NYPD Shooting Data Analysis

Andrew Savala

2025-03-22

Overview

In this project I will be analyzing historic NYPD shooting data from 2006 to 2023. The shooting data comes from the five boroughs of New York City: Manhattan, Brooklyn, Queens, the Bronx, and Staten Island. My goal is to identify any underlying trends in the data that help better understand the shooting activity varies by borough.

Step 1: Import NYPD Shooting Data

```r
# Read shooting data from NYC Open Data
shooting_data <-
 ↪  read.csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
```

Step 2: Tidy and Transform Data

Examine Our Data

```r
# Display our raw shooting data
str(shooting_data)
```

```
## 'data.frame':    28562 obs. of  21 variables:
##  $ INCIDENT_KEY           : int  231974218 177934247 255028563 25384540 72616285 85875439 79780323 8
##  $ OCCUR_DATE             : chr  "08/09/2021" "04/07/2018" "12/02/2022" "11/19/2006" ...
##  $ OCCUR_TIME             : chr  "01:06:00" "19:48:00" "22:57:00" "01:50:00" ...
##  $ BORO                   : chr  "BRONX" "BROOKLYN" "BRONX" "BROOKLYN" ...
##  $ LOC_OF_OCCUR_DESC      : chr  "" "" "OUTSIDE" "" ...
##  $ PRECINCT               : int  40 79 47 66 46 42 71 69 75 69 ...
##  $ JURISDICTION_CODE      : int  0 0 0 0 0 2 0 2 0 0 ...
##  $ LOC_CLASSFCTN_DESC     : chr  "" "" "STREET" "" ...
##  $ LOCATION_DESC          : chr  "" "" "GROCERY/BODEGA" "PVT HOUSE" ...
##  $ STATISTICAL_MURDER_FLAG: chr  "false" "true" "false" "true" ...
##  $ PERP_AGE_GROUP         : chr  "" "25-44" "(null)" "UNKNOWN" ...
##  $ PERP_SEX               : chr  "" "M" "(null)" "U" ...
##  $ PERP_RACE              : chr  "" "WHITE HISPANIC" "(null)" "UNKNOWN" ...
##  $ VIC_AGE_GROUP          : chr  "18-24" "25-44" "25-44" "18-24" ...
##  $ VIC_SEX                : chr  "M" "M" "M" "M" ...
##  $ VIC_RACE               : chr  "BLACK" "BLACK" "BLACK" "BLACK" ...
##  $ X_COORD_CD             : num  1006343 1000083 1020691 985107 1009854 ...
##  $ Y_COORD_CD             : num  234270 189065 257125 173350 247503 ...
##  $ Latitude               : num  40.8 40.7 40.9 40.6 40.8 ...
##  $ Longitude              : num  -73.9 -73.9 -73.9 -74 -73.9 ...
##  $ Lon_Lat                : chr  "POINT (-73.92019278899994 40.80967347200004)" "POINT (-73.94291302
```

Lets keep only the columns we're interested in.

```
# Drop columns we don't need
shooting_data <- shooting_data[, c("OCCUR_DATE", "BORO", "PRECINCT", "PERP_AGE_GROUP",
 ↪  "PERP_SEX", "PERP_RACE", "VIC_AGE_GROUP", "VIC_SEX", "VIC_RACE")]
```

```
# Convert OCCUR_DATE to a date
shooting_data$OCCUR_DATE <- as.Date(shooting_data$OCCUR_DATE, format="%m/%d/%Y")

summary(shooting_data$OCCUR_DATE)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2006-01-01" "2009-09-04" "2013-09-20" "2014-06-07" "2019-09-29" "2023-12-29"
```

Check For Missing Data

Lets start by checking for missing values in our data.

```
# Check for missing values in our data
colSums(is.na(shooting_data)) / nrow(shooting_data) * 100
```

```
##     OCCUR_DATE           BORO       PRECINCT PERP_AGE_GROUP       PERP_SEX
##              0              0              0              0              0
##      PERP_RACE  VIC_AGE_GROUP        VIC_SEX       VIC_RACE
##              0              0              0              0
```

It looks like there's no null values. However, we should check for empty strings as well.

```
# Check for empty strings in our data
colSums(shooting_data == "") / nrow(shooting_data) * 100
```

```
##     OCCUR_DATE           BORO       PRECINCT PERP_AGE_GROUP       PERP_SEX
##             NA        0.00000        0.00000       32.71480       32.59576
##      PERP_RACE  VIC_AGE_GROUP        VIC_SEX       VIC_RACE
##       32.59576        0.00000        0.00000        0.00000
```

PERP_AGE_GROUP, PERP_SEX and PERP_RACE also all have roughly 33% missing values. Let's take a closer look at the values in these columns.

```
# Look at unique values in PERP_AGE_GROUP
unique(shooting_data$PERP_AGE_GROUP)
```

```
##  [1] ""        "25-44"   "(null)"  "UNKNOWN" "18-24"   "<18"     "45-64"  
##  [8] "65+"     "1028"    "1020"    "940"     "224"    
```

OK there's some interesting stuff going on with the values in PERP_AGE_GROUP. There are some values that are clearly not valid ages. Lets just treat all of these values as UNKNOWN.

```r
unknown_values <- c("", "1020", "1080", "(null)", "1028", "940", "224")
shooting_data$PERP_AGE_GROUP <- ifelse(
  shooting_data$PERP_AGE_GROUP %in% unknown_values,
  "UNKNOWN",
  shooting_data$PERP_AGE_GROUP
)

unique(shooting_data$PERP_AGE_GROUP)
```

```
## [1] "UNKNOWN" "25-44"   "18-24"   "<18"     "45-64"   "65+"    
```

Lets examine the PERP_SEX column.

```r
# Look at unique values in PERP_SEX
unique(shooting_data$PERP_SEX)
```

```
## [1] ""       "M"      "(null)" "U"      "F"     
```

Same thing with PERP_SEX. There seem to be multiple labels for unknown values. Lets just treat all of these values as U.

```r
unknown_values <- c("", "(null)", "UNKNOWN")
shooting_data$PERP_SEX <- ifelse(
  shooting_data$PERP_SEX %in% unknown_values,
  "U",
  shooting_data$PERP_SEX
)

unique(shooting_data$PERP_SEX)
```

```
## [1] "U" "M" "F"
```

Lets examine the PERP_RACE column.

```r
# Look at unique values in PERP_RACE
unique(shooting_data$PERP_RACE)
```

```
## [1] ""                               "WHITE HISPANIC"                
## [3] "(null)"                         "UNKNOWN"                       
## [5] "BLACK"                          "BLACK HISPANIC"                
## [7] "ASIAN / PACIFIC ISLANDER"       "WHITE"                         
## [9] "AMERICAN INDIAN/ALASKAN NATIVE"
```

Lets do the same thing and consolidate the unknown into a single value.

```r
unknown_values <- c("", "(null)")
shooting_data$PERP_RACE <- ifelse(
  shooting_data$PERP_RACE %in% unknown_values,
  "UNKNOWN",
  shooting_data$PERP_RACE
)

unique(shooting_data$PERP_RACE)
```

```
## [1] "UNKNOWN"                        "WHITE HISPANIC"
## [3] "BLACK"                          "BLACK HISPANIC"
## [5] "ASIAN / PACIFIC ISLANDER"       "WHITE"
## [7] "AMERICAN INDIAN/ALASKAN NATIVE"
```

