NYPD_Shooting

Lei Yao

2022-04-24

1. Data Set and Inspired Question

NYPD Shooting Incident Data (Historic) is available from the NYPD website, which is extracted and reviewed by the Office of Management Analysis and Planning. Another access to the data is going to the Data.gov website (https://catalog.data.gov/dataset) and searching for it.

The data records every shooting incident that occurred in NYC from January 1, 2006, to December 31, 2020, as well as information about the event, the location and time of occurrence, and information related to suspect and victim demographics.

I am interested in exploring the relationship between the number of shooting cases and other factors, like occurrence date, borough, victim age group, victim sex, and victim race.

2. Import Data

```
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()
```

```
library(lubridate)
```

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

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

```
library(dplyr)
```

Let's begin with importing the data and see what we have.

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

```
## 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.
```

```
head(NYPD_Shooting_raw)
```

```
## # 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>
```

I would like to tidy the raw data. The variables I am interested in are OCCUR_DATE, BORO, VIC_AGE_GROUP, VIC_SEX, and VIC_RACE, so I will group the data set by these variables and add a new variable to count the number of cases for each line. Then change the OCCUR_DATE variable into the year-month-date format, which is more R-friendly in the future analysis.

```
NYPD_Shooting <- NYPD_Shooting_raw %>%
  mutate(date = mdy(OCCUR_DATE)) %>%
  mutate(cases = 1) %>%
  group_by(date, BORO, VIC_AGE_GROUP, VIC_SEX, VIC_RACE) %>%
  summarize(cases = sum(cases)) %>%
  ungroup()
```

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

```
head(NYPD_Shooting)
```

```
## # A tibble: 6 x 6
##   date       BORO      VIC_AGE_GROUP VIC_SEX VIC_RACE       cases
##   <date>     <chr>     <chr>         <chr>   <chr>          <dbl>
## 1 2006-01-01 BRONX     <18           M       BLACK              1
## 2 2006-01-01 BRONX     18-24         M       WHITE HISPANIC     1
## 3 2006-01-01 BROOKLYN  18-24         M       BLACK              1
## 4 2006-01-01 BROOKLYN  25-44         M       BLACK              1
## 5 2006-01-01 MANHATTAN 25-44         M       BLACK              1
## 6 2006-01-01 QUEENS    18-24         M       BLACK              1
```

```
summary(NYPD_Shooting)
```

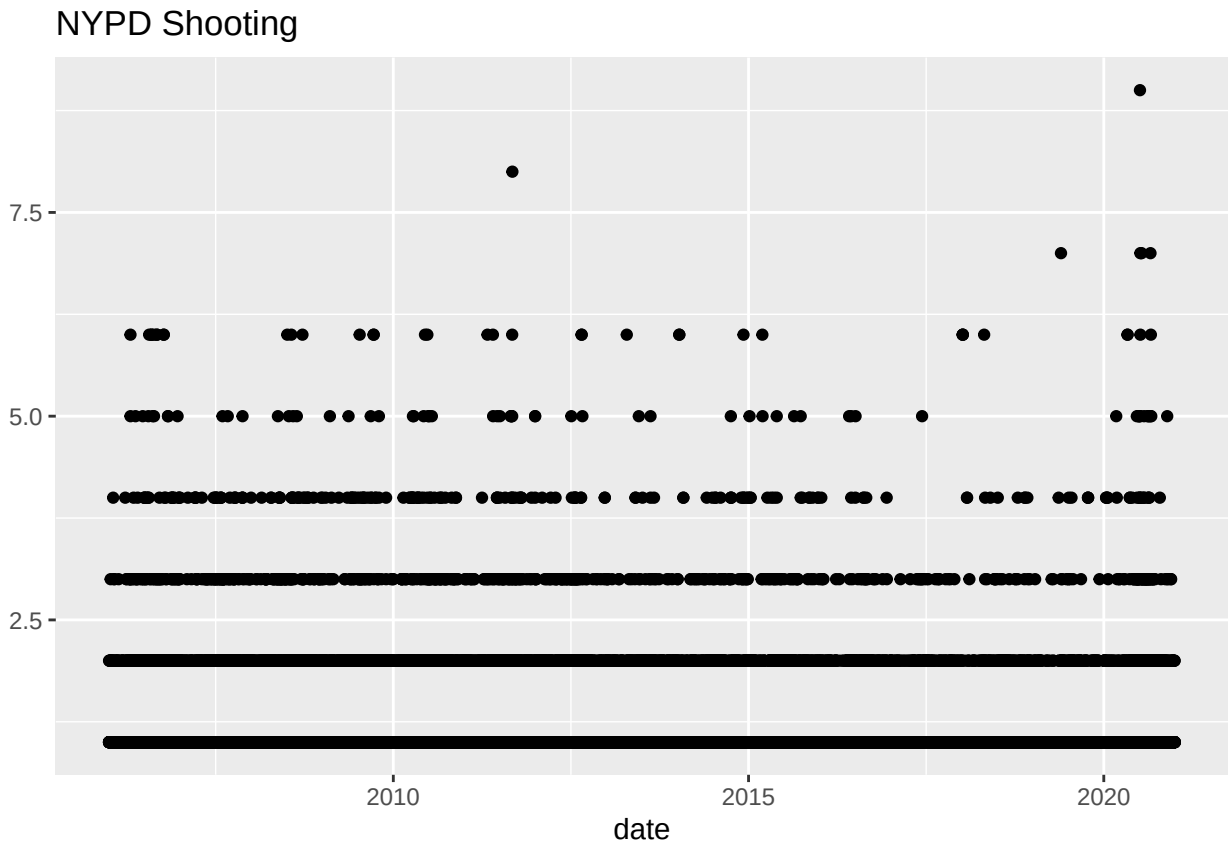
```
##       date                BORO           VIC_AGE_GROUP        VIC_SEX
##  Min.   :2006-01-01   Length:18262       Length:18262       Length:18262
##  1st Qu.:2009-02-18   Class :character   Class :character   Class :character
```

```
##  Median :2012-05-13   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2012-11-06
##  3rd Qu.:2016-04-27
##  Max.   :2020-12-31
##    VIC_RACE             cases
##  Length:18262       Min.   :1.000
##  Class :character   1st Qu.:1.000
##  Mode  :character   Median :1.000
##                     Mean   :1.291
##                     3rd Qu.:1.000
##                     Max.   :9.000
```

