NYPD Shooting Data

2023-10-11

Introduction and Data Summary

I'm evaluating this gun violence data to find patterns in times and locations to help Police precincts be better managed and equipped to deal with dangerous scenarios while also informing citizens about the safest times to be outside in their wonderful city. The recorded shooting data is from https://catalog.data.gov/ and is updated to the end of the previous calendar year. The data I pulled is from 2006-01-01 to 2022-12-31 and there are approximately 27 thousand records of a shooting incident. Each record represents a shooting incident in NYC and includes information about the event, the location and time of occurrence. In addition, information related to suspect and victim demographics is also included. I'm going to strictly focus on location and time analysis of the data so I can remove the rest of the columns I do not plan to utilize.

```
library(sf)
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1; sf_use_s2() is TRUE
```

```
library(lubridate)
```

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

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.3     v readr   2.1.4
## v forcats 1.0.0     v stringr 1.5.0
## v ggplot2 3.4.4     v tibble  3.2.1
## v purrr   1.0.2     v tidyr   1.3.0
```

```
## -- 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
```

```
shooting_data_csv <- read_csv("NYPD_Shooting_Incident_Data__Historic_.csv")
```

```
## 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.
```

```
summary(shooting_data_csv)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:27312       Length:27312       Length:27312
##  1st Qu.: 63860880   Class :character   Class1:hms         Class :character
##  Median : 90372218   Mode  :character   Class2:difftime    Mode  :character
##  Mean   :120860536                      Mode  :numeric
##  3rd Qu.:188810230
##  Max.   :261190187
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character
##                     Mean   : 65.64   Mean   :0.3269
##                     3rd Qu.: 81.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##                                      NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312       Mode :logical           Length:27312
##  Class :character   FALSE:22046             Class :character
##  Mode  :character   TRUE :5266              Mode  :character
##
##
##
##
##    PERP_SEX           PERP_RACE          VIC_AGE_GROUP        VIC_SEX
##  Length:27312       Length:27312       Length:27312       Length:27312
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE            X_COORD_CD        Y_COORD_CD        Latitude
##  Length:27312       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000028   1st Qu.:182834   1st Qu.:40.67
##  Mode  :character   Median :1007731   Median :194487   Median :40.70
##                     Mean   :1009449   Mean   :208127   Mean   :40.74
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :10
##    Longitude        Lon_Lat
##  Min.   :-74.25   Length:27312
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :10
```

```
shooting_data <- shooting_data_csv %>%
  mutate(occur_date = mdy(OCCUR_DATE)) %>%
```

```
  select(BORO, occur_date, OCCUR_TIME) %>%
  mutate(count = 1)

max(shooting_data$occur_date)
```

```
## [1] "2022-12-31"
```

```
min(shooting_data$occur_date)
```

```
## [1] "2006-01-01"
```

```
#"https://s-media.nyc.gov/agencies/dcp/assets/files/zip/data-tools/bytes/nybb_23c.zip"

nyc <- st_read("nybb_23c/")
```

```
## Reading layer `nybb' from data source `/home/mattferguson/data/nybb_23c' using driver `ESRI Shapefil
## Simple feature collection with 5 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 913175.1 ymin: 120128.4 xmax: 1067383 ymax: 272844.3
## Projected CRS: NAD83 / New York Long Island (ftUS)
```

Tidy and Transform

I'm doing extract transform load manipulations to apply the appropriate data structures the their values. I'm modifying and cleaning up the data set by changing appropriate variables to factor and date types. I'm changing the OCCUR_DATE to a date type object. I'll also do a join to connect two separate tables.