Lets do the same for the victim columns.

```r
# Look at unique values in VIC_AGE_GROUP
unique(shooting_data$VIC_AGE_GROUP)
```

```
## [1] "18-24"   "25-44"   "<18"     "45-64"   "65+"     "UNKNOWN" "1022"
```

Lets just treat the unusual 1022 value as UNKNOWN.

```r
unknown_values <- c("1022")
shooting_data$VIC_AGE_GROUP <- ifelse(
  shooting_data$VIC_AGE_GROUP %in% unknown_values,
  "UNKNOWN",
  shooting_data$VIC_AGE_GROUP
)

unique(shooting_data$VIC_AGE_GROUP)
```

```
## [1] "18-24"   "25-44"   "<18"     "45-64"   "65+"     "UNKNOWN"
```

```r
# Look at unique values in VIC_SEX
unique(shooting_data$VIC_SEX)
```

```
## [1] "M" "F" "U"
```

Victim sex data looks acceptable.

```r
# Look at unique values in VIC_RACE
unique(shooting_data$VIC_RACE)
```

```
## [1] "BLACK"                          "WHITE HISPANIC"
## [3] "BLACK HISPANIC"                 "ASIAN / PACIFIC ISLANDER"
## [5] "WHITE"                          "UNKNOWN"
## [7] "AMERICAN INDIAN/ALASKAN NATIVE"
```

Victim race data looks acceptable.

One more time, lets look at our data.

```
head(shooting_data)
```

```
##   OCCUR_DATE     BORO PRECINCT PERP_AGE_GROUP PERP_SEX      PERP_RACE
## 1 2021-08-09    BRONX       40        UNKNOWN        U        UNKNOWN
## 2 2018-04-07 BROOKLYN       79          25-44        M WHITE HISPANIC
## 3 2022-12-02    BRONX       47        UNKNOWN        U        UNKNOWN
## 4 2006-11-19 BROOKLYN       66        UNKNOWN        U        UNKNOWN
## 5 2010-05-09    BRONX       46          25-44        M          BLACK
## 6 2012-07-22    BRONX       42          18-24        M          BLACK
##   VIC_AGE_GROUP VIC_SEX VIC_RACE
## 1         18-24       M    BLACK
## 2         25-44       M    BLACK
## 3         25-44       M    BLACK
## 4         18-24       M    BLACK
## 5           <18       F    BLACK
## 6         18-24       M    BLACK
```

This is feeling a lot better. Lets factorize all of our character columns.

Factorize Columns

```
# Factorize all character columns
shooting_data <- shooting_data %>%
  mutate_if(is.character, as.factor)
```

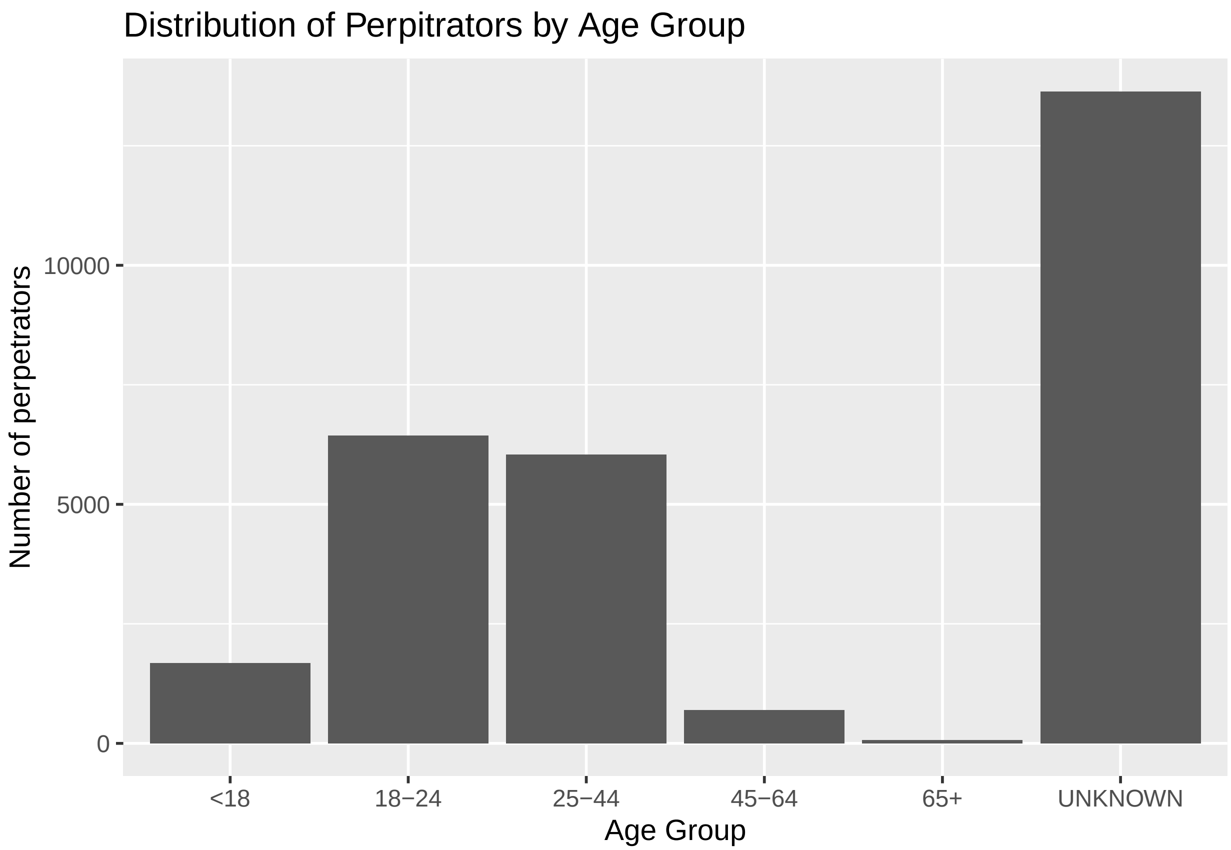
```
# Review our data
str(shooting_data)
```

```
## 'data.frame':    28562 obs. of  9 variables:
##  $ OCCUR_DATE    : Date, format: "2021-08-09" "2018-04-07" ...
##  $ BORO          : Factor w/ 5 levels "BRONX","BROOKLYN",..: 1 2 1 2 1 1 2 2 2 2 ...
##  $ PRECINCT      : int  40 79 47 66 46 42 71 69 75 69 ...
##  $ PERP_AGE_GROUP: Factor w/ 6 levels "<18","18-24",..: 6 3 6 6 3 2 6 6 3 2 ...
##  $ PERP_SEX      : Factor w/ 3 levels "F","M","U": 3 2 3 3 2 2 3 3 2 2 ...
##  $ PERP_RACE     : Factor w/ 7 levels "AMERICAN INDIAN/ALASKAN NATIVE",..: 5 7 5 5 3 3 5 5 3 3 ...
##  $ VIC_AGE_GROUP : Factor w/ 6 levels "<18","18-24",..: 2 3 3 2 1 2 3 3 3 2 ...
##  $ VIC_SEX       : Factor w/ 3 levels "F","M","U": 2 2 2 2 1 2 2 2 2 2 ...
##  $ VIC_RACE      : Factor w/ 7 levels "AMERICAN INDIAN/ALASKAN NATIVE",..: 3 3 3 3 3 3 3 7 3 3 ...
```

