NYPD Data

Raymond Bell

2/12/2022

```
knitr::opts_chunk$set(echo = TRUE)

library(lubridate)
```

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

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

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(splines)
```

Analyzing NYPD data on shooting incidents from 2006 until 2020

Specifically analyzing the per year incident rates for each individual Borough. Including the unemployment rate for the time period to find if there is a high correlation between shooting incidents and unemployment. Also, looking at the correlation between warmer/hotter months of the year and increased incident counts. Generating a model that fits the per month data that could be used to predict future trends.

Loading shooting incident data from https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD. Tidying up the data so the date and time columns are actually date and time types as opposed to strings. Removing Lon_Lat that duplicates data in other columns.

Loading unemployment data from the Bureau of Labor Statistics https://www.bls.gov/web/metro/ssamatab2.txt. Needed to tidy unemployment data as it did not start in csv format. Needed year and unemployment rate data from the dataset.

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

url_unemployment <- "https://www.bls.gov/web/metro/ssamatab2.txt"

nypd_data <- read.csv(url_in)

ny_unemployment <- read.table(url_unemployment,header = F, skip = 5, sep="\t")


nypd_data <- nypd_data %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
  mutate(OCCUR_TIME = hms(OCCUR_TIME)) %>%
  select(-c(Lon_Lat))

summary(nypd_data)
```

```
##   INCIDENT_KEY         OCCUR_DATE           OCCUR_TIME
##  Min.   :  9953245   Min.   :2006-01-01   Min.   :0S
##  1st Qu.: 55322804   1st Qu.:2008-12-31   1st Qu.:3H 20M 0S
##  Median : 83435362   Median :2012-02-27   Median :15H 0M 0S
##  Mean   :102280741   Mean   :2012-10-05   Mean   :12H 33M 7.48187407250225S
##  3rd Qu.:150911774   3rd Qu.:2016-03-02   3rd Qu.:20H 45M 0S
##  Max.   :230611229   Max.   :2020-12-31   Max.   :23H 59M 0S
##
##      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
##  Length:23585            Length:23585       Length:23585
##  Class :character        Class :character   Class :character
##  Mode  :character        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
##
```

Using a pivot_wider to group by year and place the incident count for each borough in a separate column. Displaying and then plotting this data.