3. Visualization

I will put the occurrence date on the x-axis and the number of shooting cases on the y-axis and plot it, let's see what it looks like.

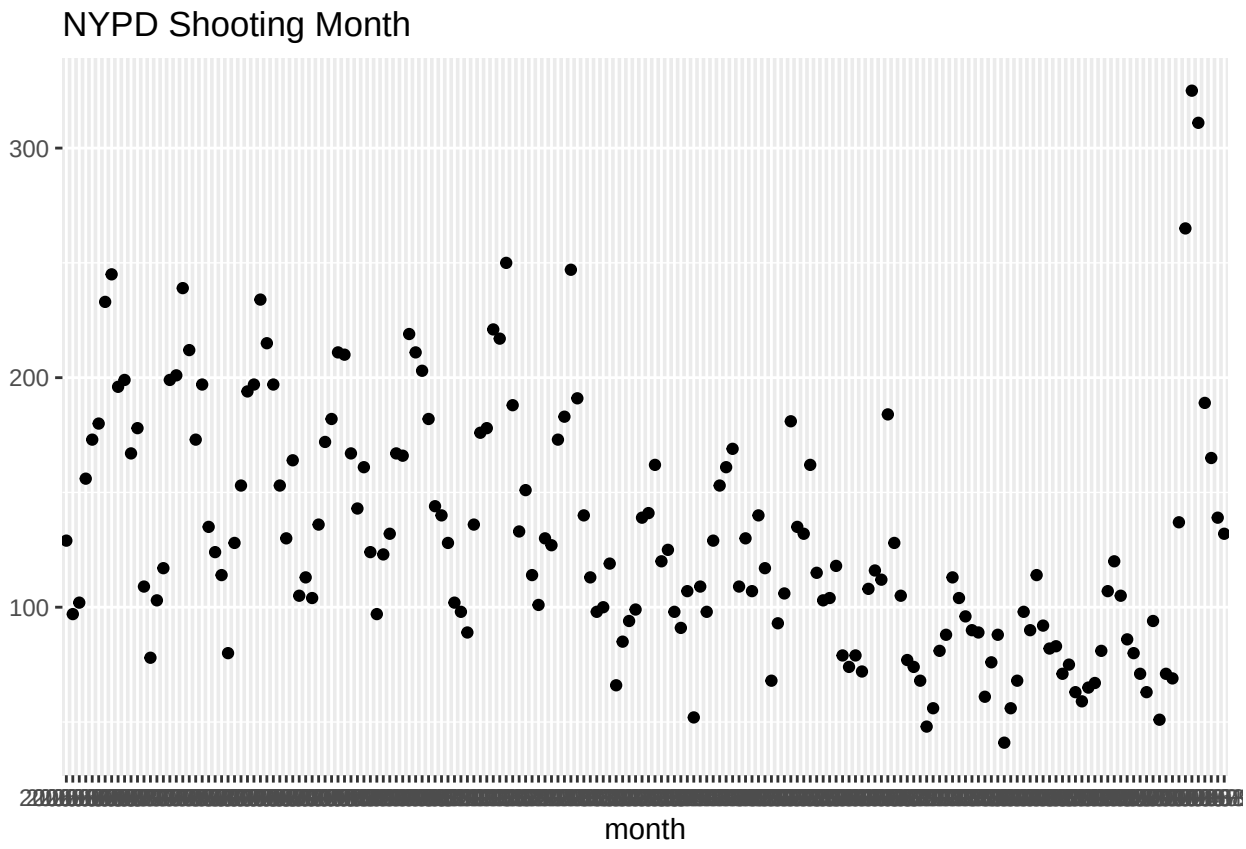
```
NYPD_Shooting %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_point() +
  labs(title = "NYPD Shooting", y = NULL)
```