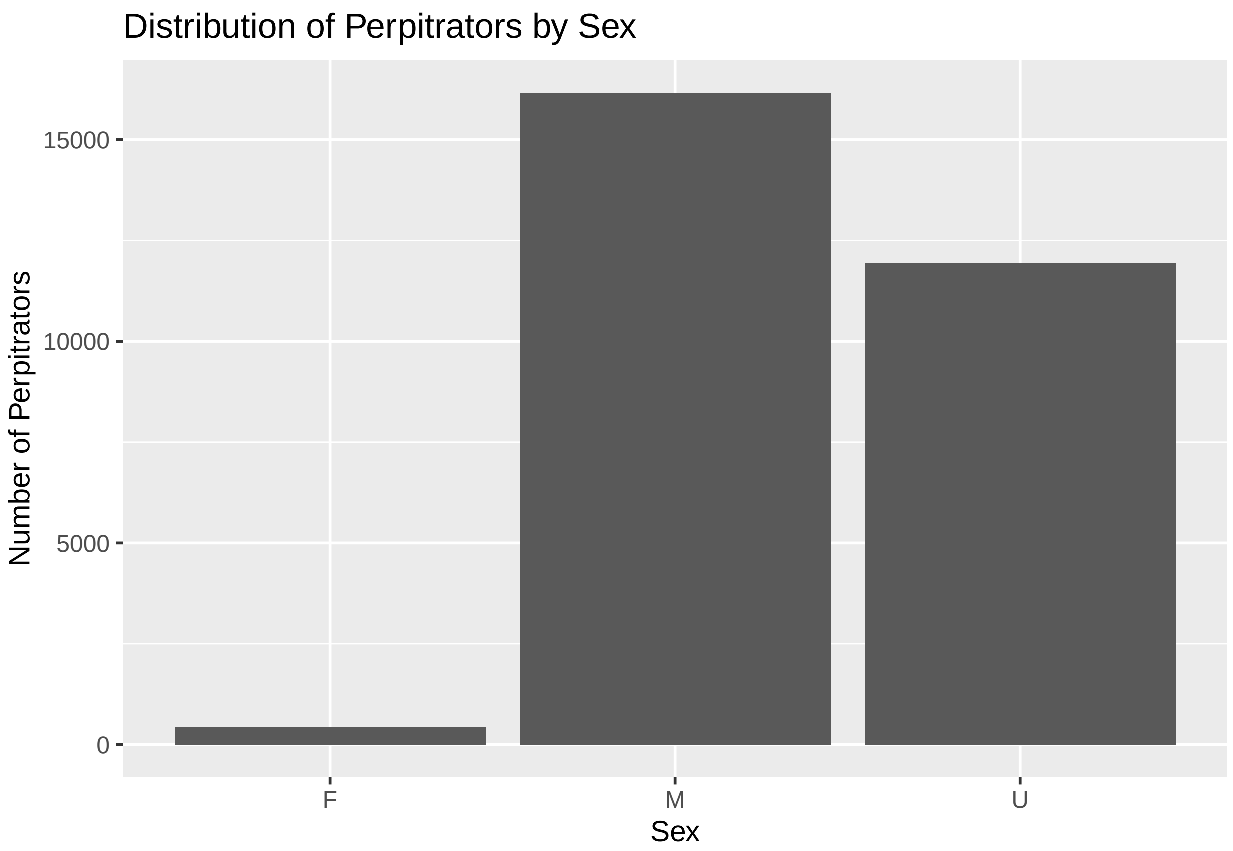
Step 3: Add Visualizations and Analysis

Visualizations

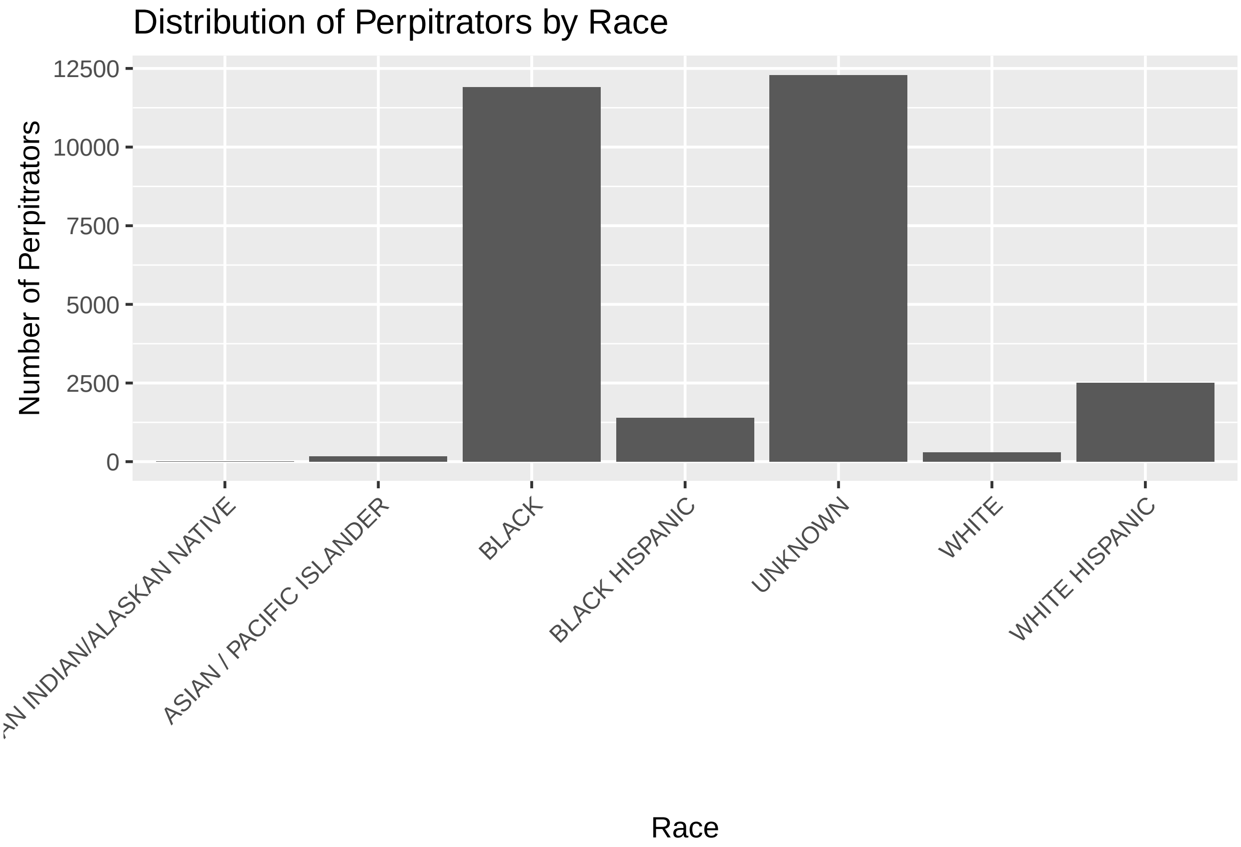
```
# Distribution of the perpetrators by age group PERP_AGE_GROUP
shooting_data %>%
  count(PERP_AGE_GROUP) %>%
  ggplot(aes(x = PERP_AGE_GROUP, y = n)) +
  geom_bar(stat = "identity") +
  labs(title = "Distribution of Perpitrators by Age Group",
       x = "Age Group",
       y = "Number of perpetrators")
```



