Week 3 Assignment

Analysis of Shooting Incidents

1/16/2022

Importing and Cleaning Data

For this assignment we are looking at historical from NYPD. Specifically every shooting incident since 2006. Jurisdiction code is not 0 in only 3954 rows, 54 of which are 1 and 3900 are 2. This data is likely irrelevant to analysis and can be dropped given the minimal variation. Location description is only set in 10004 shootings. This was describing housing types primarily. Since this is roughly half the records, we'll consider the rest as unknowns. Perpetrator age group isn't known in 8295 rows, sex in 8261, and race in 8261. It's easier to identify sex and race than a specific age range, so the slight variation is not concerning. The perpetrator information would only be known on conviction so the high number of empty values is also unconcerning and should just be treated as unknowns. All other data exists in the 23585 rows. The location is replicated 3 times and could be reduced to a single lat/lon pair.

```
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:23585       Length:23585       Length:23585
##  1st Qu.: 55322804   Class :character   Class :character   Class :character
##  Median : 83435362   Mode  :character   Mode  :character   Mode  :character
##  Mean   :102280741
##  3rd Qu.:150911774
##  Max.   :230611229
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.000     Length:23585       Length:23585
##  1st Qu.: 44.00   1st Qu.:0.000     Class :character   Class :character
##  Median : 69.00   Median :0.000     Mode  :character   Mode  :character
##  Mean   : 66.21   Mean   :0.333
##  3rd Qu.: 81.00   3rd Qu.:0.000
##  Max.   :123.00   Max.   :2.000
##                   NA's   :2
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:23585       Length:23585       Length:23585       Length:23585
##  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:23585       Length:23585       Min.   : 914928   Min.   :125757
##  Class :character   Class :character   1st Qu.: 999925   1st Qu.:182539
##  Mode  :character   Mode  :character   Median :1007654   Median :193470
##                                        Mean   :1009379   Mean   :207300
##                                        3rd Qu.:1016782   3rd Qu.:239163
##                                        Max.   :1066815   Max.   :271128
##
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:23585
##  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
##
```

```
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   Length:23585
##  1st Qu.: 55322804   1st Qu.:2008-12-31 00:00:00   Class1:hms
##  Median : 83435362   Median :2012-02-27 00:00:00   Class2:difftime
##  Mean   :102280741   Mean   :2012-10-05 01:27:47   Mode  :numeric
##  3rd Qu.:150911774   3rd Qu.:2016-03-02 00:00:00
##  Max.   :230611229   Max.   :2020-12-31 00:00:00
##
##      BORO              PRECINCT      JURISDICTION_CODE LOCATION_DESC
##  Length:23585       Min.   :  1.00   Min.   :0.000     Length:23585
##  Class :character   1st Qu.: 44.00   1st Qu.:0.000     Class :character
##  Mode  :character   Median : 69.00   Median :0.000     Mode  :character
##                     Mean   : 66.21   Mean   :0.333
##                     3rd Qu.: 81.00   3rd Qu.:0.000
##                     Max.   :123.00   Max.   :2.000
##                                      NA's   :2
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP       PERP_SEX
##  Mode :logical           Length:23585       Length:23585
##  FALSE:19085             Class :character   Class :character
##  TRUE :4500              Mode  :character   Mode  :character
##
##
##
##
##   PERP_RACE          VIC_AGE_GROUP        VIC_SEX            VIC_RACE
##  Length:23585       Length:23585       Length:23585       Length:23585
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    X_COORD_CD        Y_COORD_CD        Latitude       Longitude
```

```
##  Min.   : 914928   Min.   :125757   Min.   :40.51   Min.   :-74.25
##  1st Qu.: 999925   1st Qu.:182539   1st Qu.:40.67   1st Qu.:-73.94
##  Median :1007654   Median :193470   Median :40.70   Median :-73.92
##  Mean   :1009379   Mean   :207300   Mean   :40.74   Mean   :-73.91
##  3rd Qu.:1016782   3rd Qu.:239163   3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :1066815   Max.   :271128   Max.   :40.91   Max.   :-73.70
##
##    Lon_Lat
##  Length:23585
##  Class :character
##  Mode  :character
##
##
##
##
```

I'd like to see if shootings are clustered in certain areas and year over year trends in whether there was an increase or decrease in deaths. Particularly if the direction trend was fairly stable across regions or are there outliers that change. I'd then compare those outliers with the areas with clustered shootings to see if there's a correlation. I'd start with the high level view of by boro and then look at it with a more finer grain by precinct. Race would potentially be interesting to look at but this data set isn't complete enough to do a breakdown by race since only half the shootings have that data.