The plot is messy and there is no obvious trend, so I will change the x-axis from occurrence date to occurrence month. Here is the new visualization:

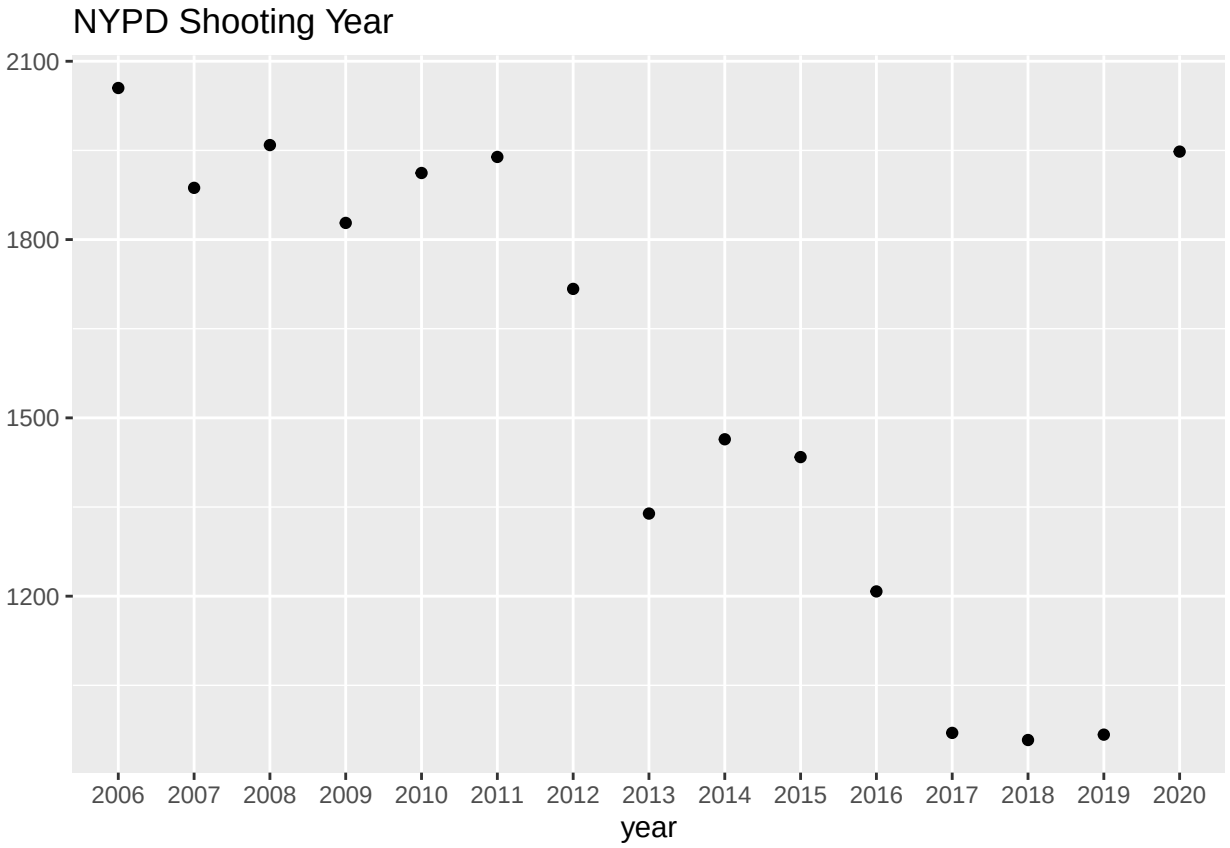
```
NYPD_Shooting_month <- NYPD_Shooting %>%
  mutate(month = format(date, "%Y-%m")) %>%
  group_by(month) %>%
```

```
  summarize(cases = sum(cases)) %>%
  ungroup()

NYPD_Shooting_month %>%
  filter(cases > 0) %>%
  ggplot(aes(x = month, y = cases)) +
  geom_point() +
  labs(title = "NYPD Shooting Month", y = NULL)
```