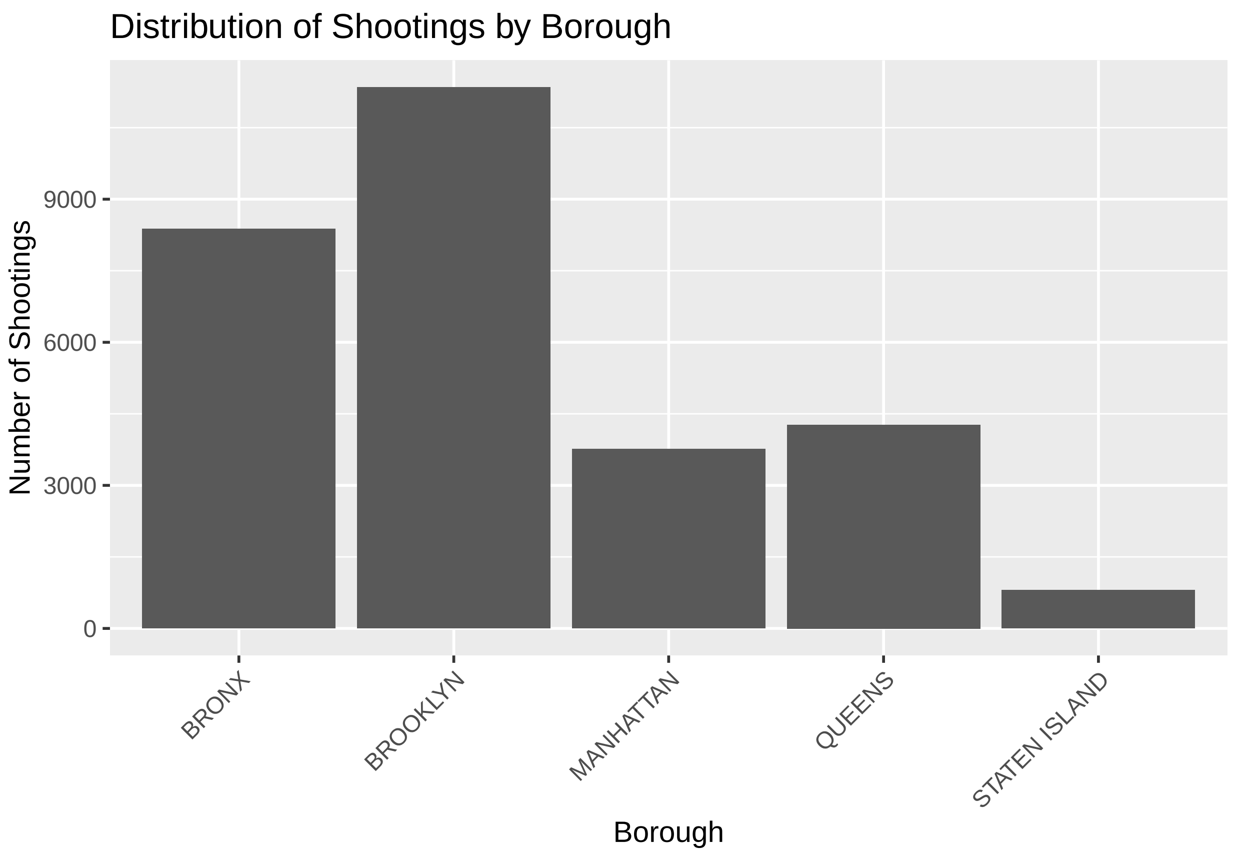
```
# Distribution of the perpetrators by sex PERP_SEX
shooting_data %>%
  count(PERP_SEX) %>%
  ggplot(aes(x = PERP_SEX, y = n)) +
  geom_bar(stat = "identity") +
  labs(title = "Distribution of Perpitrators by Sex",
       x = "Sex",
       y = "Number of Perpitrators")
```



```
# Distribution of the perpetrators by race PERP_RACE
shooting_data %>%
  count(PERP_RACE) %>%
  ggplot(aes(x = PERP_RACE, y = n)) +
  geom_bar(stat = "identity") +
  labs(title = "Distribution of Perpitrators by Race",
       x = "Race",
       y = "Number of Perpitrators")  +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
# Distribution of shootings by borough BORO
shooting_data %>%
  count(BORO) %>%
  ggplot(aes(x = BORO, y = n)) +
  geom_bar(stat = "identity") +
  labs(title = "Distribution of Shootings by Borough",
       x = "Borough",
       y = "Number of Shootings")  +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Analysis

Lets take a closer look at the shootings by borough over time and see what the trends looks like.

```
# Create a data frame of shootings over time grouped by borough
shootings_borough <- shooting_data %>%
  group_by(BORO, OCCUR_DATE) %>%
  summarise(shootings = n()) %>%
  select(BORO, OCCUR_DATE, shootings) %>%
  ungroup()
```

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

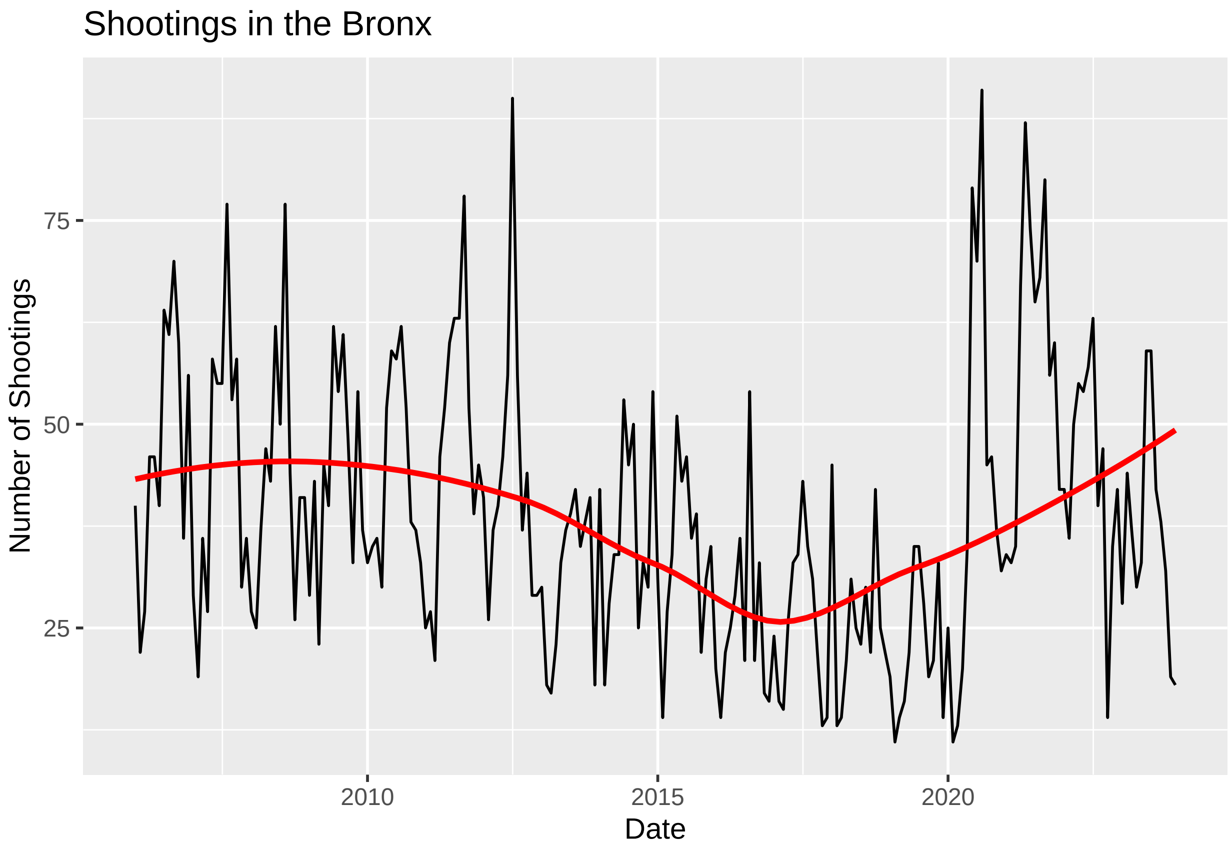
```
shootings_borough
```