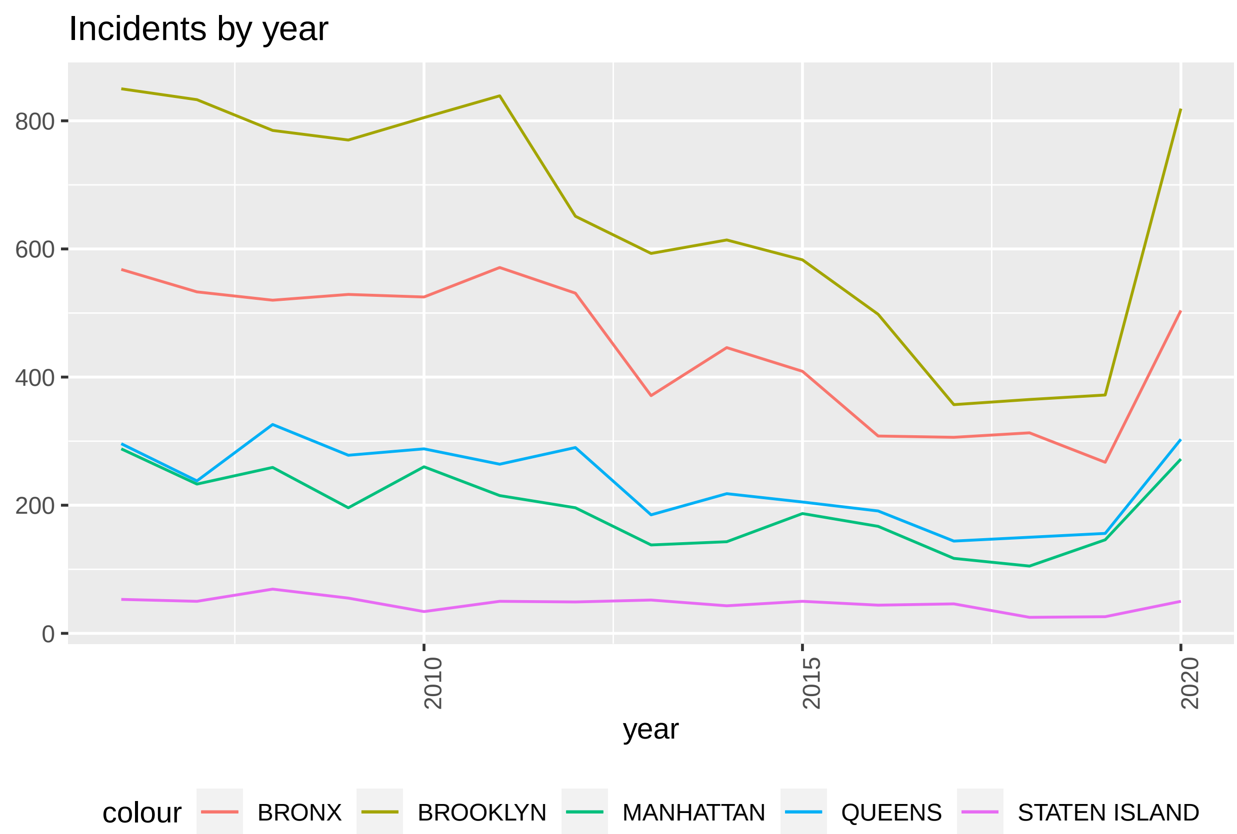
```r
by_date <- nypd_data %>%
  mutate(year = year(OCCUR_DATE)) %>%
  select(c(year, BORO)) %>%
  pivot_wider(id_cols = year,
              names_from = BORO,
              values_from = BORO,
              values_fn = list(BORO = length)) %>%
  arrange(year)

by_date
```

```
## # A tibble: 15 x 6
##     year BRONX QUEENS BROOKLYN MANHATTAN `STATEN ISLAND`
##    <dbl> <int>  <int>    <int>     <int>           <int>
##  1  2006   568    296      850       288              53
##  2  2007   533    238      833       233              50
##  3  2008   520    326      785       259              69
##  4  2009   529    278      770       196              55
##  5  2010   525    288      805       260              34
##  6  2011   571    264      839       215              50
##  7  2012   531    290      651       196              49
##  8  2013   371    185      593       138              52
##  9  2014   446    218      614       143              43
## 10  2015   409    205      583       187              50
## 11  2016   308    191      498       167              44
```

```
## 12  2017   306    144     357      117             46
## 13  2018   313    150     365      105             25
## 14  2019   267    156     372      146             26
## 15  2020   504    303     819      272             50
```

```
by_date %>%
  ggplot(aes(x= year, y= BRONX)) +
  geom_line(aes(color="BRONX")) +
  geom_line(aes(y= QUEENS, color="QUEENS")) +
  geom_line(aes(y= BROOKLYN, color="BROOKLYN")) +
  geom_line(aes(y= MANHATTAN, color="MANHATTAN")) +
  geom_line(aes(y= `STATEN ISLAND`, color="STATEN ISLAND")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents by year", y = NULL)
```



As mentioned in the addressing bias section below, this original data above did not take population size into account so the incident count for Brooklyn was higher than it would be if normalized for population. The exact opposite was true for Staten Island where the original graph might lead one to conclude that Staten Island was simply significantly safer than the other Boroughs.

