
```
#plot the data
cases_by_hour %>%
 ggplot() +
 geom_point(aes(HOUR,count))
```



### Perpetrator Age Group Analysis

Now I would like to know which age groups are more violent and which group least. This is done by analyzing the age group of perpetrators in the shooting cases. The chart shows that 18~24 is the most violent group while 65-plus is least violent, which is true to common sense.

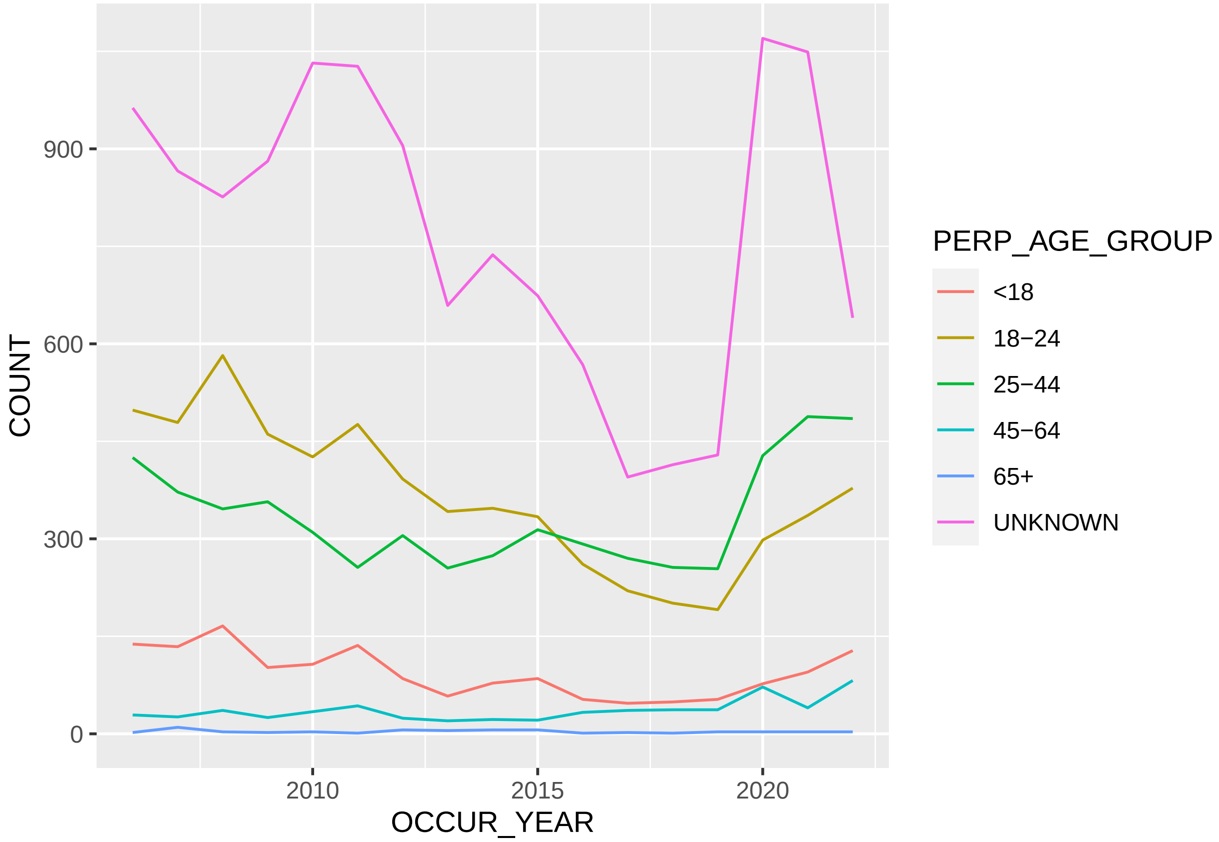
```
nypd_shooting <- nypd_shooting %>% mutate(PERP_AGE_GROUP = ifelse(PERP_AGE_GROUP %in% c("(null)","1020"

nypd_shooting %>% group_by(PERP_AGE_GROUP) %>%  summarize(count=n()) %>%
  ggplot(aes(x=PERP_AGE_GROUP,y=count)) +
  geom_bar(stat="identity")
```



But wait, the number of UNKNOWN perpetrator age group is so high? It's almost twice the amount of 18~24 cases. This pretty much means there is a significant amount of cases with perpetrator identify unidentified. Is it always the case every year or just for certain years?