```
## # A tibble: 13,744 x 3
##    BORO  OCCUR_DATE shootings
##    <fct> <date>         <int>
##  1 BRONX 2006-01-01         2
##  2 BRONX 2006-01-04         1
##  3 BRONX 2006-01-05         2
##  4 BRONX 2006-01-06         3
##   5 BRONX 2006-01-09         4
##   6 BRONX 2006-01-10         1
##   7 BRONX 2006-01-13         2
##   8 BRONX 2006-01-14         2
##   9 BRONX 2006-01-15         2
##  10 BRONX 2006-01-16         2
## # i 13,734 more rows
```

```
# Bronx
shootings_bronx_monthly <- shootings_borough %>%
  filter(BORO == "BRONX") %>%
  mutate(month = floor_date(OCCUR_DATE, unit = "month")) %>%
  group_by(month) %>%
  summarise(shootings = sum(shootings), .groups = "drop")

ggplot(shootings_bronx_monthly, aes(x = month, y = shootings)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Shootings in the Bronx",
       x = "Date",
       y = "Number of Shootings")
```

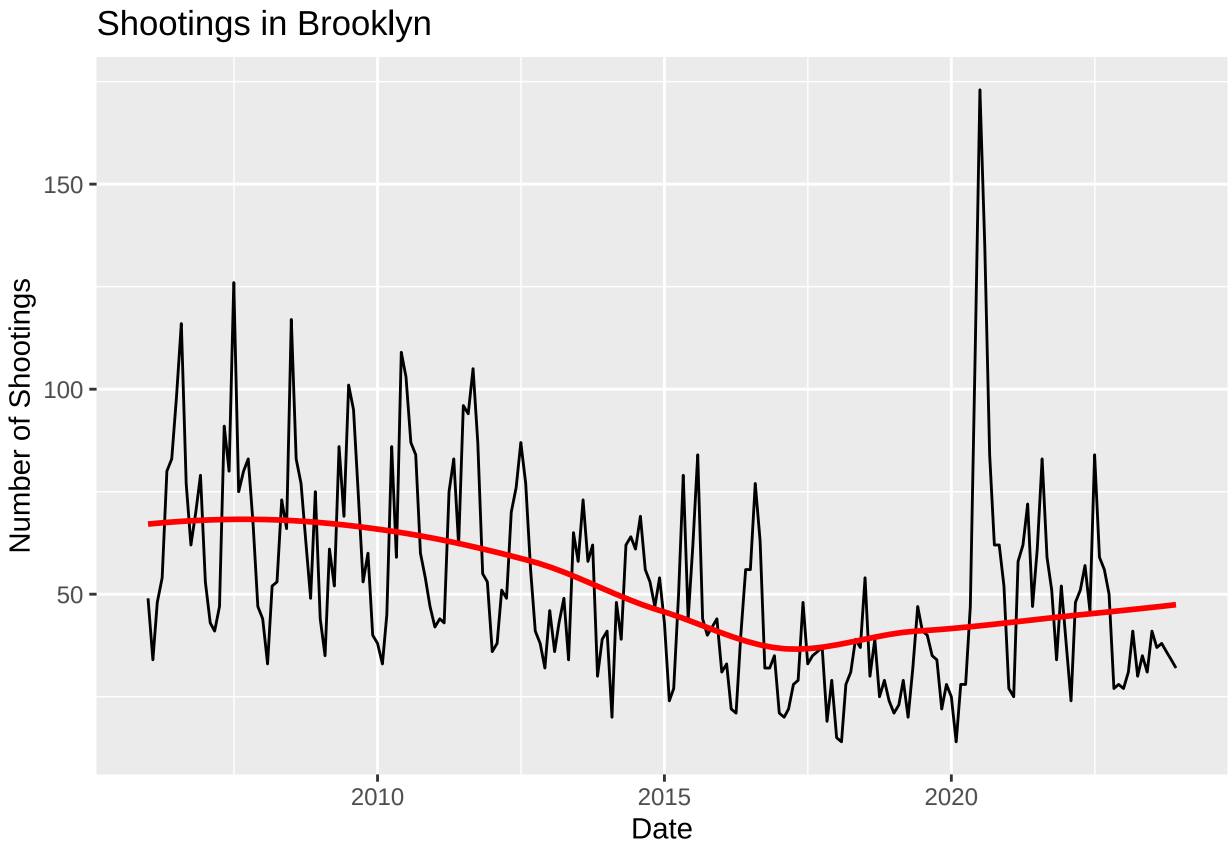
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Brooklyn
shootings_brooklyn_monthly <- shootings_borough %>%
  filter(BORO == "BROOKLYN") %>%
  mutate(month = floor_date(OCCUR_DATE, unit = "month")) %>%
  group_by(month) %>%
  summarise(shootings = sum(shootings), .groups = "drop")

ggplot(shootings_brooklyn_monthly, aes(x = month, y = shootings)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Shootings in Brooklyn",
       x = "Date",
       y = "Number of Shootings")
```

