Assignment 3

This assignment is about NYPD Shooting Incidents. This data is the breakdown of incidents that took place back in 2006. Each record has data about the shooting incidents that includes information about the event, location, and the time of occurrence.

Step 1: Import Data

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

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:27312       Length:27312       Length:27312
##  1st Qu.: 63860880   Class :character   Class :character   Class :character
##  Median : 90372218   Mode  :character   Mode  :character   Mode  :character
##  Mean   :120860536
##  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       Length:27312            Length:27312
##  Class :character   Class :character        Class :character
##  Mode  :character   Mode  :character        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
```

```
csv_data$OCCUR_DATE <- lubridate::mdy(csv_data$OCCUR_DATE, tz = "EST")
csv_data$OCCUR_TIME <- hms::hms(lubridate::hms(csv_data$OCCUR_TIME))
csv_data$STATISTICAL_MURDER_FLAG <- as.logical(csv_data$STATISTICAL_MURDER_FLAG)
summary(csv_data)
```

```
##   INCIDENT_KEY         OCCUR_DATE                      OCCUR_TIME
##  Min.   :  9953245   Min.   :2006-01-01 00:00:00.00   Length:27312
##  1st Qu.: 63860880   1st Qu.:2009-07-18 00:00:00.00   Class1:hms
##  Median : 90372218   Median :2013-04-29 00:00:00.00   Class2:difftime
##  Mean   :120860536   Mean   :2014-01-06 23:14:14.13   Mode  :numeric
##  3rd Qu.:188810230   3rd Qu.:2018-10-15 00:00:00.00
##  Max.   :261190187   Max.   :2022-12-31 00:00:00.00
##
##      BORO           LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE
##  Length:27312       Length:27312       Min.   :  1.00   Min.   :0.0000
##  Class :character   Class :character   1st Qu.: 44.00   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median : 68.00   Median :0.0000
##                                        Mean   : 65.64   Mean   :0.3269
##                                        3rd Qu.: 81.00   3rd Qu.:0.0000
##                                        Max.   :123.00   Max.   :2.0000
##                                                         NA's   :2
##  LOC_CLASSFCTN_DESC LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Length:27312       Length:27312       Mode :logical
##  Class :character   Class :character   FALSE:22046
##  Mode  :character   Mode  :character   TRUE :5266
##
##
##
##
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  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_SEX            VIC_RACE           X_COORD_CD        Y_COORD_CD
##  Length:27312       Length:27312       Min.   : 914928   Min.   :125757
##  Class :character   Class :character   1st Qu.:1000028   1st Qu.:182834
##  Mode  :character   Mode  :character   Median :1007731   Median :194487
##                                        Mean   :1009449   Mean   :208127
```

```
##                                      3rd Qu.:1016838   3rd Qu.:239518
##                                      Max.   :1066815   Max.   :271128
##
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:27312
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Median :40.70   Median :-73.92   Mode  :character
##  Mean   :40.74   Mean   :-73.91
##  3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :40.91   Max.   :-73.70
##  NA's   :10      NA's   :10
```

Step 2: Data cleaning and transformation

Here, I would like to see incidents over the years and determine if the incidents increased or decreased the fatality rate. To run the data in a way where I can determine the death, I will need to nullify or remove some records to gather the correct data. Let's go ahead and eliminate PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD, Lon_Lat.

```
csv_data <- dplyr::select(csv_data, -c(INCIDENT_KEY, OCCUR_TIME, JURISDICTION_CODE,
                                LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_RACE,
                                VIC_AGE_GROUP, VIC_SEX, VIC_RACE, X_COORD_CD, Y_COORD_CD,
                                Latitude, Longitude, Lon_Lat))
summary(csv_data)
```