This means I don't need the INCIDENT_KEY for my analysis since it's just a unique identifier. I will need the OCCUR_DATE field but right now just the year. Since we're only looking at 18 years of data we may need more data points so month information should be left in as well. I won't need OCCUR_TIME and can drop that. BORO and PRECINCT as controlled variables for analysis I obviously want to keep in. JURISDICTION_CODE, LOCATION_DESC I can drop. I'd like to keep the STATISTICAL_MURDER_FLAG for now since comparing the two may be interesting and it might help with finding clusters later. PERP_* and VIC_* demographics I obviously don't need as well as the X and Y coordinates, Latitude, Longitude and Lon_Lat because that would have me analyzing too fine grained data if I did. Those locations are likely too specific to tell me anything about how regions trended over time.

```
csv_data <- 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              PRECINCT
##  Min.   :2006-01-01 00:00:00   Length:23585       Min.   :  1.00
##  1st Qu.:2008-12-31 00:00:00   Class :character   1st Qu.: 44.00
##  Median :2012-02-27 00:00:00   Mode  :character   Median : 69.00
##  Mean   :2012-10-05 01:27:47                      Mean   : 66.21
##  3rd Qu.:2016-03-02 00:00:00                      3rd Qu.: 81.00
##  Max.   :2020-12-31 00:00:00                      Max.   :123.00
##  STATISTICAL_MURDER_FLAG
##  Mode :logical
##  FALSE:19085
##  TRUE :4500
##
##
##
```

Now instead of having OCCUR_DATE be a column, I'd like to make a row for each month and year. For the next step, instead of adding additional columns to my data set, I'm actually going to create four new data frames:

1. For each Boro/Precinct location, count how many incidents occurred in a month
2. For each Boro/Precinct location, count how many incidents occurred in a year
3. For each Boro, count how many incidents occurred in a month
4. For each Boro, count how many incidents occurred in a year.

```
csv_data <- add_column(csv_data, tibble(MONTH = lubridate::month(csv_data$OCCUR_DATE),
                                        YEAR = lubridate::year(csv_data$OCCUR_DATE))) %>%
            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)
```

Now let's take a quick look at a few rows to make sure the data makes sense (months have lower counts than years, precincts have lower counts than boros).

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

```
## # A tibble: 7,251 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
## # ... with 7,246 more rows
```

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

```
## # A tibble: 1,053 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
## # ... with 1,048 more rows
```

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

```
## # A tibble: 894 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
## # ... with 889 more rows
```

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

```
## # A tibble: 75 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
## # ... with 70 more rows
```

You can see from the sample lines that it's an easier way view the data instead of adding monthly and yearly columns to the existing data frame for each precinct (and then either add repetitive extra columns for the boro totals in each precinct or have to calculate them every time I need them). This also allows me to preserve the STATISTICAL_MURDER_FLAG in case I do want to use it for analysis in the future.

Visualizations

Before getting into charts, I'd like to do some quick summary math that will help with interpreting the graph and set expectations for what I'll see. These will be very quick statistics: mean, median, standard deviation and variance. I want to see them two different ways for each data frame, looking at variance within precinct or boro over time and then across the entire data set.

Let's start by looking at each precinct:

```
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.   :0.440959   Min.   : 0.19444
##  1st Qu.: 32.00   1st Qu.:1.603774   1st Qu.:1.008785   1st Qu.: 1.01769
##  Median : 66.00   Median :1.927083   Median :1.439824   Median : 2.07332
##  Mean   : 63.32   Mean   :2.557066   Mean   :1.810005   Mean   : 4.38933
##  3rd Qu.:100.00   3rd Qu.:3.179487   3rd Qu.:2.335258   3rd Qu.: 5.45473
##  Max.   :123.00   Max.   :7.638889   Max.   :5.761327   Max.   :33.19289
##                   NA                 NA's   :1          NA's   :1
```

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

```
## [1] 3.252655
```

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

```
## [1] 2.980561
```

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

```
## [1] 8.883743
```

```
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.   : 0.57735     Min.   :   0.3333
##  1st Qu.: 32.00   1st Qu.: 4.46154     1st Qu.: 3.08256     1st Qu.:   9.5024
##  Median : 66.00   Median :12.25000     Median : 5.55349     Median :  30.8667
##  Mean   : 63.32   Mean   :20.68902     Mean   : 7.95818     Mean   : 104.5093
##  3rd Qu.:100.00   3rd Qu.:29.26667     3rd Qu.:11.86388     3rd Qu.: 140.7524
##  Max.   :123.00   Max.   :91.66667     Max.   :31.75951     Max.   :1008.6667
##                   NA                   NA's   :1            NA's   :1
```