It seems that the number of shooting cases gradually decreases except at the very end part of the plot. To see it more clearly, I am going on changing the x-axis, from occurrence month to occurrence year.

```
NYPD_Shooting_year <- NYPD_Shooting %>%
  mutate(year = format(date, "%Y")) %>%
  group_by(year) %>%
  summarize(cases = sum(cases)) %>%
  ungroup()

NYPD_Shooting_year %>%
  filter(cases > 0) %>%
  ggplot(aes(x = year, y = cases)) +
  geom_point() +
  labs(title = "NYPD Shooting Year", y = NULL)
```



It seems that by year is a good option since the trend is more obvious now. Let's explore the relationship between occurrence year and the number of shooting cases, by a linear model.

4. Modeling

```
typeof(NYPD_Shooting_year$year) # type of year in NYPD_Shooting_year is character
```

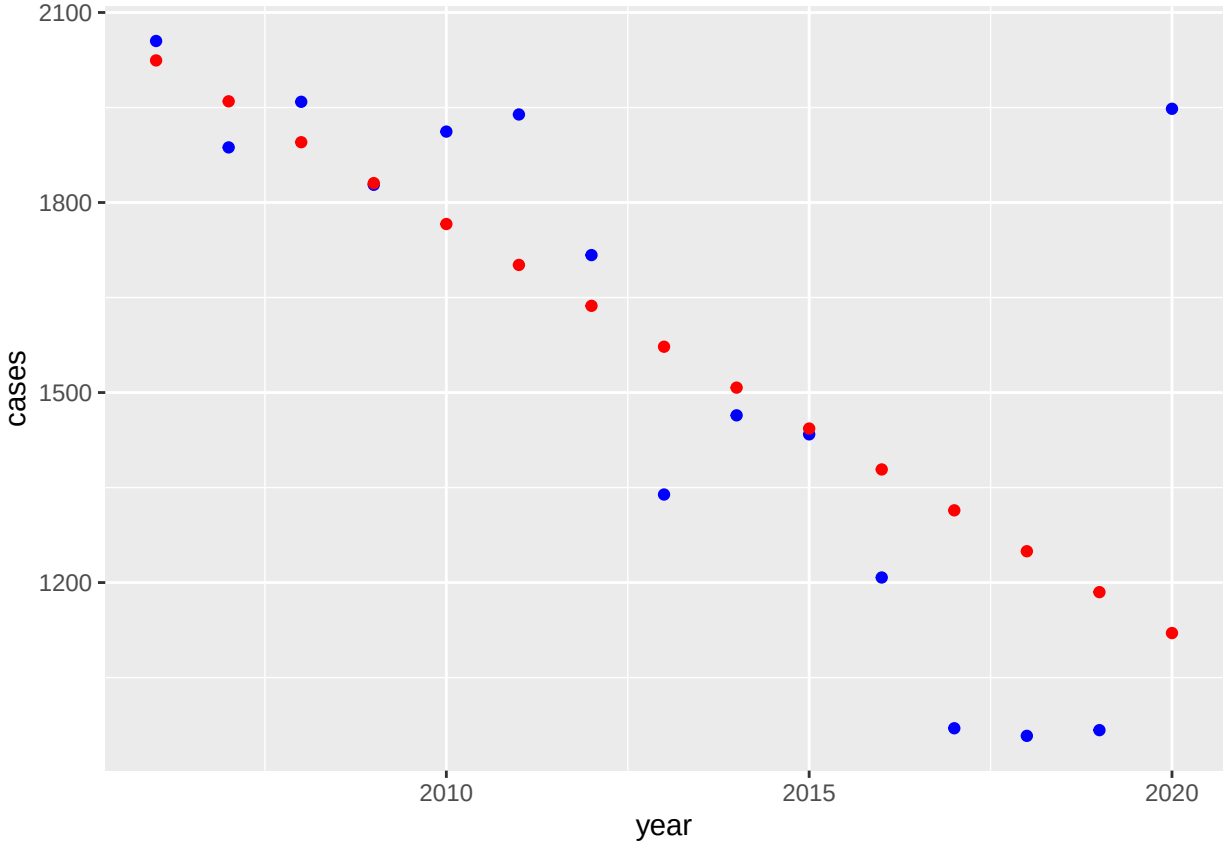
```
## [1] "character"
```

```
NYPD_Shooting_year <- NYPD_Shooting_year %>%
  mutate(year = as.numeric(year)) # change character into double

model <- lm(cases ~ year, data = NYPD_Shooting_year)
summary(model)
```

```
##
## Call:
## lm(formula = cases ~ year, data = NYPD_Shooting_year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -344.03 -194.25   -9.18   71.94  827.69
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 131561.81   35244.87   3.733  0.00251 **
## year           -64.58      17.51  -3.688  0.00273 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 293 on 13 degrees of freedom
## Multiple R-squared:  0.5113, Adjusted R-squared:  0.4737
## F-statistic:  13.6 on 1 and 13 DF,  p-value: 0.002731
```

```
NYPD_Shooting_year_w_pred <- NYPD_Shooting_year %>%
  mutate(pred = predict(model))

NYPD_Shooting_year_w_pred %>% ggplot() +
  geom_point(aes(x = year, y = cases), color = "blue") +
  geom_point(aes(x = year, y = pred), color = "red")
```