```
##    OCCUR_DATE                         BORO           LOC_OF_OCCUR_DESC
##  Min.   :2006-01-01 00:00:00.00   Length:27312       Length:27312
##  1st Qu.:2009-07-18 00:00:00.00   Class :character   Class :character
##  Median :2013-04-29 00:00:00.00   Mode  :character   Mode  :character
##  Mean   :2014-01-06 23:14:14.13
##  3rd Qu.:2018-10-15 00:00:00.00
##  Max.   :2022-12-31 00:00:00.00
##     PRECINCT      LOC_CLASSFCTN_DESC STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Length:27312       Mode :logical
##  1st Qu.: 44.00   Class :character   FALSE:22046
##  Median : 68.00   Mode  :character   TRUE :5266
##  Mean   : 65.64
##  3rd Qu.: 81.00
##  Max.   :123.00
```

As my next step, I would filter the see each month and year to analyze in detail. To achieve that, I will be creating various data frames.

- How many incidents occurred in a year?
- How many incidents occurred in a month?

```
csv_data <- add_column(csv_data, tibble(MONTH = lubridate::month(csv_data$OCCUR_DATE),
                                        YEAR = lubridate::year(csv_data$OCCUR_DATE))) %>%
dplyr:: select(-c(OCCUR_DATE))
p_monthly_incidents <- tibble(BORO = csv_data$BORO, PRECINCT = csv_data$PRECINCT,
                              DATE = lubridate::make_date(csv_data$YEAR, csv_data$MONTH))
p_monthly_incidents <- p_monthly_incidents %>% count(BORO, PRECINCT, DATE) %>%
                       rename(INCIDENTS = n)
p_yearly_incidents <- tibble(BORO = csv_data$BORO, PRECINCT = csv_data$PRECINCT,
                             YEAR = csv_data$YEAR)
p_yearly_incidents <- p_yearly_incidents %>% count(BORO, PRECINCT, YEAR) %>%
```

```
                        rename(INCIDENTS = n)
b_monthly_incidents <- tibble(BORO = csv_data$BORO,
                              DATE = lubridate::make_date(csv_data$YEAR, csv_data$MONTH))
b_monthly_incidents <- b_monthly_incidents %>% count(BORO, DATE) %>% rename(INCIDENTS = n)
b_yearly_incidents <- tibble(BORO = csv_data$BORO, YEAR = csv_data$YEAR)
b_yearly_incidents <- b_yearly_incidents %>% count(BORO, YEAR) %>% rename(INCIDENTS = n)
```

Months have lower count than years.

```
print(p_monthly_incidents, n = 5)
```

```
## # A tibble: 8,334 x 4
##   BORO  PRECINCT DATE       INCIDENTS
##   <chr>    <int> <date>         <int>
## 1 BRONX       40 2006-01-01         3
## 2 BRONX       40 2006-02-01         3
## 3 BRONX       40 2006-03-01         6
## 4 BRONX       40 2006-04-01         4
## 5 BRONX       40 2006-05-01         2
## # i 8,329 more rows
```

```
print(p_yearly_incidents, n = 5)
```

```
## # A tibble: 1,202 x 4
##   BORO  PRECINCT  YEAR INCIDENTS
##   <chr>    <int> <dbl>     <int>
## 1 BRONX       40  2006        49
## 2 BRONX       40  2007        56
## 3 BRONX       40  2008        54
## 4 BRONX       40  2009        71
## 5 BRONX       40  2010        67
## # i 1,197 more rows
```

```
print(b_monthly_incidents, n = 5)
```

```
## # A tibble: 1,012 x 3
##   BORO  DATE       INCIDENTS
##   <chr> <date>         <int>
## 1 BRONX 2006-01-01        40
## 2 BRONX 2006-02-01        22
## 3 BRONX 2006-03-01        27
## 4 BRONX 2006-04-01        46
## 5 BRONX 2006-05-01        46
## # i 1,007 more rows
```

```
print(b_yearly_incidents, n = 5)
```

```
## # A tibble: 85 x 3
##   BORO   YEAR INCIDENTS
##   <chr> <dbl>     <int>
## 1 BRONX  2006       568
## 2 BRONX  2007       533
## 3 BRONX  2008       520
## 4 BRONX  2009       529
## 5 BRONX  2010       525
## # i 80 more rows
```

This method helps us to view the data easily.

Step 3: Add Data Visualizations

We will start creating visualizations for precincts using different statistical concepts such as mean, median, variance, and standard deviation.