```
shooting_data_hourly <- shooting_data %>%
  filter(occur_date > "2013-9-29") %>%
  mutate(occur_hour = lubridate::hour(OCCUR_TIME)) %>%
  mutate(count = 1) %>%
  group_by(occur_hour, BORO)  %>%
  summarise(count_sum = sum(count))
```

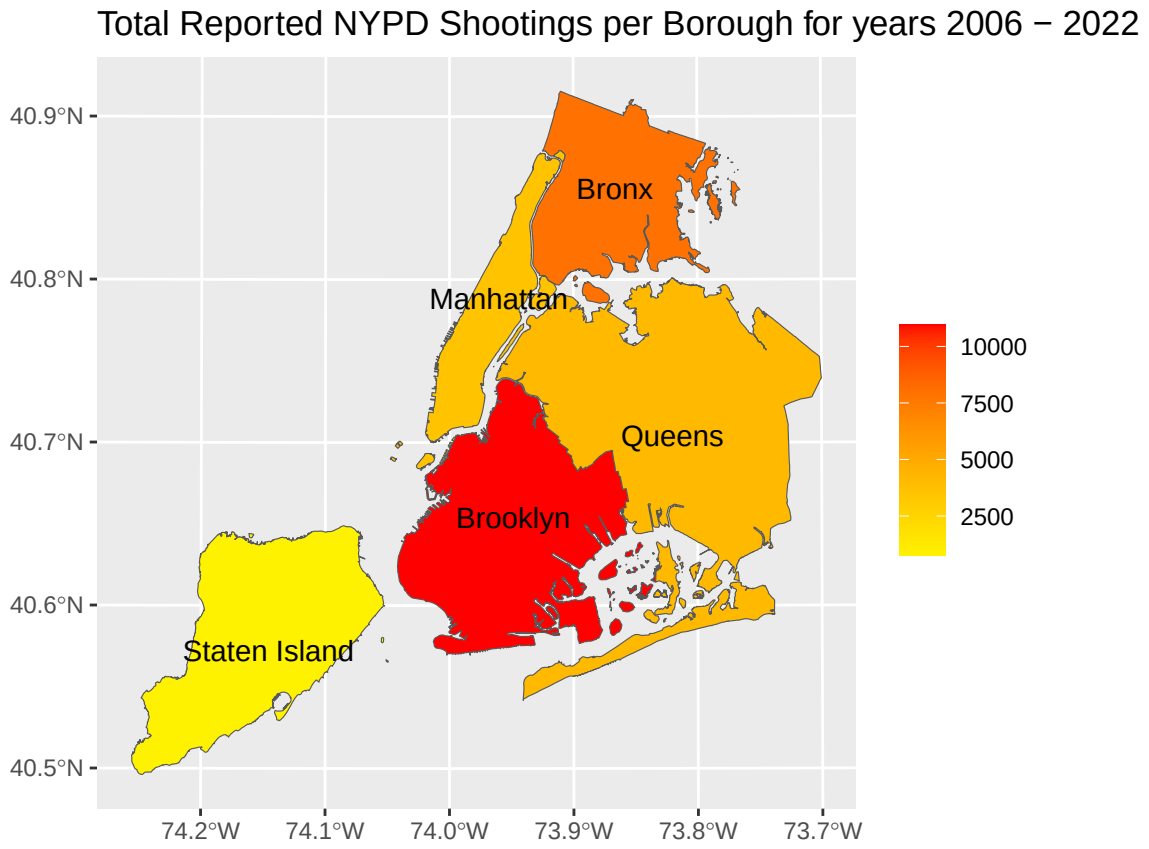
```
## `summarise()` has grouped output by 'occur_hour'. You can override using the
## `.groups` argument.
```

```
shooting_data_binned_boro <- shooting_data_csv %>%
  mutate(hour = format(strptime(shooting_data_csv$OCCUR_TIME, "%H:%M"), "%H:00")) %>%
  select(BORO, OCCUR_TIME, hour) %>%
  mutate(count = 1) %>%
  group_by(BORO) %>%
  summarise(count_sum = sum(count))

nyc <- nyc %>%
  mutate(BORO = toupper(BoroName)) %>%
  left_join(shooting_data_binned_boro, by = "BORO")

shooting_data_binned_monthly <- shooting_data_csv %>%
  mutate(occur_date = as.Date(OCCUR_DATE, "%m/%d/%Y")) %>%
  select(BORO, occur_date) %>%
  mutate(count = 1) %>%
  group_by(BORO, month = lubridate::floor_date(occur_date, "month")) %>%
  summarise(count_sum = sum(count))
```

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

Visualization and Analysis 1