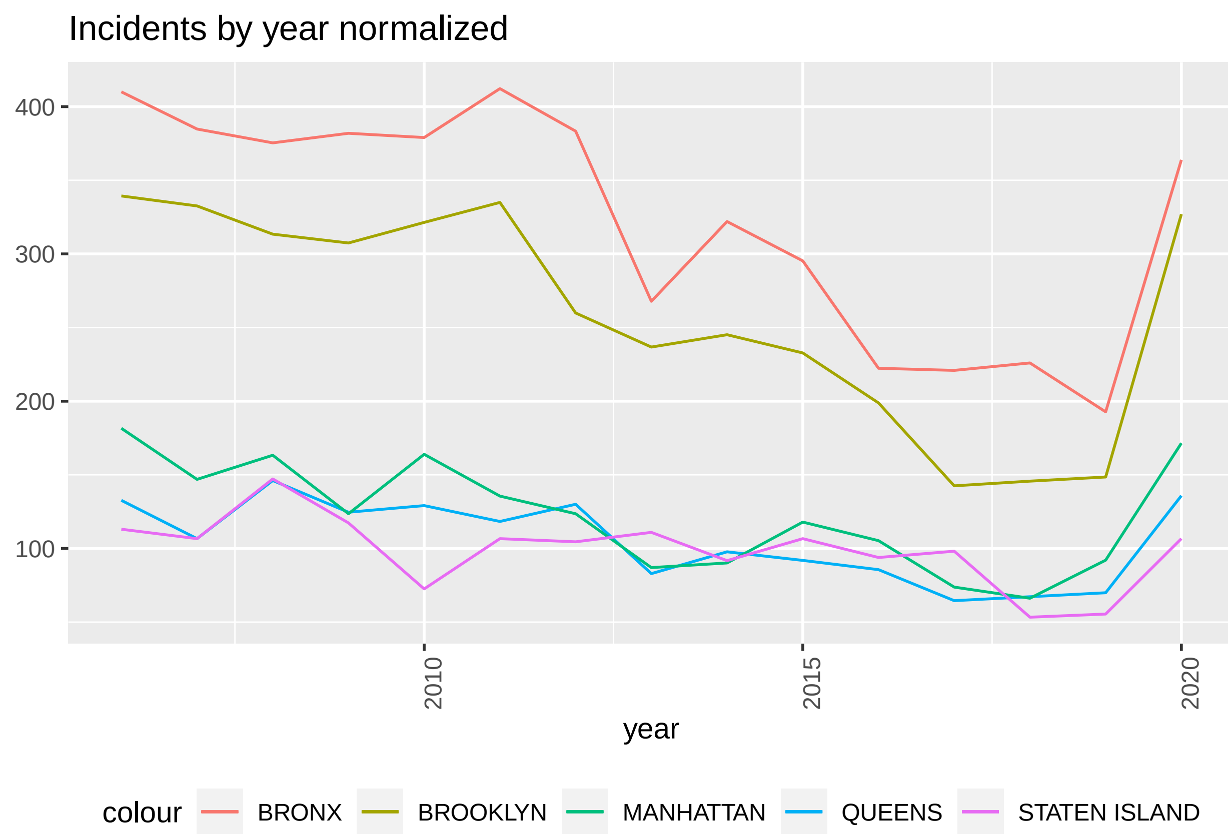
```
bronx_population <- 1385108
brooklyn_population <- 2504700
manhattan_population <- 1585873
queens_population <- 2230722
staten_island_population <- 468730
```

```r
# Use an adjustment number to get numbers back to a
# similar magnitude to the originals.
adjuster<- 1000000
by_date_normalized <- by_date %>%
  mutate(BRONX = (BRONX / bronx_population) * adjuster) %>%
  mutate(BROOKLYN = (BROOKLYN / brooklyn_population) * adjuster) %>%
  mutate(MANHATTAN = (MANHATTAN / manhattan_population) * adjuster) %>%
  mutate(QUEENS = (QUEENS / queens_population) * adjuster) %>%
  mutate(`STATEN ISLAND` = (`STATEN ISLAND` / staten_island_population) * adjuster)


by_date_normalized
```

```
## # A tibble: 15 x 6
##     year BRONX QUEENS BROOKLYN MANHATTAN `STATEN ISLAND`
##    <dbl> <dbl>  <dbl>    <dbl>     <dbl>           <dbl>
##  1  2006  410.  133.      339.     182.            113.
##  2  2007  385.  107.      333.     147.            107.
##  3  2008  375.  146.      313.     163.            147.
##  4  2009  382.  125.      307.     124.            117.
##  5  2010  379.  129.      321.     164.             72.5
##  6  2011  412.  118.      335.     136.            107.
##  7  2012  383.  130.      260.     124.            105.
##  8  2013  268.   82.9     237.      87.0           111.
##  9  2014  322.   97.7     245.      90.2            91.7
## 10  2015  295.   91.9     233.     118.            107.
## 11  2016  222.   85.6     199.     105.             93.9
## 12  2017  221.   64.6     143.      73.8            98.1
## 13  2018  226.   67.2     146.      66.2            53.3
## 14  2019  193.   69.9     149.      92.1            55.5
## 15  2020  364.  136.      327.     172.            107.
```

```r
by_date_normalized %>%
  ggplot(aes(x= year, y= BRONX)) +
  geom_line(aes(color="BRONX")) +
  geom_line(aes(y= QUEENS, color="QUEENS")) +
  geom_line(aes(y= BROOKLYN, color="BROOKLYN")) +
  geom_line(aes(y= MANHATTAN, color="MANHATTAN")) +
  geom_line(aes(y= `STATEN ISLAND`, color="STATEN ISLAND")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents by year normalized", y = NULL)
```