```
stats_p_monthly <- aggregate(INCIDENTS ~ PRECINCT, p_monthly_incidents,
                             function(x) c(M = mean(x), SD = sd(x), VAR = var(x)))
summary(stats_p_monthly)
```

```
##     PRECINCT          INCIDENTS.M         INCIDENTS.SD        INCIDENTS.VAR
##  Min.   :  1.00   Min.   :1.000000   Min.   :0.447214   Min.   : 0.200000
##  1st Qu.: 32.00   1st Qu.:1.629032   1st Qu.:1.044277   1st Qu.: 1.090935
##  Median : 66.00   Median :2.018018   Median :1.522301   Median : 2.317490
##  Mean   : 63.32   Mean   :2.597242   Mean   :1.841815   Mean   : 4.465072
##  3rd Qu.:100.00   3rd Qu.:3.195531   3rd Qu.:2.406016   3rd Qu.: 5.789859
##  Max.   :123.00   Max.   :7.632353   Max.   :5.583855   Max.   :31.179441
##                   NA                 NA's   :1          NA's   :1
```

```
mean(p_monthly_incidents$INCIDENTS)
```

```
## [1] 3.277178
```

```
sd(p_monthly_incidents$INCIDENTS)
```

```
## [1] 2.981839
```

```
var(p_monthly_incidents$INCIDENTS)
```

```
## [1] 8.891362
```

```
stats_p_yearly <- aggregate(INCIDENTS ~ PRECINCT, p_yearly_incidents,
                            function(x) c(M = mean(x), SD = sd(x), VAR = var(x)))
summary(stats_p_yearly)
```

```
##     PRECINCT          INCIDENTS.M         INCIDENTS.SD        INCIDENTS.VAR
##  Min.   :  1.00   Min.   : 1.00000   Min.   : 0.752773   Min.   :  0.5667
##  1st Qu.: 32.00   1st Qu.: 5.05882   1st Qu.: 3.396687   1st Qu.: 11.5727
##  Median : 66.00   Median :12.35294   Median : 5.733492   Median : 32.8897
##  Mean   : 63.32   Mean   :21.12328   Mean   : 8.168798   Mean   :106.2455
##  3rd Qu.:100.00   3rd Qu.:29.41176   3rd Qu.:11.813479   3rd Qu.:139.5588
##  Max.   :123.00   Max.   :91.58824   Max.   :29.742770   Max.   :884.6324
##                   NA                 NA's   :1           NA's   :1
```

```
mean(p_yearly_incidents$INCIDENTS)
```

```
## [1] 22.72213
```

```
sd(p_yearly_incidents$INCIDENTS)
```

```
## [1] 23.3518
```

```
var(p_yearly_incidents$INCIDENTS)
```

```
## [1] 545.3066
```

All the incidents vary according to the precinct. The data reflects the monthly and the magnitude of numbers is larger, it's easier to see the variation. We might get a similar result from Boro.

```r
stats_b_monthly <- aggregate(INCIDENTS ~ BORO, b_monthly_incidents,
                             function(x) c(M = mean(x), SD = sd(x), VAR = var(x)))
summary(stats_b_monthly)
```

```
##      BORO
##  Length:5
##  Class :character
##  Mode  :character
##
##
##
##      INCIDENTS.M         INCIDENTS.SD         INCIDENTS.VAR
##  Min.   : 3.95918     Min.   : 2.575699    Min.   :  6.6342
##  1st Qu.:17.50980     1st Qu.: 8.558975    1st Qu.: 73.2561
##  Median :20.06863     Median : 9.123412    Median : 83.2366
##  Mean   :26.80752     Mean   :12.423263    Mean   :214.3204
##  3rd Qu.:38.90686     3rd Qu.:16.902973    3rd Qu.:285.7105
##  Max.   :53.59314     Max.   :24.955254    Max.   :622.7647
```