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

```
## [1] 22.39791
```

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

```
## [1] 23.29188
```

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

```
## [1] 542.5117
```

Looking at the distribution tells us the number of incidents vary significantly by precinct. The yearly data reflects the monthly data as one would expect. Because the magnitude of numbers is larger, it's easier to see the variation. The mean and median are not that close to each other with the median being lower. This suggests a clumping of precincts with a low number of incidents but several precincts pulling up the mean. The minimal mean being 1 and the maximum rounding to 92 shows a wide variation. The standard deviation and variance also run from very small to very large, also indicating that there are areas of significantly higher incidents. We should see similar results if we look at it by boro, but that may smooth out some of the variation if there's only a few outliers in a particular 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.   : 4.00000     Min.   : 2.644659    Min.   :  6.9942
##  1st Qu.:16.23333     1st Qu.: 7.876916    1st Qu.: 62.0458
##  Median :19.62222     Median : 8.880432    Median : 78.8621
##  Mean   :26.23222     Mean   :12.283971    Mean   :215.4492
##  3rd Qu.:37.22778     3rd Qu.:16.182213    3rd Qu.:261.8640
##  Max.   :54.07778     Max.   :25.835633    Max.   :667.4800
```

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

```
## [1] 26.38143
```

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

```
## [1] 22.84273
```

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

```
## [1] 521.7905
```

```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.   : 46.4000     Min.   : 11.22370    Min.   :  125.97
##  1st Qu.:194.8000     1st Qu.: 58.63349    1st Qu.: 3437.89
##  Median :235.4667     Median : 61.25342    Median : 3751.98
##  Mean   :314.4667     Mean   : 84.45834    Mean   :10513.65
##  3rd Qu.:446.7333     3rd Qu.:107.78051    3rd Qu.:11616.64
##  Max.   :648.9333     Max.   :183.40060    Max.   :33635.78
```

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

```
## [1] 314.4667
```

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

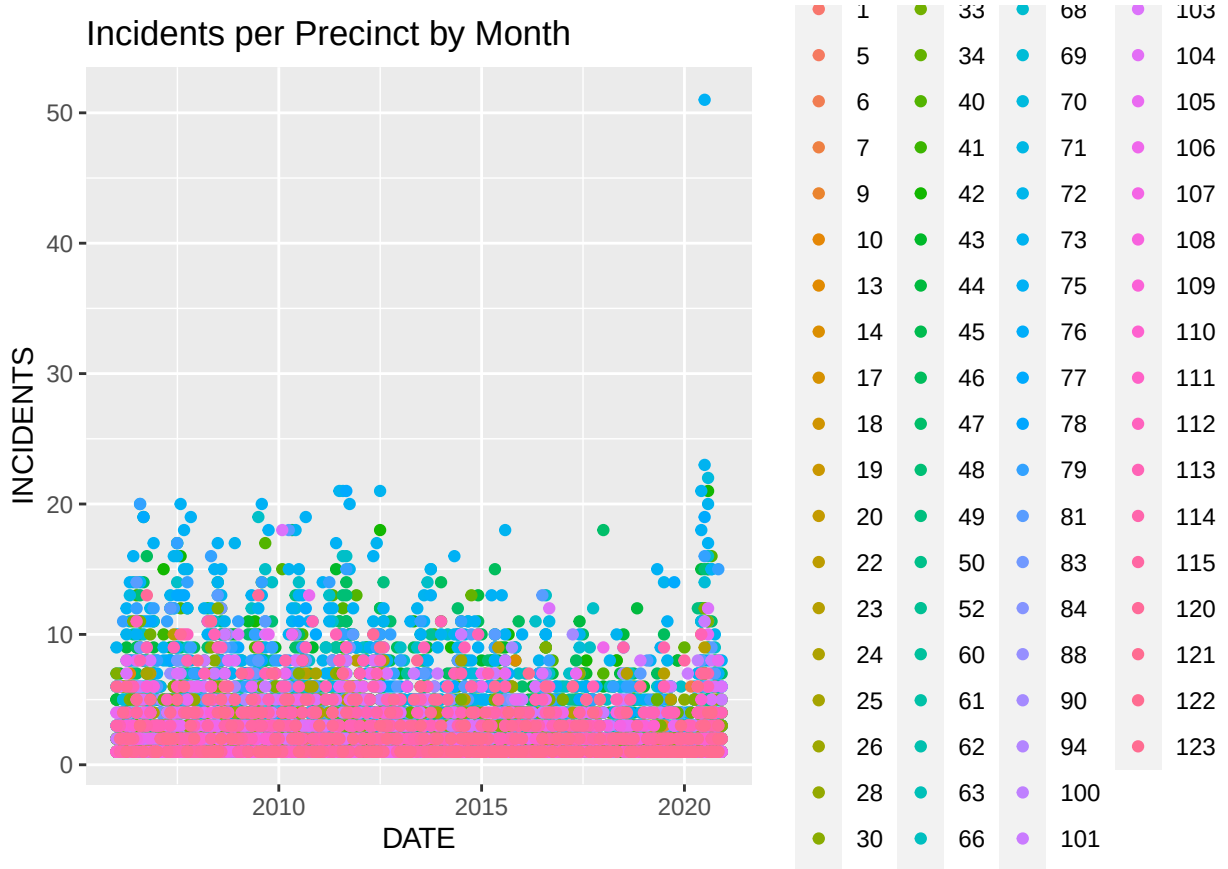
```
## [1] 234.3105
```

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