Tidying the unemployment data. Needed to load important data points into columns by column number as the data was in a non-csv text file. Using the max unemployment rate per year.

```
ny_unemployment_filtered <- ny_unemployment %>%
  filter(grepl('New York-Newark-Jersey City', V1))

ny_unemployment_split <- ny_unemployment_filtered %>%
  separate(V1, c("d1", "Year", "Month", "d2", "Rate"), sep=c(105,113,120,172))

ny_unemployment_by_date <- ny_unemployment_split %>%
  select(c(Year, Month, Rate))

ny_unemployment_by_date <- ny_unemployment_by_date %>%
  mutate(Year = as.numeric(Year)) %>%
  filter(Year > 2005 & Year < 2021) %>%
  select(c(Year, Rate))

ny_unemployment_max <- ny_unemployment_by_date %>%
  mutate(year = Year) %>%
  group_by(year) %>%
  summarise(max_rate = max(Rate)) %>%
  mutate(max_rate = as.numeric(as.character(max_rate))) %>%
  arrange(year)
```

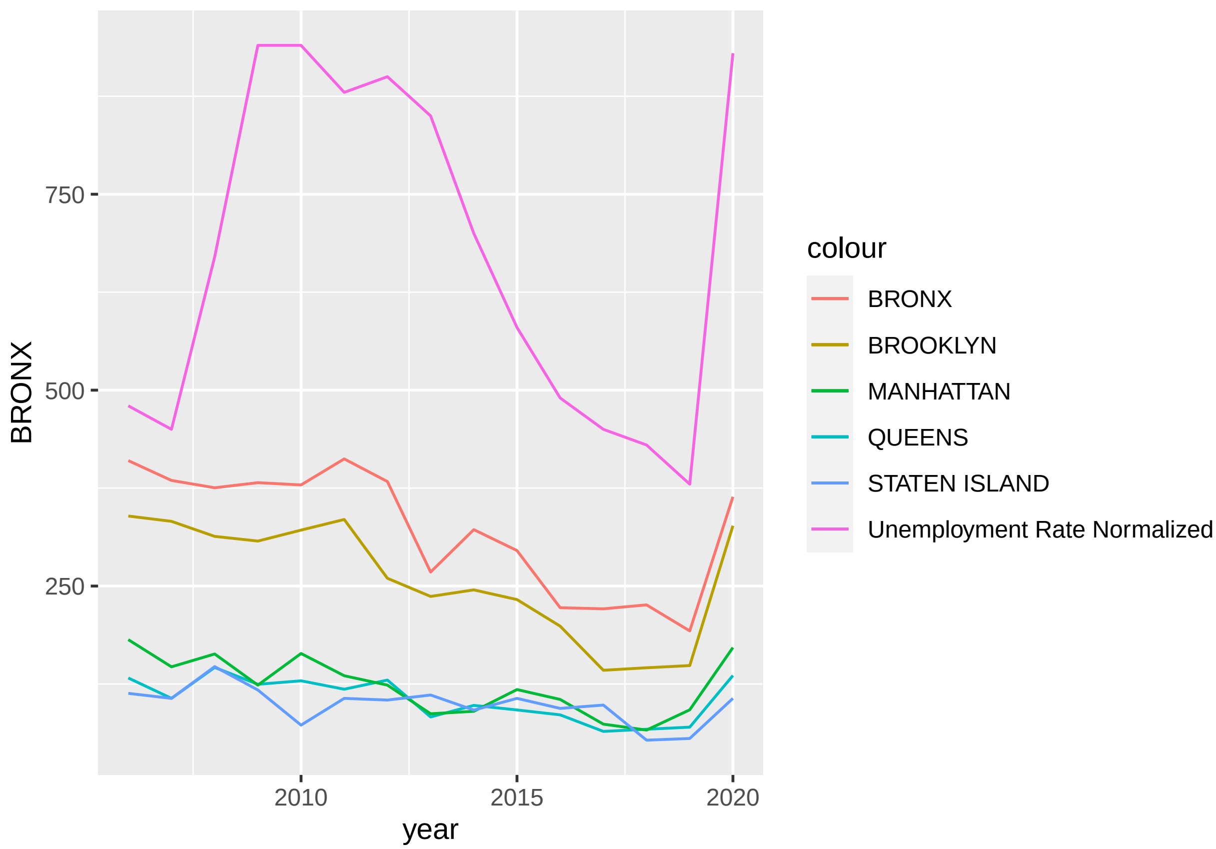
Joining unemployment and shooting incident data and plotting both. It appears there is a very strong correlation between the unemployment rate and shooting incidents.

```r
by_date_norm_w_unemploy <- by_date_normalized %>%
  left_join(ny_unemployment_max)
```

```
## Joining, by = "year"
```

```r