```r
mean(b_monthly_incidents$INCIDENTS)
```

```
## [1] 26.98814
```

```r
sd(b_monthly_incidents$INCIDENTS)
```

```
## [1] 22.74222
```

```r
var(b_monthly_incidents$INCIDENTS)
```

```
## [1] 517.2086
```

```r
stats_b_yearly <- aggregate(INCIDENTS ~ BORO, b_yearly_incidents,
                             function(x) c(M = mean(x), SD = sd(x), VAR = var(x)))
summary(stats_b_yearly)
```

```
##      BORO
##  Length:5
##  Class :character
##  Mode  :character
##
##
##
##      INCIDENTS.M         INCIDENTS.SD         INCIDENTS.VAR
##  Min.   : 45.6471     Min.   : 10.71180    Min.   :  114.743
##  1st Qu.:210.1176     1st Qu.: 59.49605    1st Qu.: 3539.779
##  Median :240.8235     Median : 70.13102    Median : 4918.360
##  Mean   :321.3176     Mean   : 86.49129    Mean   :10531.721
##  3rd Qu.:466.8824     3rd Qu.:119.41884    3rd Qu.:14260.860
##  Max.   :643.1176     Max.   :172.69876    Max.   :29824.860
```

```r
mean(b_yearly_incidents$INCIDENTS)
```

```
## [1] 321.3176
```

```r
sd(b_yearly_incidents$INCIDENTS)
```

```
## [1] 233.3874
```

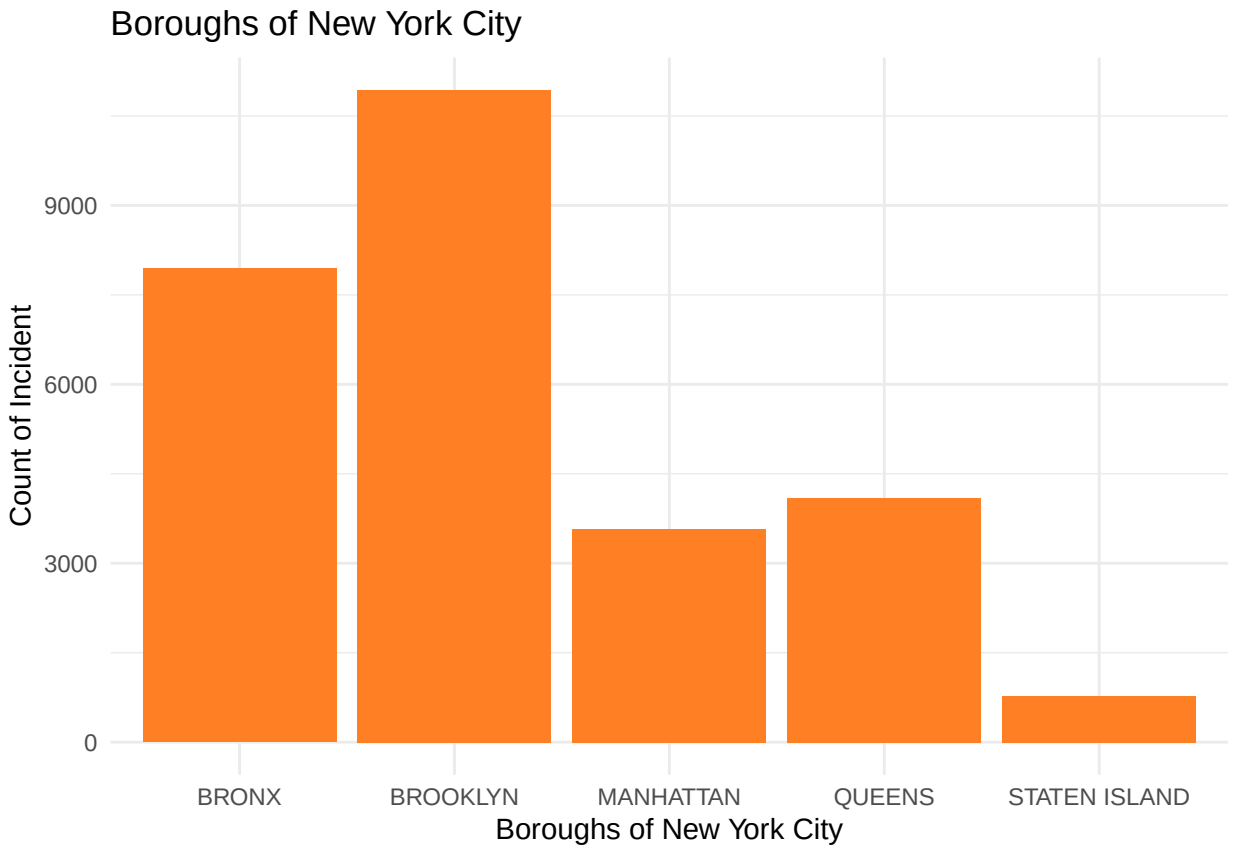
```r
var(b_yearly_incidents$INCIDENTS)
```