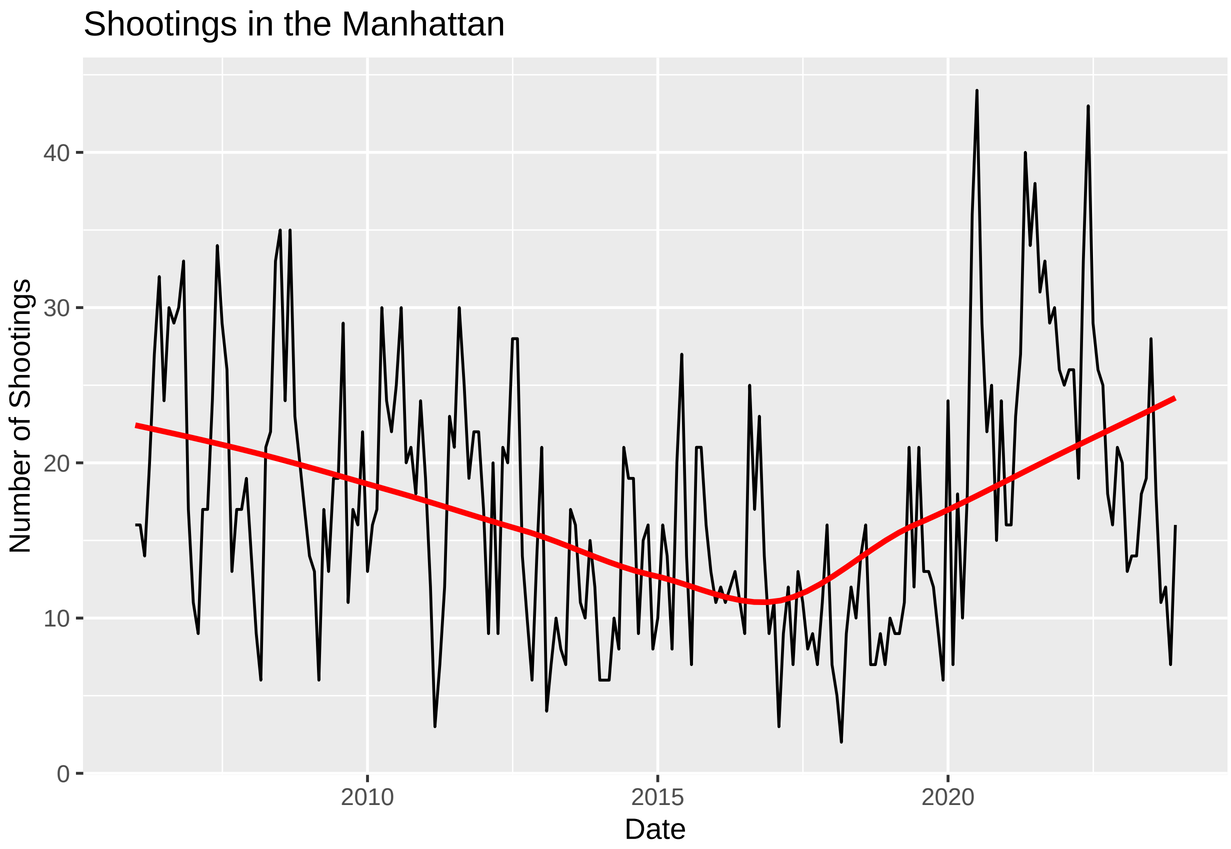
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Manhattan
shootings_manhattan_monthly <- shootings_borough %>%
  filter(BORO == "MANHATTAN") %>%
  mutate(month = floor_date(OCCUR_DATE, unit = "month")) %>%
  group_by(month) %>%
  summarise(shootings = sum(shootings), .groups = "drop")

ggplot(shootings_manhattan_monthly, aes(x = month, y = shootings)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Shootings in the Manhattan",
       x = "Date",
       y = "Number of Shootings")
```

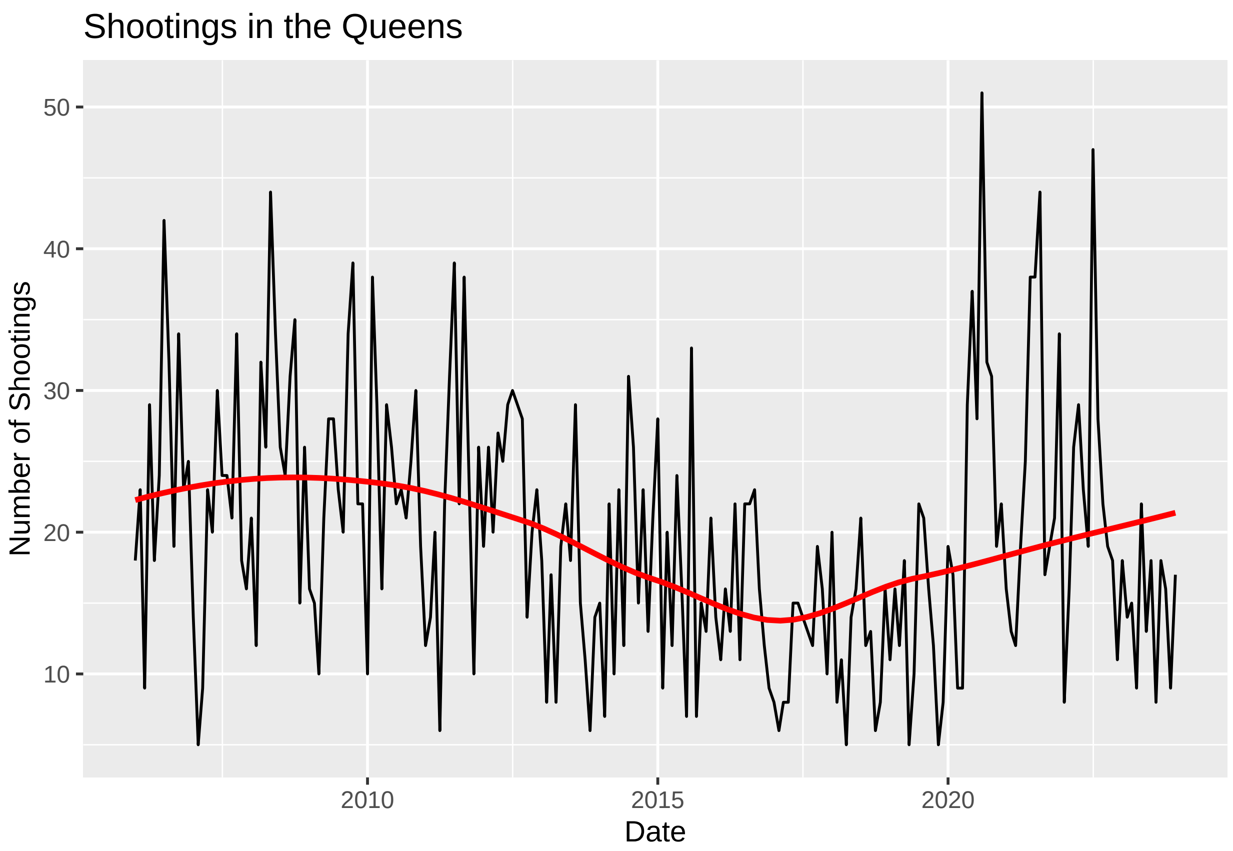
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Queens
shootings_queens_monthly <- shootings_borough %>%
  filter(BORO == "QUEENS") %>%
  mutate(month = floor_date(OCCUR_DATE, unit = "month")) %>%
  group_by(month) %>%
  summarise(shootings = sum(shootings), .groups = "drop")

ggplot(shootings_queens_monthly, aes(x = month, y = shootings)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Shootings in the Queens",
       x = "Date",
       y = "Number of Shootings")
```