```
nyc %>%
  ggplot() +
  geom_sf(aes(fill = count_sum)) +
  geom_sf_text(aes(label = BoroName)) +
  scale_fill_gradient2( mid = "yellow", high = "red") +
  theme(legend.position="right") +
  theme(legend.title=element_blank()) +
  labs(title = "Total Reported NYPD Shootings per Borough for years 2006 - 2022", y = NULL, x = NULL)
```



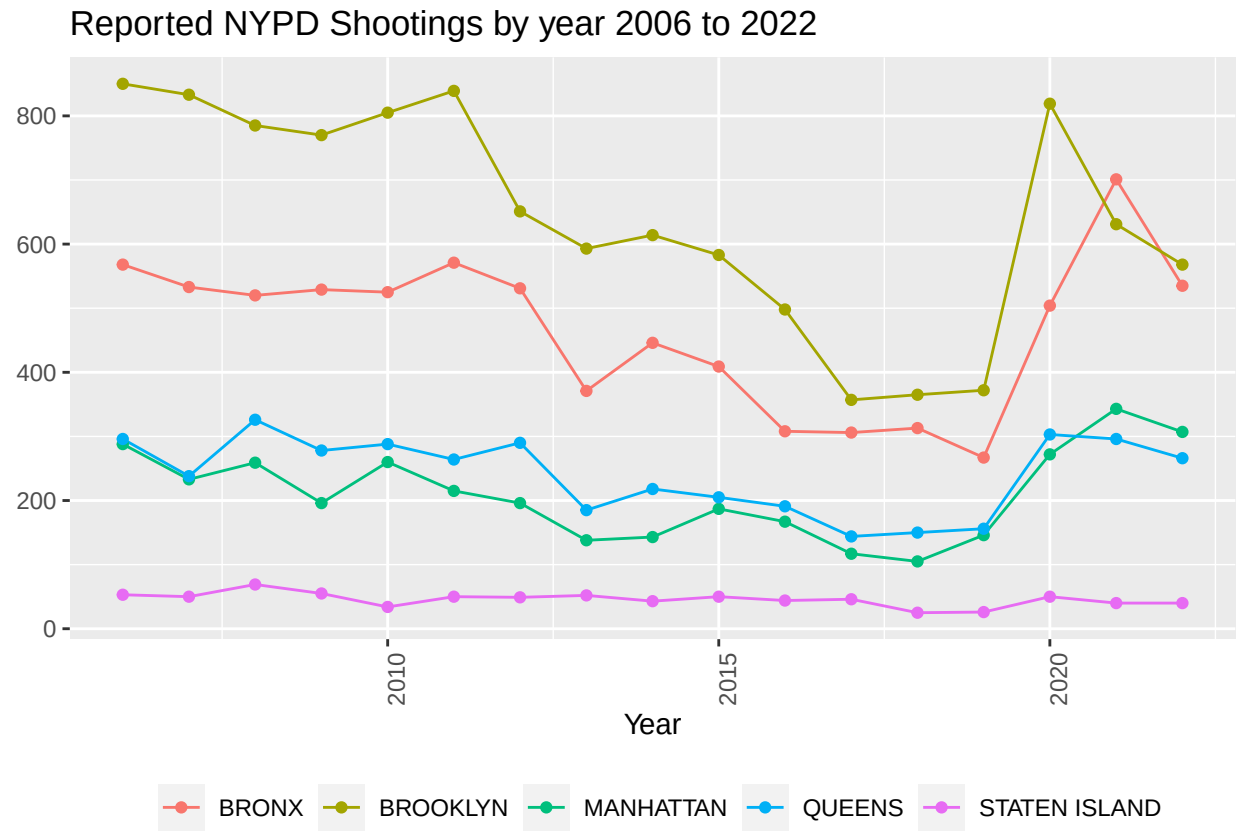
Brooklyn is the most likely borough of a reported shooting while Staten Island is the least likely borough of a reported shooting.

Visualization and Analysis 2