by_date_norm_w_unemploy <- by_date_norm_w_unemploy %>%
  mutate(normalized_rate = max_rate * 100)

by_date_norm_w_unemploy %>%
  ggplot(aes(x= year, y= BRONX)) +
  geom_line(aes(color="BRONX")) +
  geom_line(aes(y= QUEENS, color="QUEENS")) +
  geom_line(aes(y= BROOKLYN, color="BROOKLYN")) +
  geom_line(aes(y= MANHATTAN, color="MANHATTAN")) +
  geom_line(aes(y= `STATEN ISLAND`, color="STATEN ISLAND")) +
  geom_line(aes(y= normalized_rate, color="Unemployment Rate Normalized"))
```




```r
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  ylab("Incidents") +
  labs(title = "Incidents by year normalized with Uneployement Data", y = NULL)
```

```
## List of 4
##  $ axis.text.x     :List of 11
```

```
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.position: chr "bottom"
##  $ y              : NULL
##  $ title          : chr "Incidents by year normalized with Uneployement Data"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

Creating a heat map of the shooting incidents by using Latitude and Longitude. The darker areas do appear to correspond to the Bronx and Brooklyn.

```
incidences_filtered <- nypd_data %>%
  mutate(year = year(OCCUR_DATE)) %>%
  filter(year >= 2020) %>%
  select(c(Latitude, Longitude))