We can see the model predicts well at some level, but there are still points off the modeling line. So we may consider other factors as part of the prediction.

5. Visualization Continue - More Factors

I will create a new data set based on the occurrence year because it looks nicer than the month and date. And then visualize the number of shooting cases versus occurrence year and other factors: borough, victim age group, victim sex, and victim race.

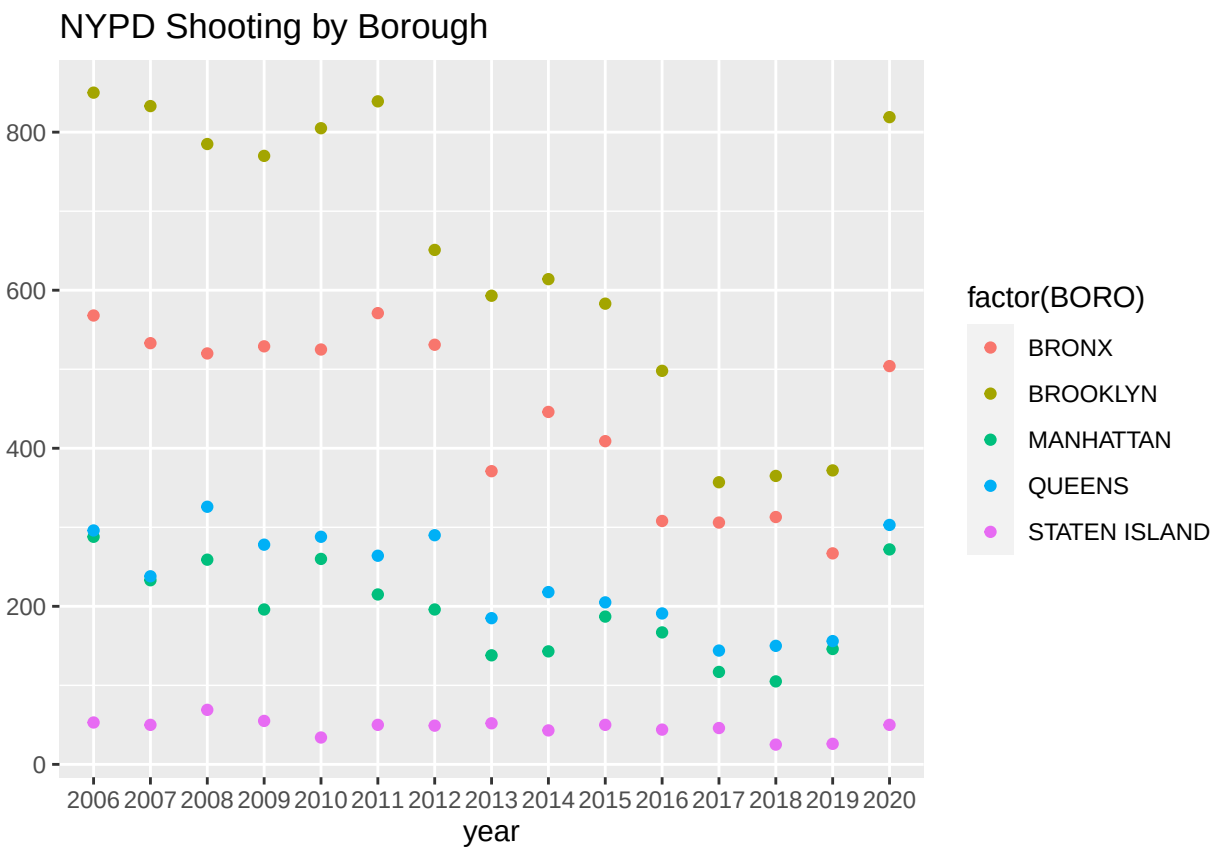
```
NYPD_Shooting_new <- NYPD_Shooting %>%
  mutate(year = format(date, "%Y")) %>%
  select(-date) %>%
  select(year, everything())
```