```
## [1] 54901.41
```

The monthly and yearly stats are of higher magnitude as expected since we're dealing with a larger geographical region. The data does appear to have less variance when comparing boros by both month and year, but there is still a skew toward an outlier with a higher average. Graphs are an excellent way to visually pull out this type of data, so let's go over our data sets.

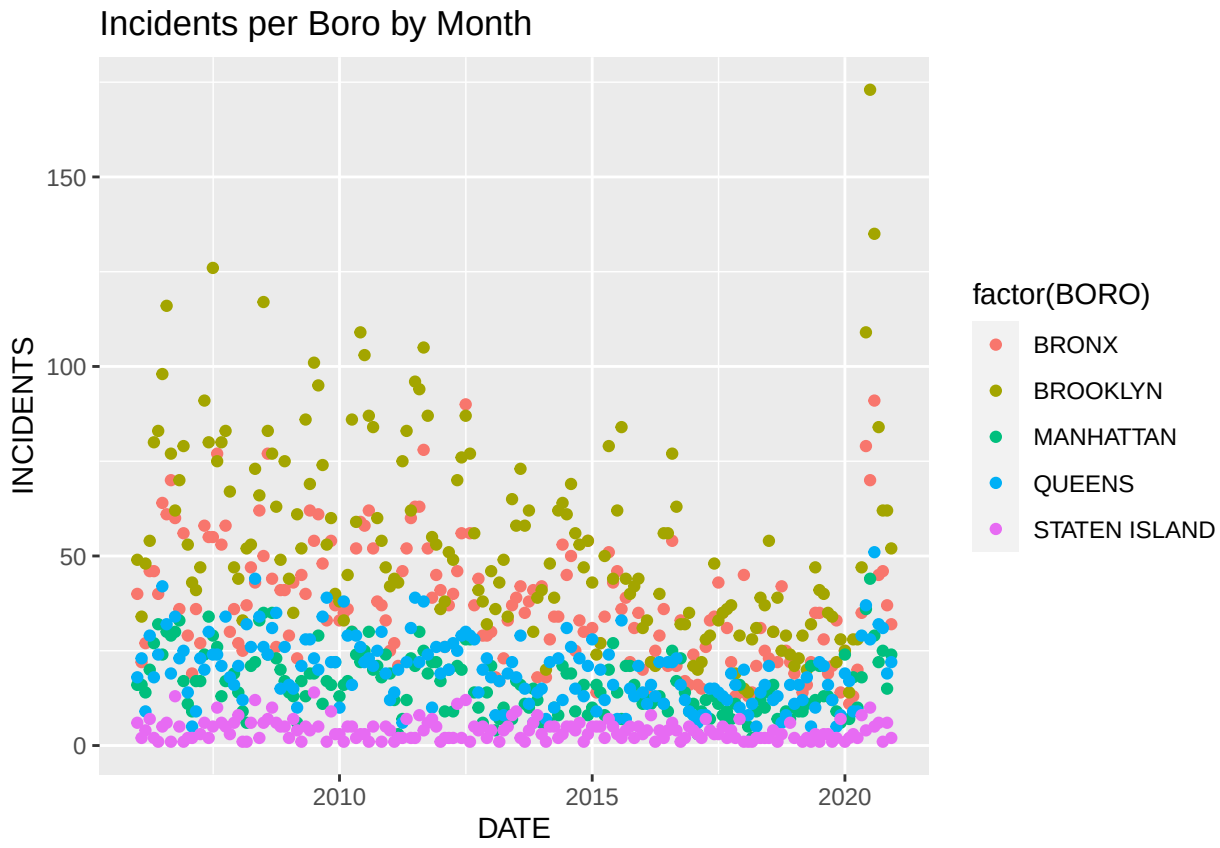
```
ggplot(p_monthly_incidents, aes(x=DATE, y=INCIDENTS)) +
    geom_point(aes(color = factor(PRECINCT))) +