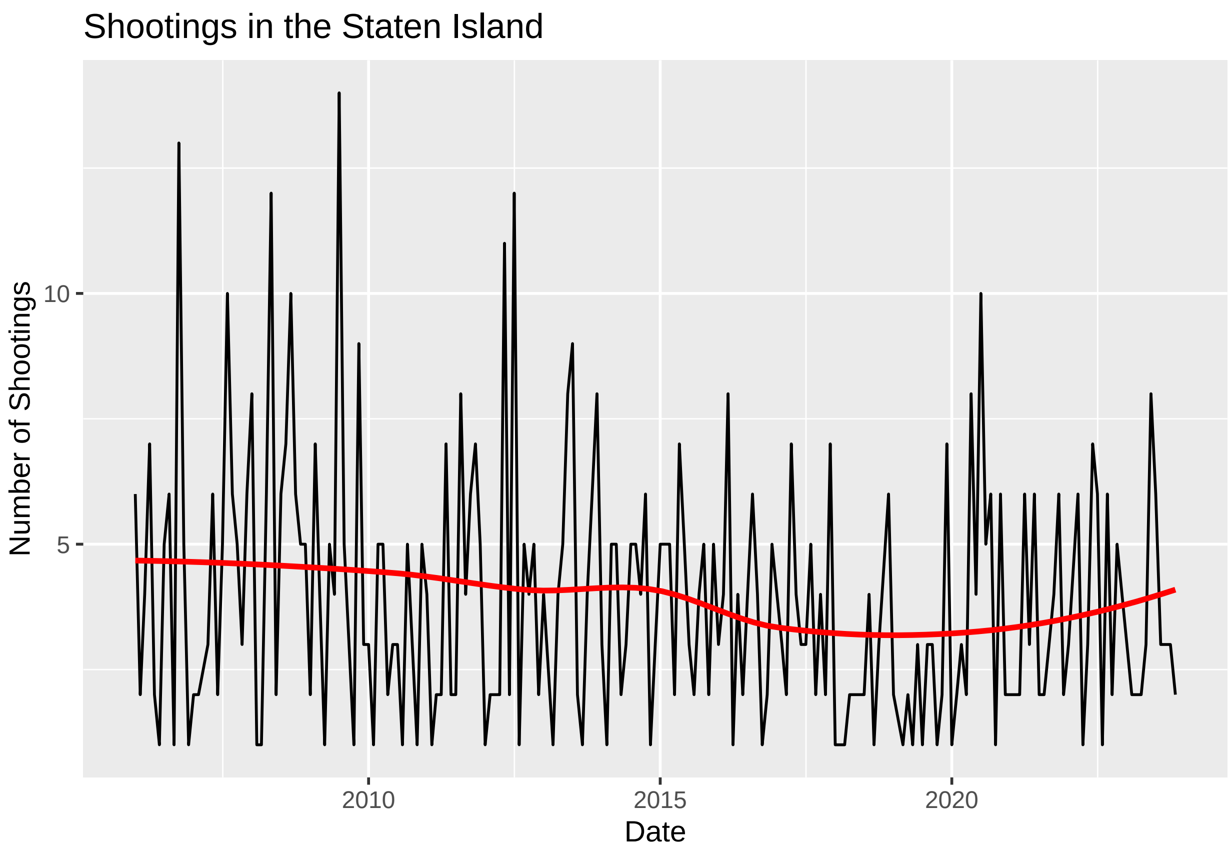
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Staten Island
shootings_statenisland_monthly <- shootings_borough %>%
  filter(BORO == "STATEN ISLAND") %>%
  mutate(month = floor_date(OCCUR_DATE, unit = "month")) %>%
  group_by(month) %>%
  summarise(shootings = sum(shootings), .groups = "drop")

ggplot(shootings_statenisland_monthly, aes(x = month, y = shootings)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Shootings in the Staten Island",
       x = "Date",
       y = "Number of Shootings")
```

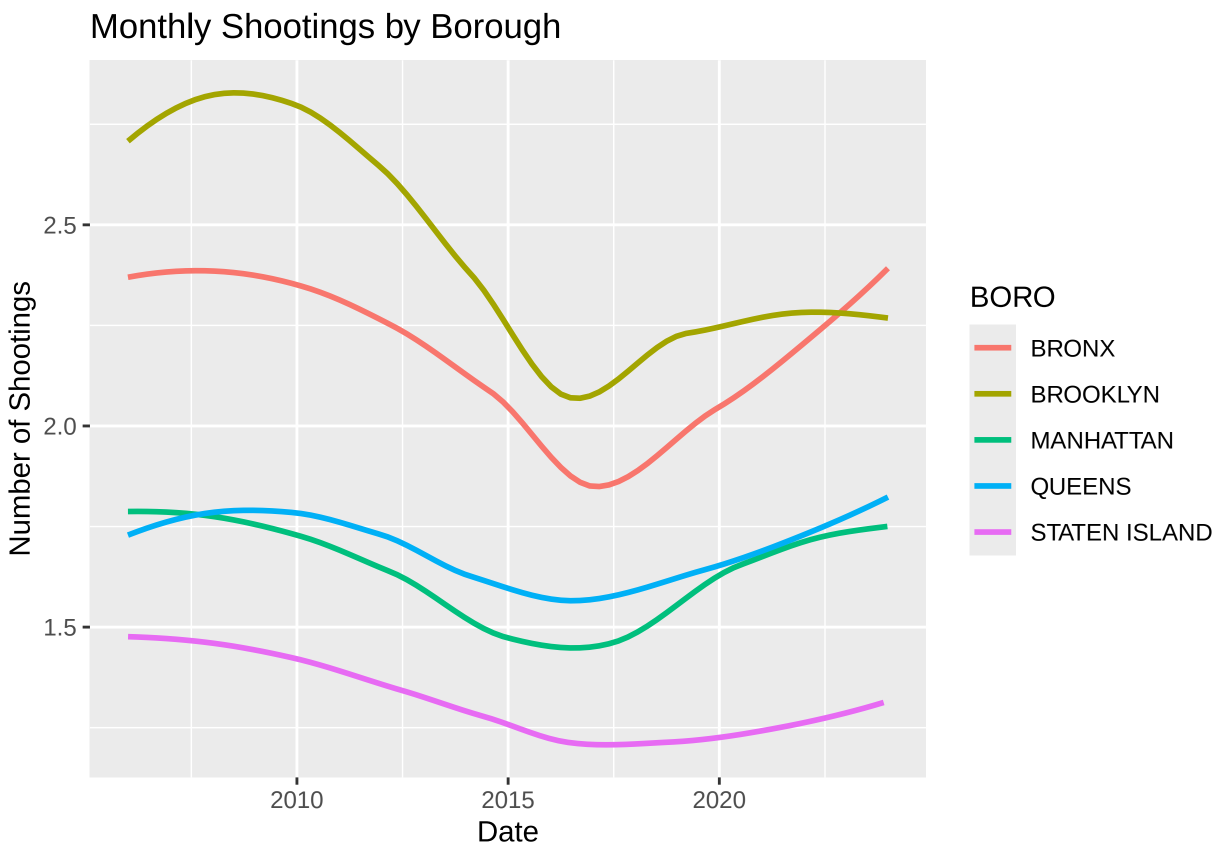
```
## `geom_smooth()` using formula = 'y ~ x'
```



There seems to be a trend here. Let's overlay the trends for all the boroughs to see if there's a pattern.

```
ggplot(shootings_borough, aes(x = OCCUR_DATE, y = shootings, color = BORO)) +
  geom_smooth(se = FALSE, method = "loess") +
  labs(title = "Monthly Shootings by Borough",
       x = "Date",
       y = "Number of Shootings")
```

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



Look at that! They all take a dip around the same time. It would be interesting to research what was happening in the city at that time.

Model

Lets try and predict the number of shootings in each borough by year.

```
shooting_data$Year <- as.numeric(format(shooting_data$OCCUR_DATE, "%Y"))

# Group by borough and year
shooting_borough_yearly <- shooting_data %>%
  group_by(BORO, Year) %>%
  summarise(Shootings = n(), .groups = "drop")

head(shooting_borough_yearly)
```

```
## # A tibble: 6 x 3
##   BORO   Year Shootings
##   <fct> <dbl>     <int>
## 1 BRONX  2006       568
## 2 BRONX  2007       533
## 3 BRONX  2008       520
## 4 BRONX  2009       529
## 5 BRONX  2010       525
## 6 BRONX  2011       571
```

Lets fit a simple linear model where the number of shootings is a function of the year and borough.