```
## [1] 54469.7
```

Visualization 1:

A few questions that we need to analyze through the visualiations are Question: Which part of NY has more number of incidents?**

```r
g <- ggplot(csv_data, aes(x = BORO)) +
  geom_bar(fill="#FF7F24") +
  labs(title = "Boroughs of New York City",
       x = "Boroughs of New York City",
       y = "Count of Incident") +
  theme_minimal()
g
```




```r
table(csv_data$BORO, csv_data$STATISTICAL_MURDER_FLAG)
```

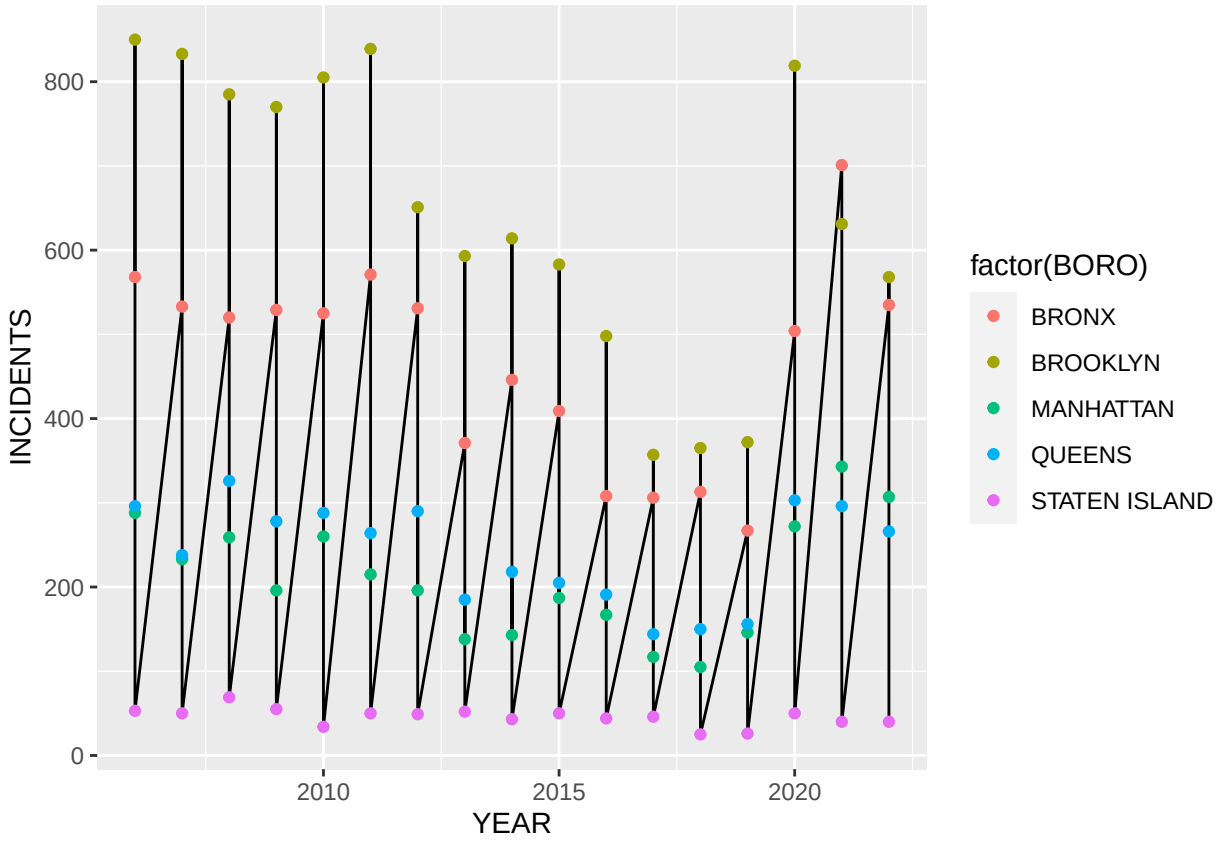
```
##
##                  FALSE TRUE
##    BRONX          6395 1542
##    BROOKLYN       8811 2122
##    MANHATTAN      2942  630
##    QUEENS         3284  810
##    STATEN ISLAND   614  162
```

Looking at the above visualization, it seems that Brooklyn is the highest in terms of maximum incidents

followed by Bronx, Queens, Manhattan, and Staten Island. Staten Island is the region with the lowest number of incidents according to the data.

Visualization 2