    labs(title = "Incidents per Precinct by Month")
```



The immediate impression is looking at incidents per precinct per month is not a good visualization. The data is way too dense to fit on a printed page in a way that the data points are visually distinct, which also

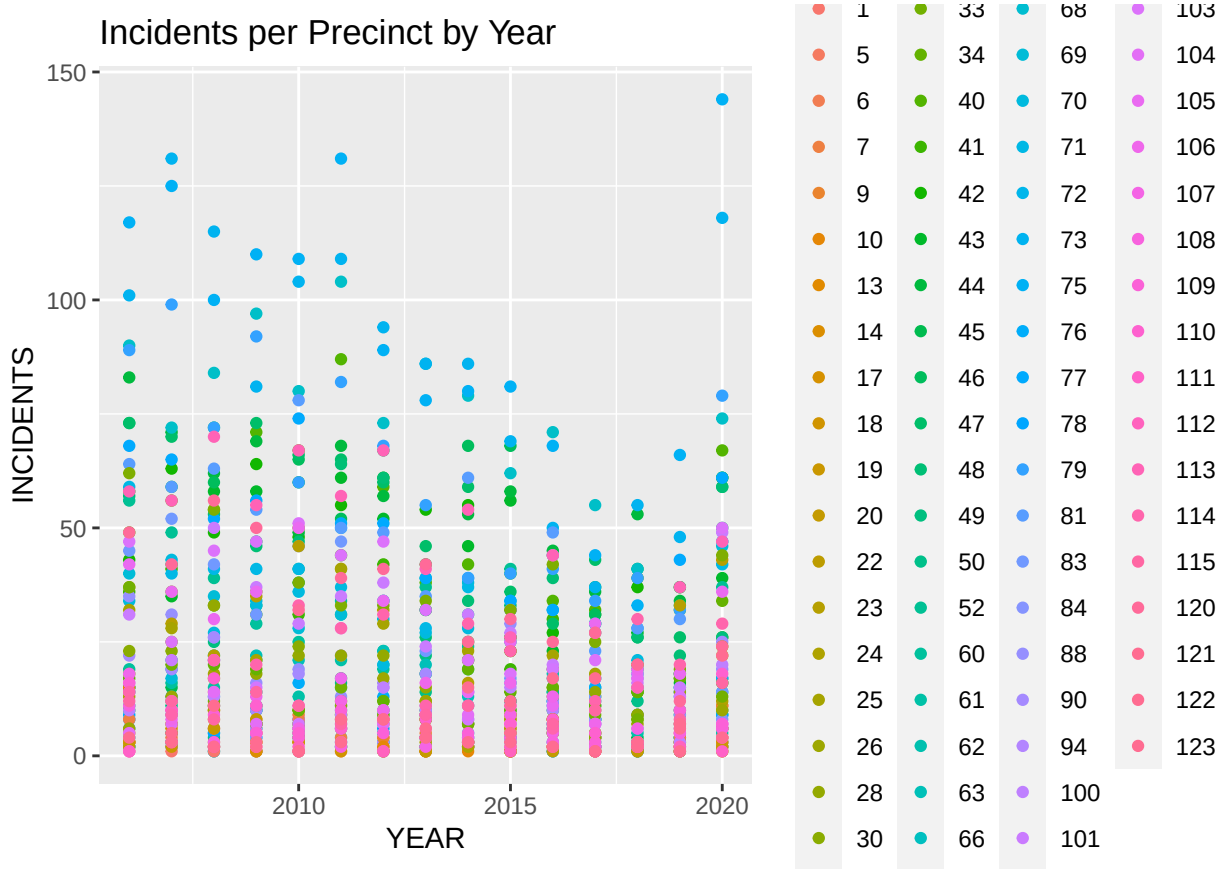
means it won't present well. So while this data is useful potentially for analysis, it is not useful for telling the viewer a story because they're just looking at a giant blob of color with 123 data points at each date mark.