```
shooting_data %>%
  group_by(BORO, year = floor_date(occur_date, "year")) %>%
  summarise(count_sum = sum(count)) %>%
  ggplot(aes(x = year, y = count_sum, group = BORO)) +
  geom_line(aes(color = BORO)) +
```

```
  geom_point(aes(color = BORO)) +
  theme(legend.title=element_blank()) +
  theme(legend.position ="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Reported NYPD Shootings by year 2006 to 2022", y = NULL, x = "Year")
```

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



The chart looks like there is a downward slope in reported shootings from all boroughs until 2020 and then a large increase of reported shootings.

```
shooting_data_lm_prep <- shooting_data %>%
  filter(occur_date > "2012-9-29") %>%
  mutate(occur_month = month(occur_date)) %>%
  mutate(occur_year = year(occur_date)) %>%
  group_by(BORO, occur_year, occur_month) %>%
  summarise(count_sum = sum(count)) %>%
  ungroup()
```

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

```
year_month_mod <- lm( count_sum ~ occur_year + occur_month, data = shooting_data_lm_prep)

shooting_data_mod <- shooting_data_lm_prep %>%
  mutate(pred = predict(year_month_mod))
```

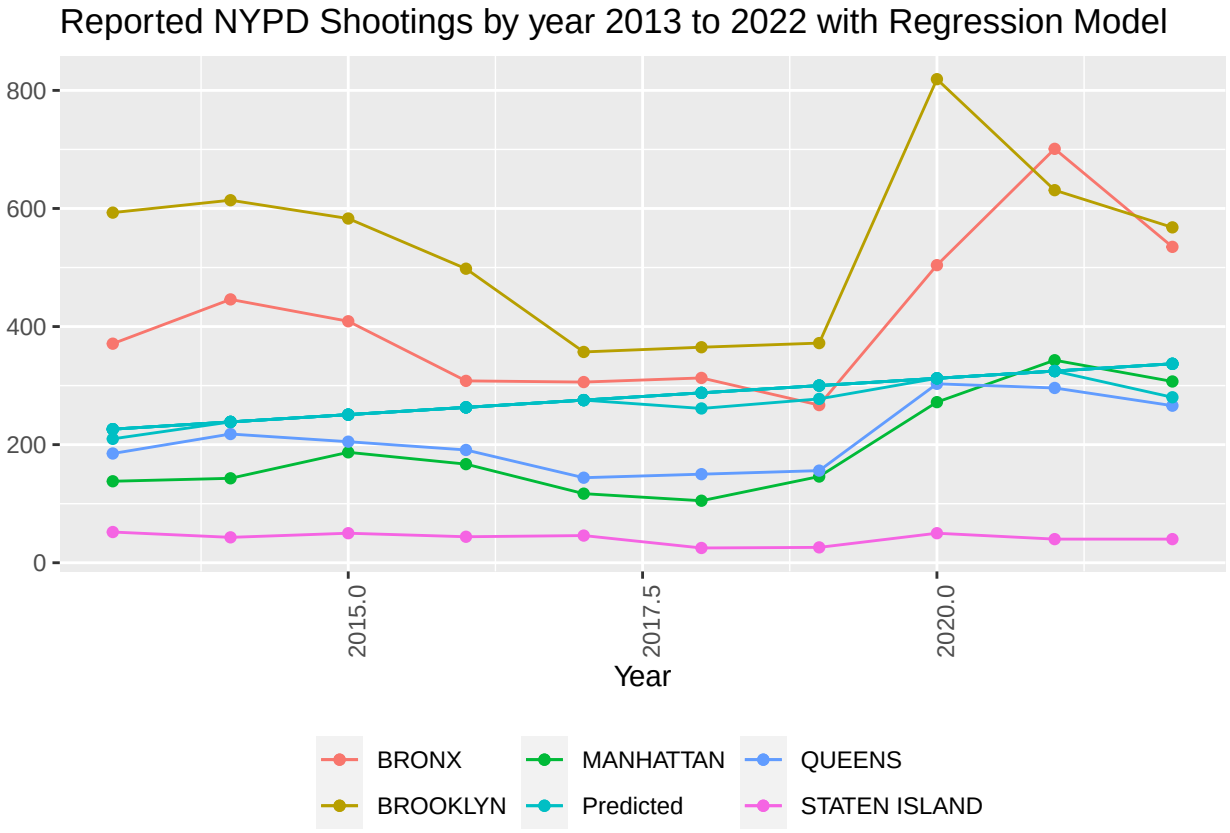
```
shooting_data_mod %>%
  filter(occur_month > 10) %>%
  group_by(BORO, occur_year) %>%
  summarise(count_sum_tot = sum(count_sum), count_pred = sum(pred))
```

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