```
ggplot(b_yearly_incidents, aes(x=YEAR, y=INCIDENTS)) +
    geom_line() +
    geom_point(aes(color = factor(BORO)))
```




```
    labs(title = "Incidents by Month in New York City") +
  theme_minimal()
```

```
## NULL
```

Here, Brooklyn has more incidents even though population is not counted yet.

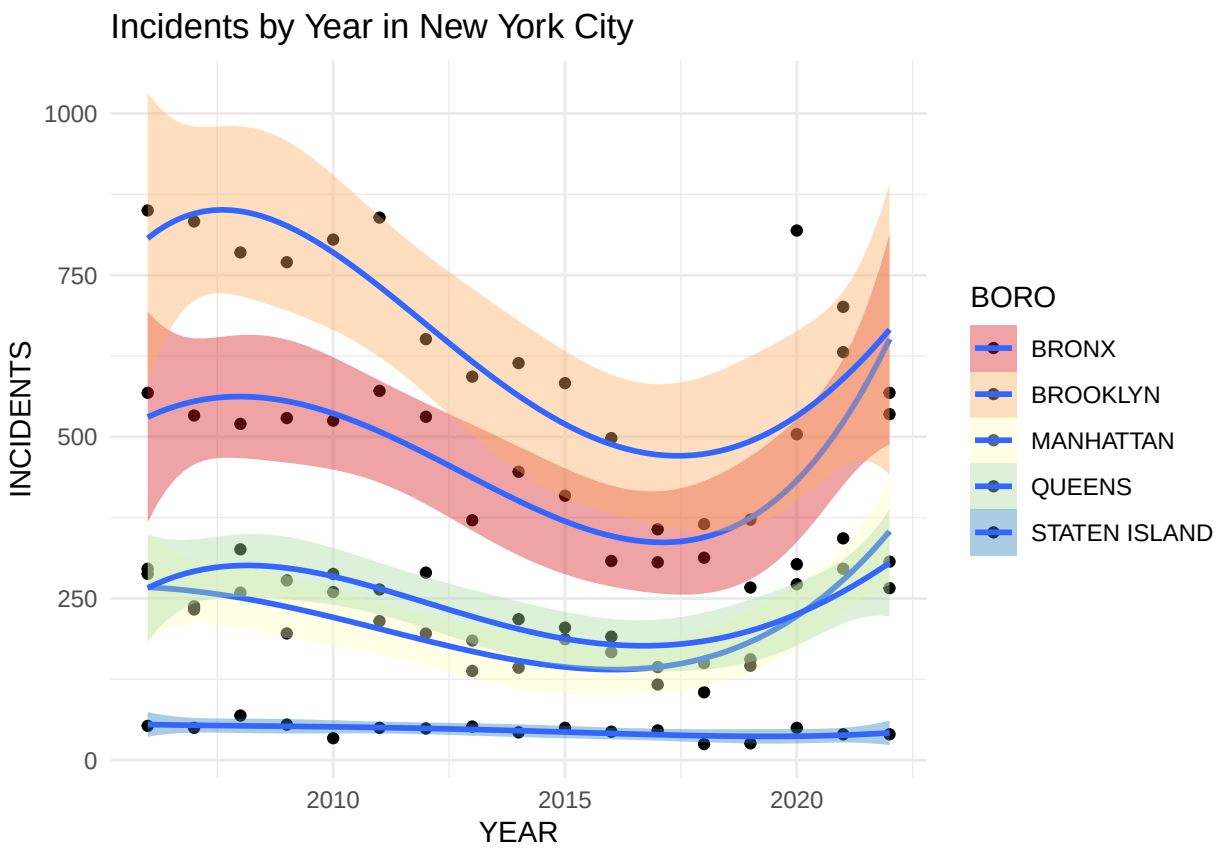
Model

Building linear regression model to predict the incidents by borough in New York by year?

I will use linear regression model to to predict the incidents by borough in New York by year

Brooklyn shows a significantly higher incident rate. Data by year shows fairly fine grained data. For now, let's just compute the regression line for each borough.