```
ggplot(b_monthly_incidents, aes(x=DATE, y=INCIDENTS)) +
    geom_point(aes(color = factor(BORO))) +
    labs(title = "Incidents per Boro by Month")
```



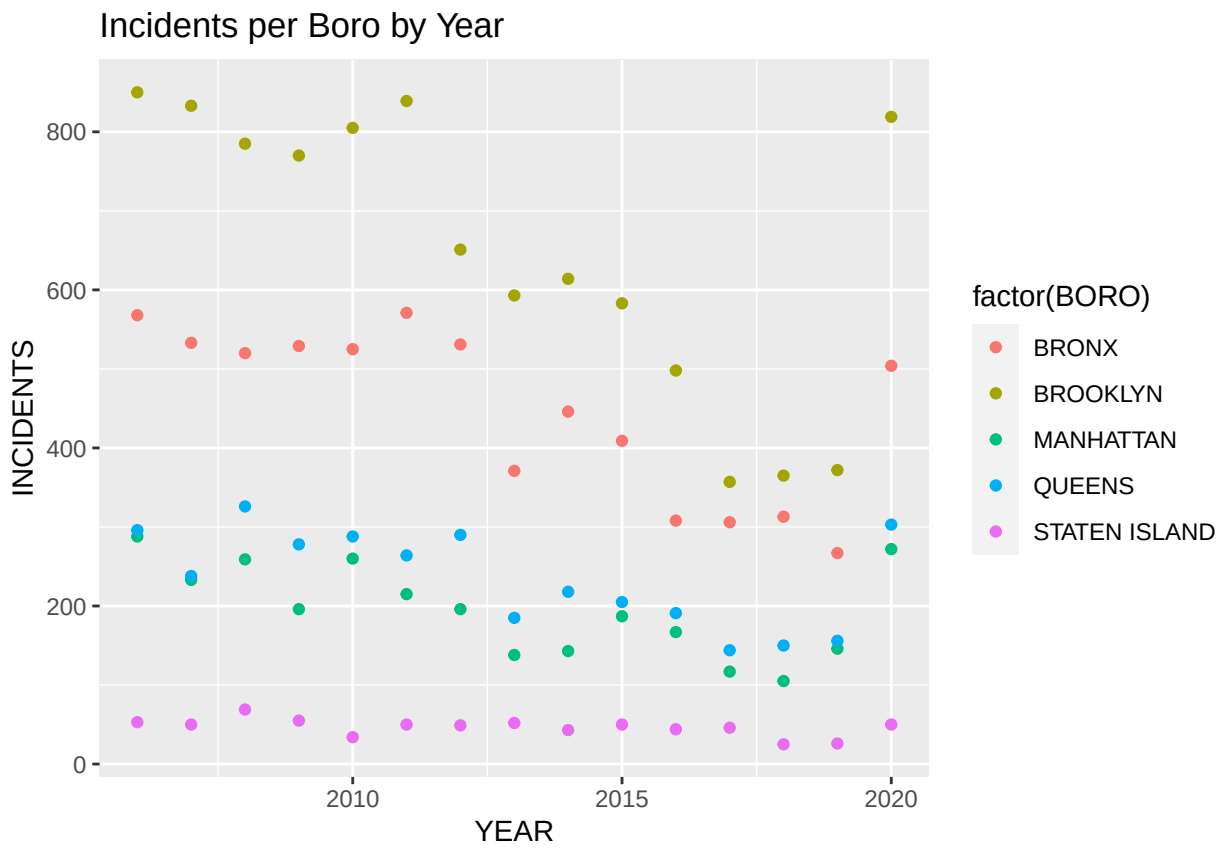
Bringing it up a level to look at incidents per boro per month is slightly better. You can see that Brooklyn has more incidents for example, but we haven't taken populations into account yet. The data points in close linear vicinity also significantly overlap so while you can see a general trend over time, you can't see anything useful within a specific year for a trend. Again, I'd mark this data as useful for analysis, but not for visualization. We still need data that's less dense for someone listening to a presentation to look at and get the point.