```
## # A tibble: 55 x 4
## # Groups:   BORO [5]
##    BORO  occur_year count_sum_tot count_pred
##    <chr>      <dbl>         <dbl>      <dbl>
##  1 BRONX       2012            58       41.1
##  2 BRONX       2013            59       43.2
##  3 BRONX       2014            84       45.2
##  4 BRONX       2015            66       47.3
##  5 BRONX       2016            33       49.3
##  6 BRONX       2017            27       51.4
##  7 BRONX       2018            47       53.4
##  8 BRONX       2019            47       55.5
##  9 BRONX       2020            69       57.5
## 10 BRONX       2021           102       59.6
## # i 45 more rows
```

```
shooting_data_mod %>%
  filter(occur_year > 2012) %>%
  group_by(BORO, occur_year) %>%
  summarise(count_sum_tot = sum(count_sum), count_pred = sum(pred)) %>%
  ggplot(aes(x = occur_year, y = count_sum_tot, group = BORO)) +
  geom_line(aes(color = BORO)) +
  geom_point(aes(color = BORO)) +
  geom_line(aes(x = occur_year, y = count_pred, color = "Predicted")) +
  geom_point(aes(x = occur_year, y = count_pred, color = "Predicted") ) +
  theme(legend.title=element_blank()) +
  theme(legend.position ="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Reported NYPD Shootings by year 2013 to 2022 with Regression Model", y = NULL, x = "Yea
```

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



The linear regression model fitted to the windowed data for the years 2013 to 2020 has an upward trend of reported shootings with an increase of 1.0255 shootings per year.

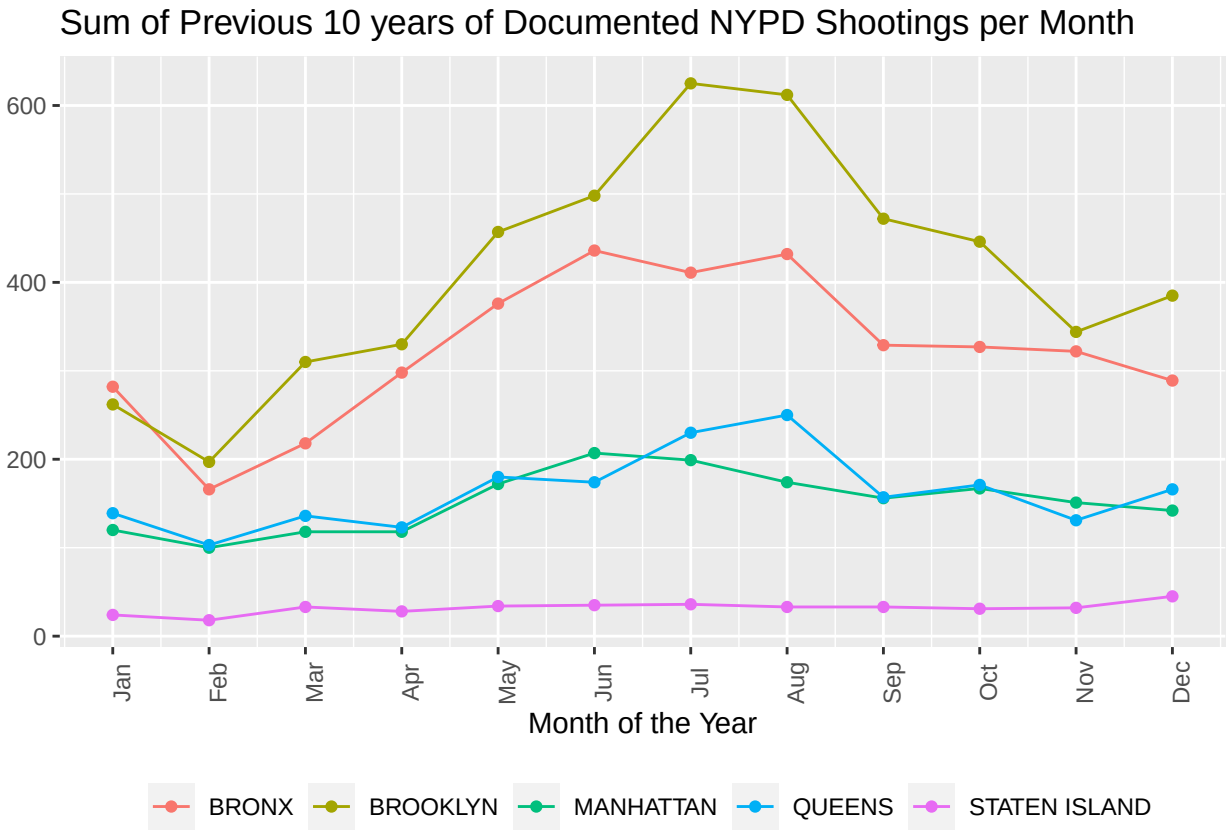
Visualization and Analysis 3