For borough:

```
NYPD_Shooting_by_boro <- NYPD_Shooting_new %>%
  group_by(year, BORO) %>%
  summarize(cases = sum(cases)) %>%
  ungroup()
```

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

```
NYPD_Shooting_by_boro %>%
  filter(cases > 0) %>%
  ggplot(aes(x = year, y = cases)) +
  geom_point(aes(color = factor(BORO))) +
  labs(title = "NYPD Shooting by Borough", y = NULL)
```



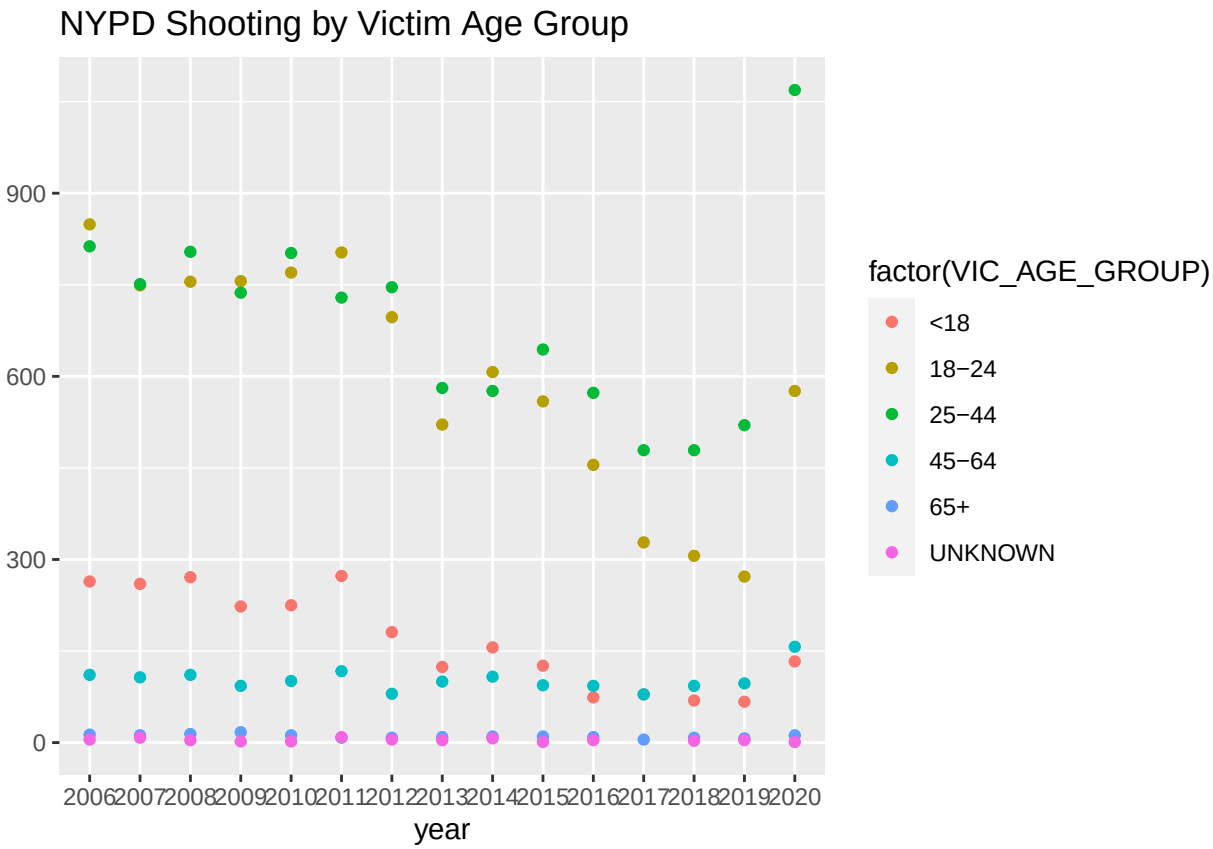
Brooklyn has the most shooting cases, followed by Bronx, Queens, Manhattan, and Staten Island.