Let me try to find out:

```
perp_age_trend <- nypd_shooting %>%
  mutate(OCCUR_DATE=mdy(OCCUR_DATE)) %>%
  mutate(OCCUR_YEAR = year(OCCUR_DATE)) %>%
  group_by(OCCUR_YEAR,PERP_AGE_GROUP) %>% summarize(COUNT = n(),.groups = "drop")


perp_age_trend %>%
 ggplot(aes(OCCUR_YEAR,COUNT)) +
 geom_line(aes(color=PERP_AGE_GROUP))
```



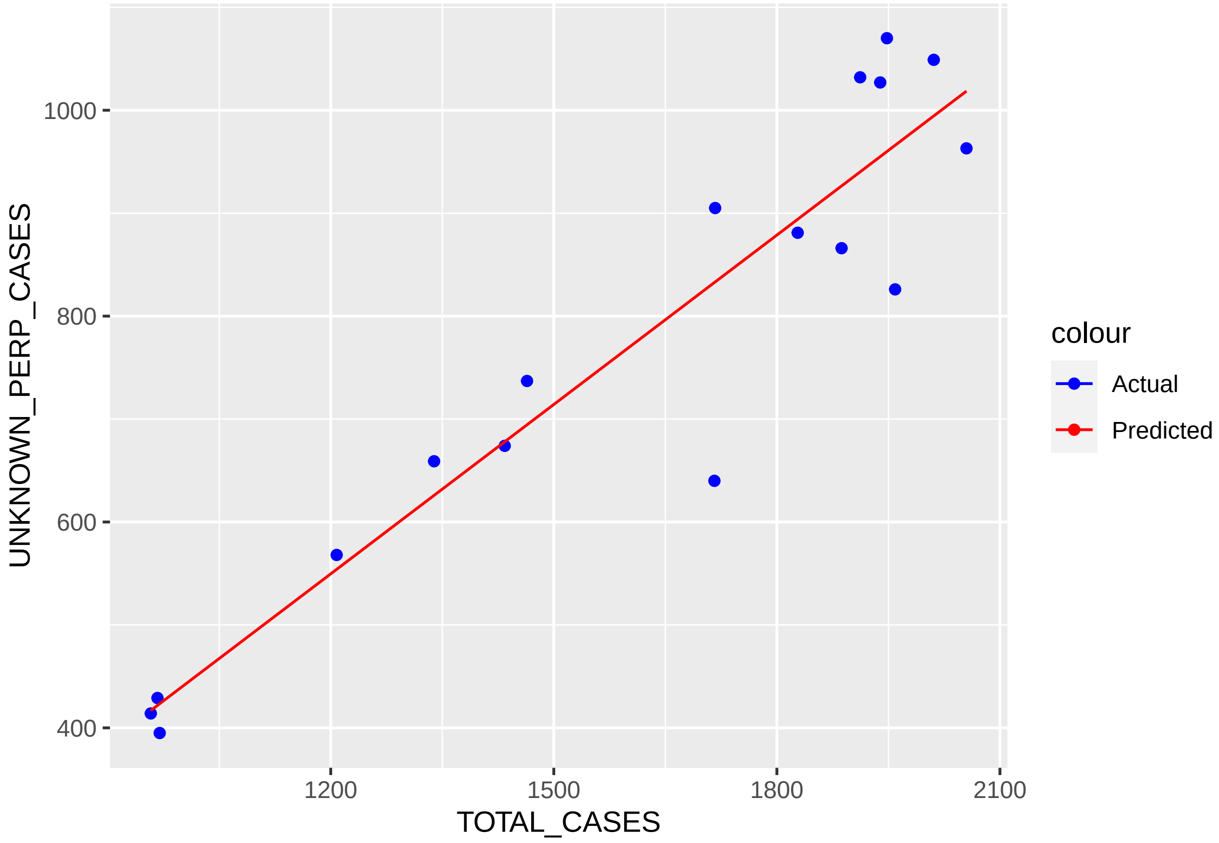
From the above chart, we can safely say that for every year the number of unidentified perpetrators takes up a significant part of total shooting cases in New York.

Can we say the higher the total number of cases, the more cases of unidentified perpetrator? Let's find out using linear modeling and the answer is yes.

```
# get total number of cases for each year
cases_by_year_total <- perp_age_trend %>% group_by(OCCUR_YEAR) %>%  summarize(TOTAL_CASES=sum(COUNT))

# get number of cases with unknown perpetrators for each year
cases_by_year_unknown_perp <- perp_age_trend %>% filter(PERP_AGE_GROUP == "UNKNOWN") %>% mutate(UNKNOWN_

#merge the two data frames above
cases_by_year <- full_join(cases_by_year_total,cases_by_year_unknown_perp)
```

```
## Joining with ‘by = join_by(OCCUR_YEAR)‘
```

```
#all set for linear modeling
mod <- lm(UNKNOWN_PERP_CASES~TOTAL_CASES,data=cases_by_year)

# prediction based on the model created
cases_by_year <- cases_by_year%>% mutate(pred = predict(mod))

#visualization
cases_by_year %>%
  ggplot() +
  geom_point(aes(x = TOTAL_CASES, y = UNKNOWN_PERP_CASES, color = "Actual")) +
  geom_line(aes(x = TOTAL_CASES, y = pred, color = "Predicted")) +
  scale_color_manual(values = c("Actual" = "blue", "Predicted" = "red"))
```


UNKNOWN_PERP_CASES
TOTAL_CASES
colour
Actual
Predicted