```
mymonths <- c("Jan","Feb","Mar",
              "Apr","May","Jun",
              "Jul","Aug","Sep",
              "Oct","Nov","Dec")

shooting_data_monthly <- shooting_data %>%
  filter(occur_date > "2013-9-29") %>%
  mutate(occur_month = month(occur_date)) %>%
  group_by(BORO, occur_month) %>%
  summarise(count_sum = sum(count))
```

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

```
shooting_data_monthly$month_abv <- mymonths[shooting_data_monthly$occur_month]

shooting_data_monthly %>%
  ggplot(aes(x = occur_month, y = count_sum, group = BORO)) +
  geom_line(aes(color = BORO)) +
  geom_point(aes(color = BORO)) +
  theme(legend.title=element_blank()) +
```

```
  scale_x_continuous(
    breaks = seq_along(shooting_data_monthly$month_abv),
    labels = shooting_data_monthly$month_abv
  ) +
  theme(legend.position ="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Sum of Previous 10 years of Documented NYPD Shootings per Month", y = NULL, x = "Month
```



It looks like shootings are more likely to occur in the summer months compared to winter.

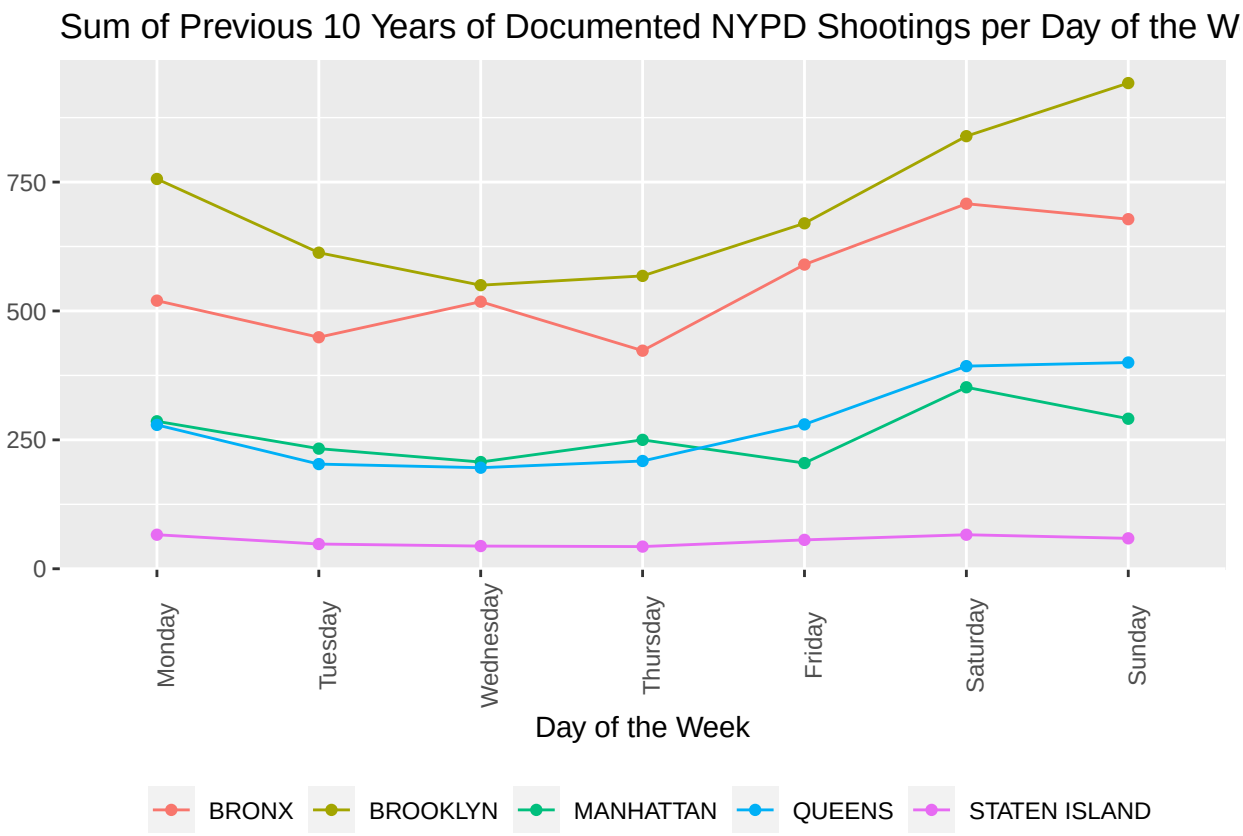
Visualization and Analysis 4

```
shooting_data_weekly <- shooting_data %>%
  filter(occur_date > "2013-9-29") %>%
  mutate(dow = strftime(occur_date, "%A")) %>%
  mutate(dow = factor(dow, levels = c("Monday", "Tuesday","Wednesday", "Thursday", "Friday","Saturday",
  group_by(BORO, dow) %>%
  summarise(count_sum = sum(count))
```

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