For victim age group:

```
NYPD_Shooting_by_vic_age_group <- NYPD_Shooting_new %>%
  group_by(year, VIC_AGE_GROUP) %>%
  summarize(cases = sum(cases)) %>%
  ungroup()
```

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

```
NYPD_Shooting_by_vic_age_group %>%
  filter(cases > 0) %>%
  ggplot(aes(x = year, y = cases)) +
  geom_point(aes(color = factor(VIC_AGE_GROUP))) +
  labs(title = "NYPD Shooting by Victim Age Group", y = NULL)
```



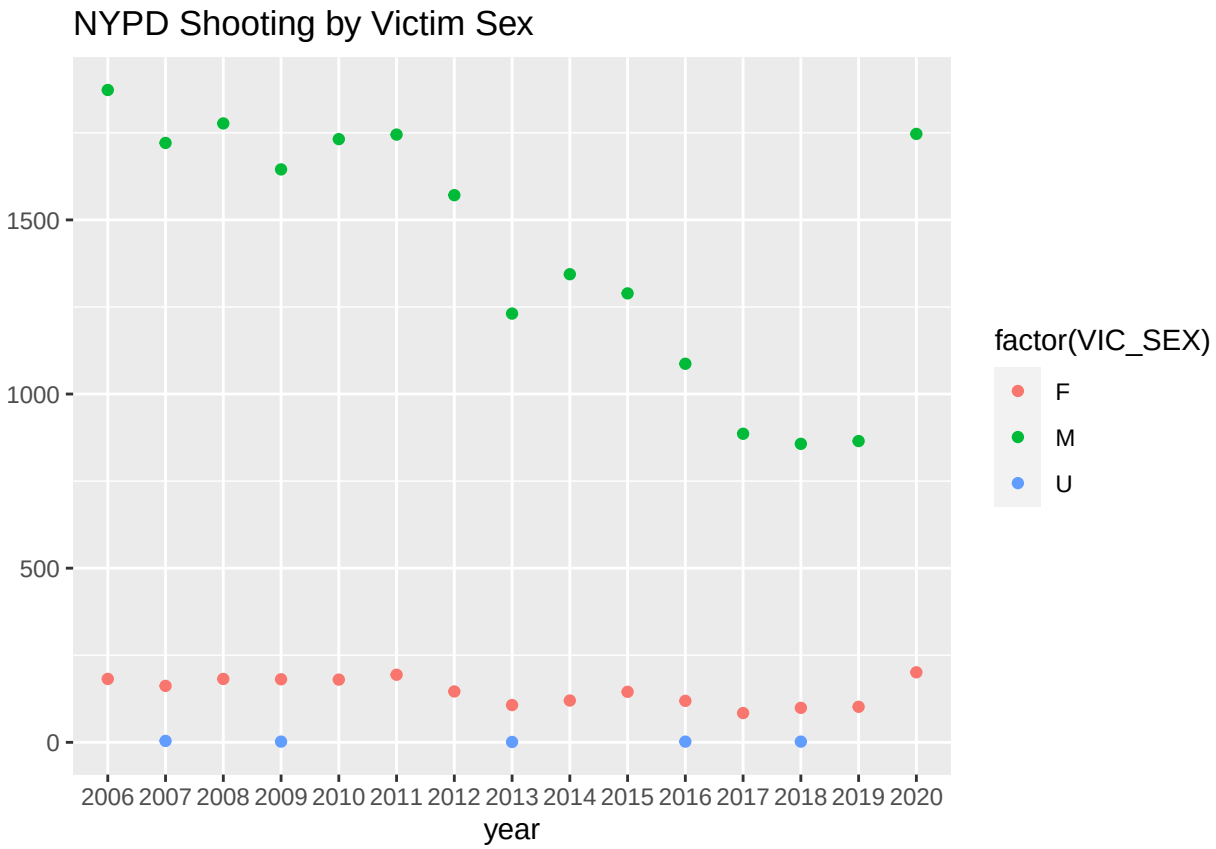
Shooting cases occur most in the 25-44 age group, followed by 18-24, <18, 45-64, 65+, and unknown age group.

For victim sex:

```
NYPD_Shooting_by_vic_sex <- NYPD_Shooting_new %>%
  group_by(year, VIC_SEX) %>%
  summarize(cases = sum(cases)) %>%
  ungroup()
```

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

```
NYPD_Shooting_by_vic_sex %>%
  filter(cases > 0) %>%
  ggplot(aes(x = year, y = cases)) +
  geom_point(aes(color = factor(VIC_SEX))) +
  labs(title = "NYPD Shooting by Victim Sex", y = NULL)
```