```
ggplot(p_yearly_incidents, aes(x=YEAR, y = INCIDENTS)) +
    geom_point(aes(color = factor(PRECINCT))) +
    labs(title = "Incidents per Precinct by Year")
```



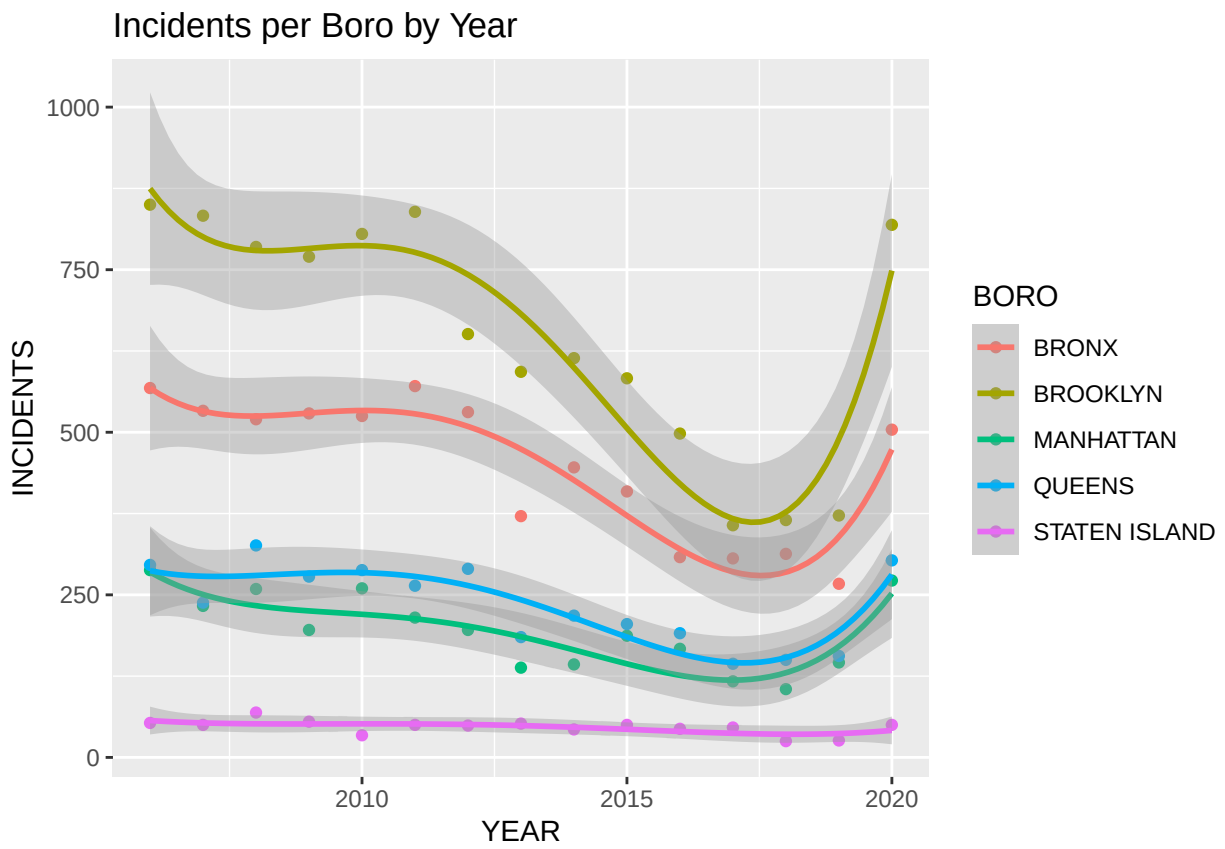
Let's take a look at per year since that will have significantly fewer x-axis points. Per precinct strikes me as a brightly colored bar graph with a few outlier points. Again, the number of variables (123 precincts) is providing too much visual clutter to see useful information. When we look at boro, now we see something useful. For each data point, you can see a distinct mark. The boros are visually separable as are the points in time. This is the kind of graph we would want to use in a presentation to show raw data.