```
options(warn=-1)
ggplot(b_yearly_incidents, aes(x=YEAR, y = INCIDENTS, fill = BORO)) + scale_fill_brewer(palette="Spectra
    geom_point() + labs(title = "Incidents by Year in New York City") +
    geom_smooth(method = "lm", formula = y ~ x + poly(x,4)) +
  theme_minimal()
```




```
options(warn=1)
```

Analyze Bias

Already identify bias using the data and visualizations. The above chart depicts that Brooklyn has the maximum number of incidents and tells that it is the most dangerous place to stay. Another bias is the lack of location and type of incident information in the original data set. Most incidents didn't have an attached location and incident details (such as murder, robbery). Also, it can provoke discrimination and create unspoken bias among individuals. It's intriguing to find out that Brooklyn has the most number of incidents, followed by the Bronx and Queens. In addition, there are significantly a huge difference in incidents among victim sex, with more incidents with males than those of females.

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Big Sur 11.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Los_Angeles
```

```
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0   dplyr_1.1.2
##  [5] purrr_1.0.1     readr_2.1.4     tidyr_1.3.0     tibble_3.2.1
##  [9] ggplot2_3.4.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] Matrix_1.5-4.1     gtable_0.3.3       highr_0.10         compiler_4.3.1
##  [5] tidyselect_1.2.0   splines_4.3.1      scales_1.2.1       yaml_2.3.7
##  [9] fastmap_1.1.1      lattice_0.21-8     R6_2.5.1           labeling_0.4.2
## [13] generics_0.1.3     knitr_1.43         munsell_0.5.0      RColorBrewer_1.1-3
## [17] pillar_1.9.0       tzdb_0.4.0         rlang_1.1.1        utf8_1.2.3
## [21] stringi_1.7.12     xfun_0.39          timechange_0.2.0   cli_3.6.1
## [25] mgcv_1.8-42        withr_2.5.0        magrittr_2.0.3     digest_0.6.33
## [29] grid_4.3.1         rstudioapi_0.15.0  hms_1.1.3          nlme_3.1-162
## [33] lifecycle_1.0.3    vctrs_0.6.3        evaluate_0.21      glue_1.6.2
## [37] farver_2.1.1       fansi_1.0.4        colorspace_2.1-0   rmarkdown_2.23
## [41] tools_4.3.1        pkgconfig_2.0.3    htmltools_0.5.5
```