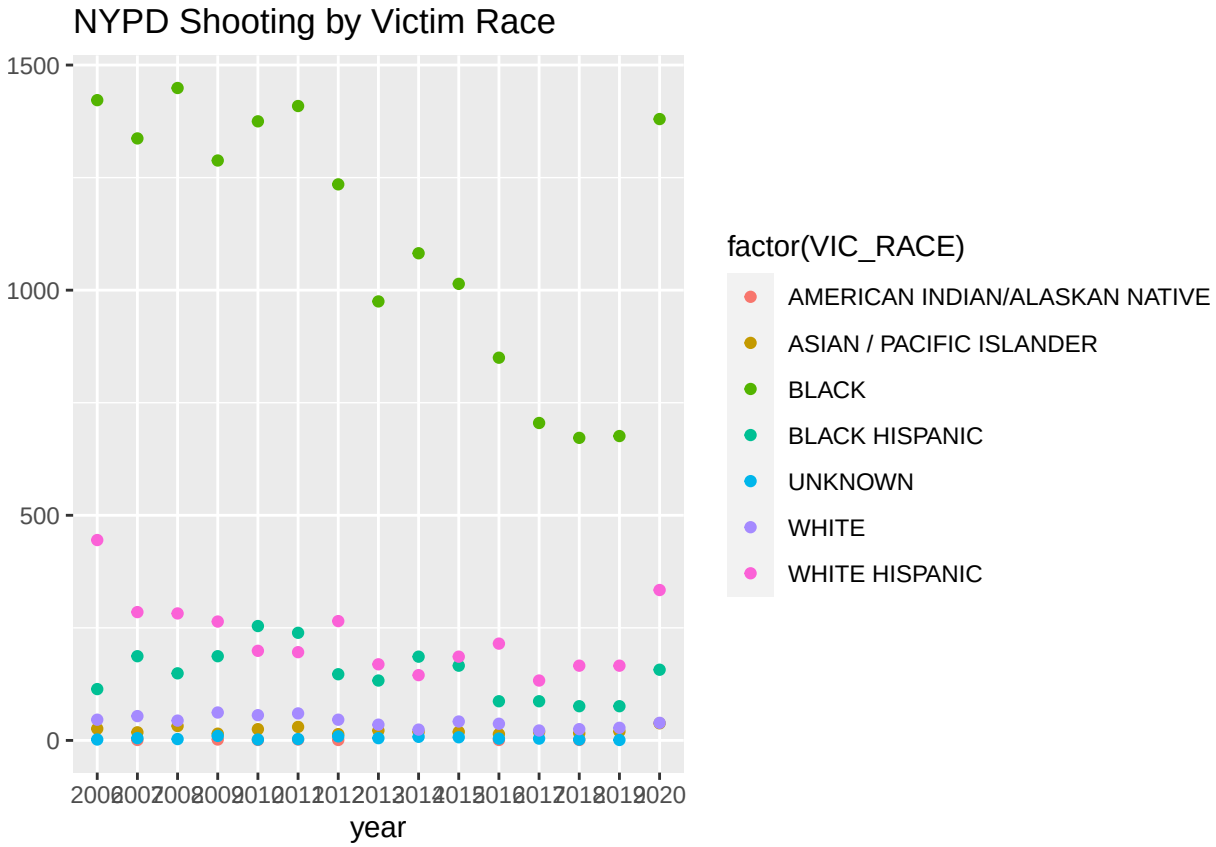
The number of male victims is larger than that of female victims, and that of unknown sex is the least.

For victim race:

```
NYPD_Shooting_by_vic_race <- NYPD_Shooting_new %>%
  group_by(year, VIC_RACE) %>%
  summarize(cases = sum(cases)) %>%
  ungroup()
```

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

```
NYPD_Shooting_by_vic_race %>%
  filter(cases > 0) %>%
  ggplot(aes(x = year, y = cases)) +
  geom_point(aes(color = factor(VIC_RACE))) +
  labs(title = "NYPD Shooting by Victim Race", y = NULL)
```



The number of Black victims is the most, followed by White Hispanics, Black Hispanics, etc.

6. Conclusion and Future Work

In general, the number of shooting cases decreases over time, except for the latest year. Although the number of shooting cases and occurrence year linear model does a reasonably good job of predicting, there are still some points off the modeling line. It is clear that there is some relationship between the factors visualized above and the number of shooting cases, so in the future, I'd better investigate more by considering all of the above factors as well as occurrence year, and remodel.

7. Possible Sources of Bias

According to the NYPD Shooting by Victim Age Group plot, Brooklyn has the most shooting cases, followed by Bronx, Queens, Manhattan, and Staten Island. But it does not necessarily mean Brooklyn is the most dangerous borough in New York City, maybe it is just because there is a much more population in Brooklyn. After considering the population, the rank might change.

8. Session Info

```
sessionInfo()
```

```
## R version 4.2.0 (2022-04-22)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.3.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
```

```
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## 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_2.5.0      colorspace_2.0-3
##  [5] vctrs_0.4.1      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_4.2.0   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_4.2.0
## [45] cli_3.2.0        tools_4.2.0      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_4.2.0
```