```
shooting_data_weekly %>%
  ggplot(aes(x = dow, y = count_sum, group = BORO)) +
  geom_line(aes(color = BORO)) +
  geom_point(aes(color = BORO)) +
```

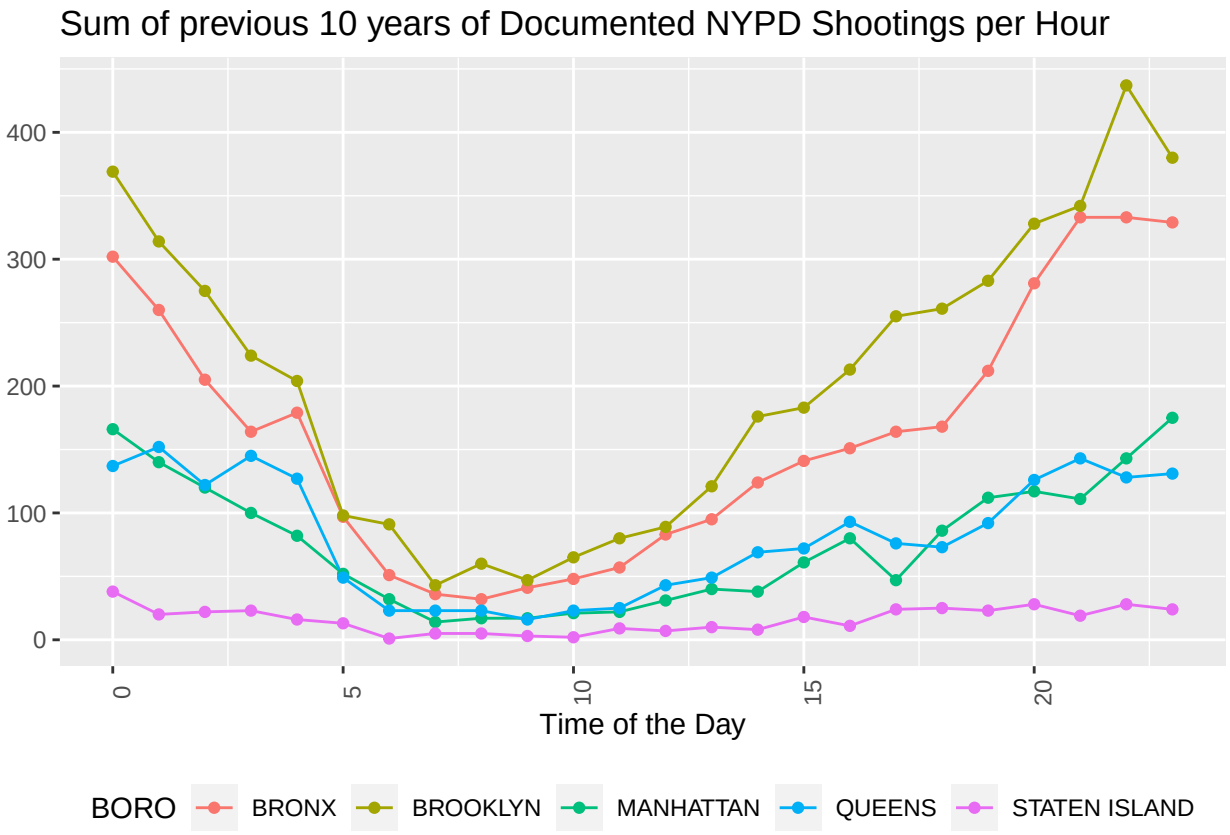
```
  theme(legend.title=element_blank()) +
  theme(legend.position ="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Sum of Previous 10 Years of Documented NYPD Shootings per Day of the Week", y = NULL, x
```



It looks like the weekends have a slightly higher rate of reported shootings compared to the weekdays.

Visualization and Analysis 5

```
shooting_data_hourly %>%
  ggplot(aes(x = occur_hour, y = count_sum, group = BORO)) +
  geom_line(aes(color = BORO)) +
  geom_point(aes(color = BORO)) +
  theme(legend.position ="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Sum of previous 10 years of Documented NYPD Shootings per Hour", y = NULL, x = "Time of
```



It looks like the mid morning to mid afternoon are the least amount of reported shootings.

Conclusion and Bias Identification

After evaluating this gun violence data to find patterns in times and locations to help Police precincts be better managed and equipped to deal with dangerous scenarios while also informing citizens about the safest times to be outside in their wonderful city. I've found that year (P value from the regression table was 0.000157) and months of the year (P value from the regression table was 0.018546) are better predictors for reported shooting incidences; day of the week seems is not as big of a predictor as I previously expected. The borough seems to be a significant predictor as well.

My personal bias is that I'm assuming most of the shootings are males so I checked to validate that bias and it was true. I went ahead and validated that with data to mitigate that bias. There could be bias in data from the collections that many of the precincts do not have data. This bias could be that there isn't available data or they reported no data. I've also wondered how each precinct reports data could include errors. For example maybe Staten Island or some of the precincts in certain boroughs do not have updated technology to appropriately report shooting incidences.