```r
# Fit a simple linear model
mod <- lm(Shootings ~ Year + BORO, data = shooting_borough_yearly)
summary(mod)
```

```
##
## Call:
## lm(formula = Shootings ~ Year + BORO, data = shooting_borough_yearly)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -255.901  -47.627   -0.778   43.398  280.991
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       14512.365   3943.941   3.680 0.000411 ***
## Year                 -6.973      1.958  -3.562 0.000610 ***
## BOROBROOKLYN        165.000     32.119   5.137 1.78e-06 ***
## BOROMANHATTAN      -256.333     32.119  -7.981 6.63e-12 ***
## BOROQUEENS         -228.056     32.119  -7.100 3.70e-10 ***
## BOROSTATEN ISLAND  -420.500     32.119 -13.092  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.36 on 84 degrees of freedom
## Multiple R-squared:  0.8347, Adjusted R-squared:  0.8249
## F-statistic: 84.84 on 5 and 84 DF,  p-value: < 2.2e-16
```

Lets add our predicted shootings back into our data frame.

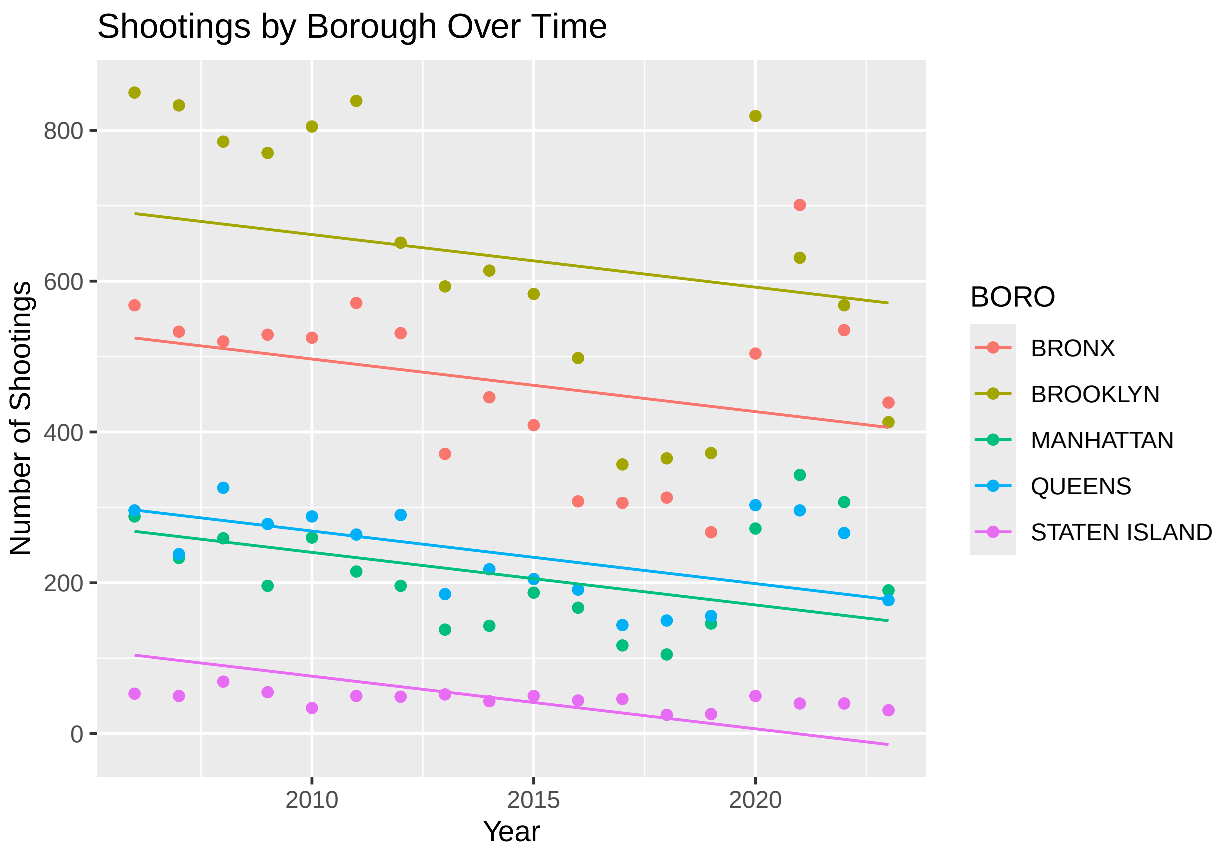
```r
shooting_borough_yearly$Shootings_Pred <- predict(mod, shooting_borough_yearly)

head(shooting_borough_yearly)
```

```
## # A tibble: 6 x 4
##   BORO   Year Shootings Shootings_Pred
##   <fct> <dbl>     <int>          <dbl>
## 1 BRONX  2006       568           525.
## 2 BRONX  2007       533           518.
## 3 BRONX  2008       520           511.
## 4 BRONX  2009       529           504.
## 5 BRONX  2010       525           497.
## 6 BRONX  2011       571           490.
```

Lets visualize our predictions.

```r
ggplot(shooting_borough_yearly, aes(x = Year, y = Shootings, color = BORO)) +
  geom_point() +
  geom_line(aes(y = Shootings_Pred)) +
  labs(title = "Shootings by Borough Over Time",
       x = "Year",
       y = "Number of Shootings")
```



The dots show the actual number of shootings and the lines show the predicted number of shootings. It looks like the model is a better fit for Manhattan and Staten Island. The model also has a negative correlation with the year. This is interesting because it suggests that the number of shootings is decreasing over time.

Step 4: Add Bias Identification / Conclusion

I think this would have been particularly relevant if I had been investigating perpetrators. Especially since there were so many missing values. This makes sense because they wouldn't always be able to catch the shooter or get that information from the victim.

I live in a city (Fresno, CA) where there is a lot of crime, and shootings are a common occurrence. I think it would be interesting to compare the data from Fresno to the data from NY to see if there are any similarities or differences. I am picturing NY through the lens of my hometown and I think it would be interesting to see if the data supports that.

Session Info

```
sessionInfo()
```

```
## R version 4.4.3 (2025-02-28 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
```

```
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] corrplot_0.95   lubridate_1.9.4 ggplot2_3.5.1   dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] Matrix_1.7-2      gtable_0.3.6      compiler_4.4.3    tidyselect_1.2.1
##  [5] splines_4.4.3     scales_1.3.0      yaml_2.3.10       fastmap_1.2.0
##  [9] lattice_0.22-6    R6_2.6.1          labeling_0.4.3    generics_0.1.3
## [13] knitr_1.49        tibble_3.2.1      munsell_0.5.1     pillar_1.10.1
## [17] rlang_1.1.5       utf8_1.2.4        xfun_0.51         timechange_0.3.0
## [21] cli_3.6.4         withr_3.0.2       magrittr_2.0.3    mgcv_1.9-1
## [25] digest_0.6.37     grid_4.4.3        rstudioapi_0.17.1 lifecycle_1.0.4
## [29] nlme_3.1-167      vctrs_0.6.5       evaluate_1.0.3    glue_1.8.0
## [33] farver_2.1.2      colorspace_2.1-1  rmarkdown_2.29    tools_4.4.3
## [37] pkgconfig_2.0.3   htmltools_0.5.8.1
```