```
ggplot(b_yearly_incidents, aes(x=YEAR, y = INCIDENTS)) +
    geom_point(aes(color = factor(BORO))) +
    labs(title = "Incidents per Boro by Year")
```



There's a few different ways we can add visual information to this graph to add information for the viewer. We could simply draw linear lines between each point so the trend is more visible. A better step could be to find the best fit equation for each boro and show that. On top of that you could add the regression for all of the boros added together and look at how each boro varies from the average regression. The primary reason it doesn't make sense to look at variance against the average right now is we haven't taken population in to account. Brooklyn visually has a significantly higher incident rate, but how does that compare per capita? We would have to go back and find historical population data to make that type of analysis useful. Finding it by year would definitely be possible, by month might be difficult because we're looking at fairly fine grained data. For now, let's just compute the regression line for each boro and add it to the graph to get a better visual comparison of variance in incidents over time (the shaded gray area being the error range for each fit line).

```
options(warn=-1)
ggplot(b_yearly_incidents, aes(x=YEAR, y = INCIDENTS, color = BORO)) +
    geom_point() + labs(title = "Incidents per Boro by Year") +
    geom_smooth(method = "lm", formula = y ~ x + poly(x,4))
```




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

For our original question, are shootings clustered in a particular area, at this point we can identify Brooklyn as having the most. Like the other boros, it trends up and down over time. If we wanted to narrow it down further, we could filter the data down to just the precincts in Brooklyn to see if any are outliers from the others, but I wouldn't want to do that yet, given some of the points I'll discuss on data bias.

Bias

We've already identified one type of bias in this graph: the population size of each boro isn't taken into account. While this graph visually says Brooklyn has the most incidents, which you may interpret as the most dangerous place to live, if it had four times as many people as Queens or Manhattan, it would be firmly in the middle if we graphed incidents per capita.

Another source of bias is the lack of location information in the original data set. Most incidents didn't have an attached location and each boro has a different structure. You would expect a more residential boro with local service oriented businesses (grocery stores, coffee shops, etc.) to have a different profile than a boro that has less residents and more corporate buildings. For example, the metropolitan area I live in, the core metro region has no house and some apartment living in the periphery. You have to get out into the surrounding cities to find residential areas. One would surmise that shooting incidents are more likely to occur in residential type areas than in a building full of actuaries.

Additionally, our data set is missing a large amount of demographic information which can drastically influence outcomes. For example, my husband is an International Association of Arson Investigators Certified Fire Investigator, which is the highest certification one can hold for investigating fires. A number of known factors are directly related to your ability to survive a house fire. Having functioning smoke alarms is by far

the highest factor. Many people can't afford smoke alarms, which is why most fire departments do smoke alarm install campaigns in low income housing. Having a house that isn't clean raises your risk of dying because there are more things to ignite and may make it more difficult for fire crews to find you. Ease of escape is typically next, influenced by the prior factor of how clean is your how (how easy is it to get to your fire escape point) but also do you have a safe fire escape point. Poverty is a high risk factor because you are likely to use unconventional heating methods or live in non-standard housing like manufactured homes. The same goes for homelessness, where non-standard house like tents and unconventional heating and cooking methods are used. Seeing how this type of demographic information is directly correlated to whether or not you will survive a fire, one can see how the similar information would be correlated to being involved in a shooting incident.

Without population data, demographics, build occupancy types we can't create a norming factor to directly compare boros. It's not a deliberate insertion of bias but it does add bias since we can see a clear ordering of highest to least shooting incidents across the boros with the data that we have. But accounting for this other data could drastically change the outcome.

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/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 datasets  utils     methods   base
##
## other attached packages:
## [1] forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7     purrr_0.3.4
## [5] readr_2.1.1     tidyr_1.1.4     tibble_3.1.6    ggplot2_3.3.5
## [9] tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8       lubridate_1.8.0  lattice_0.20-45  assertthat_0.2.1
##  [5] digest_0.6.29    utf8_1.2.2       R6_2.5.1         cellranger_1.1.0
##  [9] backports_1.4.1  reprex_2.0.1     evaluate_0.14    httr_1.4.2
## [13] highr_0.9        pillar_1.6.4     rlang_1.0.1      readxl_1.3.1
## [17] rstudioapi_0.13  Matrix_1.3-4     rmarkdown_2.11   labeling_0.4.2
## [21] splines_4.1.2    munsell_0.5.0    broom_0.7.11     compiler_4.1.2
## [25] modelr_0.1.8     xfun_0.29        pkgconfig_2.0.3  mgcv_1.8-38
## [29] htmltools_0.5.2  tidyselect_1.1.1 fansi_1.0.2      crayon_1.4.2
## [33] tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.3      grid_4.1.2
## [37] nlme_3.1-153     jsonlite_1.7.2   gtable_0.3.0     lifecycle_1.0.1
## [41] DBI_1.1.2        magrittr_2.0.1   scales_1.1.1     cli_3.1.0
## [45] stringi_1.7.6    farver_2.1.0     renv_0.15.1      fs_1.5.2
## [49] xml2_1.3.3       ellipsis_0.3.2   generics_0.1.1   vctrs_0.3.8
## [53] tools_4.1.2      glue_1.6.0       hms_1.1.1        fastmap_1.1.0
## [57] yaml_2.2.1       colorspace_2.0-2 rvest_1.0.2      knitr_1.37
## [61] haven_2.4.3
```