incidences_filtered %>%
  ggplot(aes(Longitude, Latitude)) +
  geom_bin2d(binwidth=.01) +
  geom_tile() +
  scale_fill_gradient(low = "white",high = "steelblue")
```

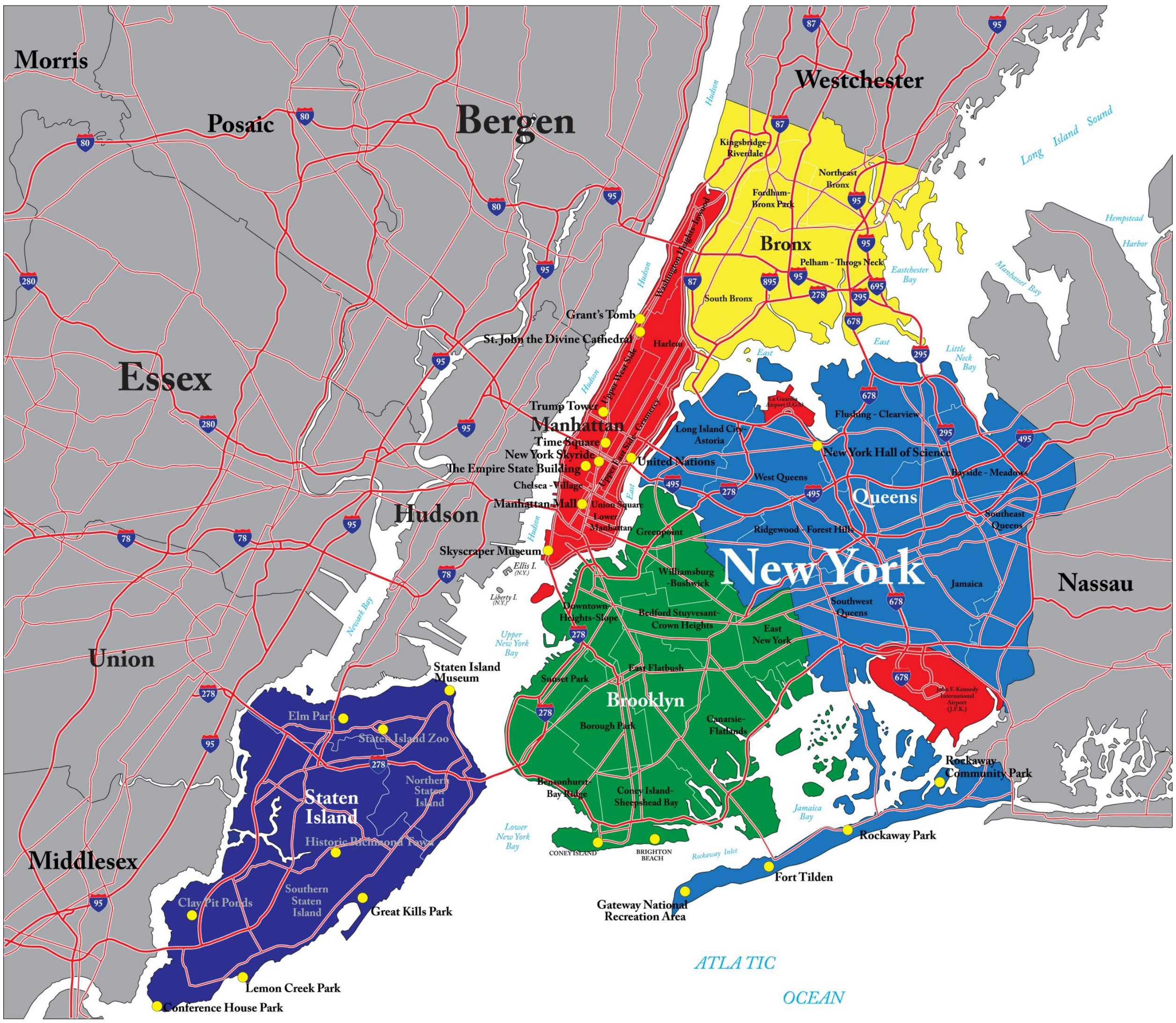
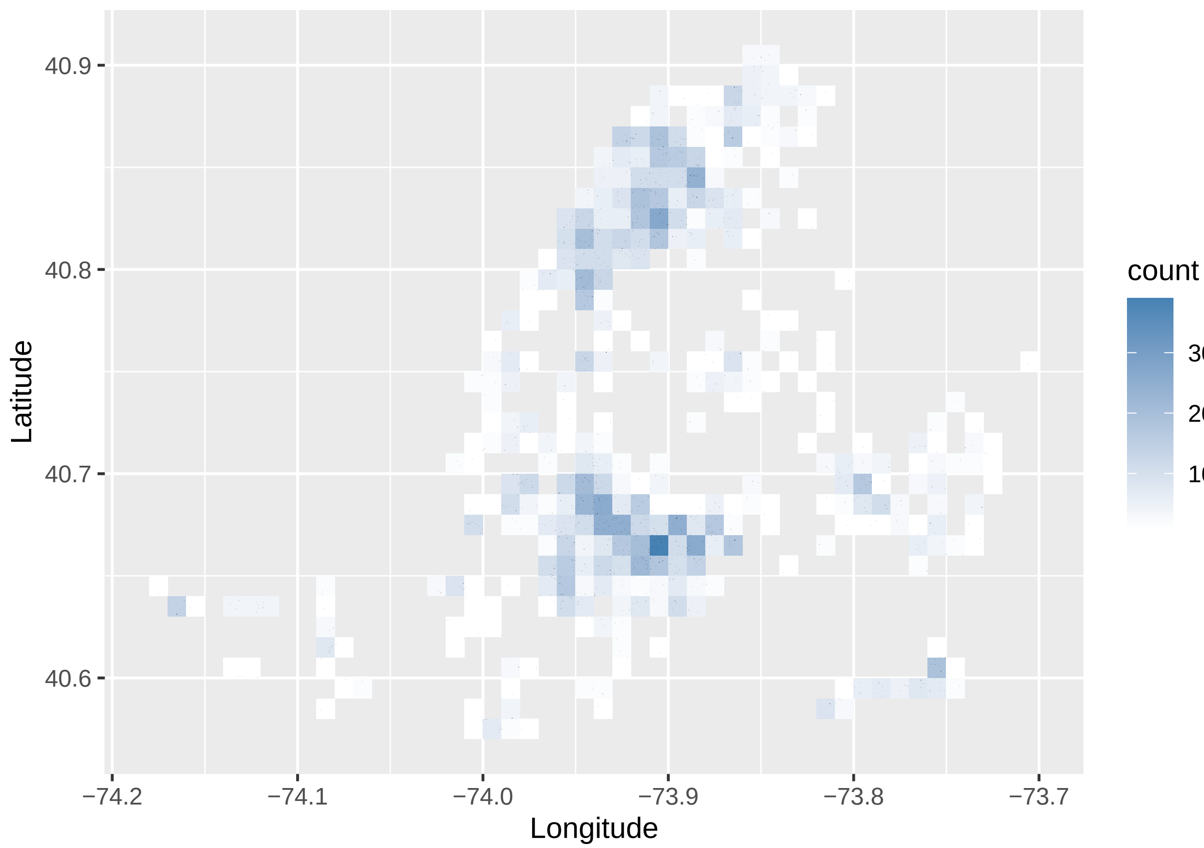


Figure 1: Map of New York City boroughs. stock.adobe.com



Next looking to see if the incidents are tied to the season of the year. Now categorizing by year and month and looking for a trend. The data and the plots show that the incidences are much higher during the warmer/hotter months.

```
by_year_n_month_orig <- nypd_data %>%
  mutate(elem_year = year(OCCUR_DATE)) %>%
  mutate(filter_year = as.numeric(elem_year)) %>%
  mutate(elem_month = sprintf("%02i", month(OCCUR_DATE))) %>%
  unite("Year_W_Month",
        c(elem_year, elem_month),
        sep = " - ",
        remove = FALSE)


by_year_n_month_all <- by_year_n_month_orig %>%
  select(Year_W_Month) %>%
  count(Year_W_Month) %>%
  mutate(n = as.numeric(as.character(n))) %>%
  arrange(Year_W_Month)

by_year_n_month_all
```

```
##     Year_W_Month   n
## 1      2006 - 01 129
## 2      2006 - 02  97
## 3      2006 - 03 102
```

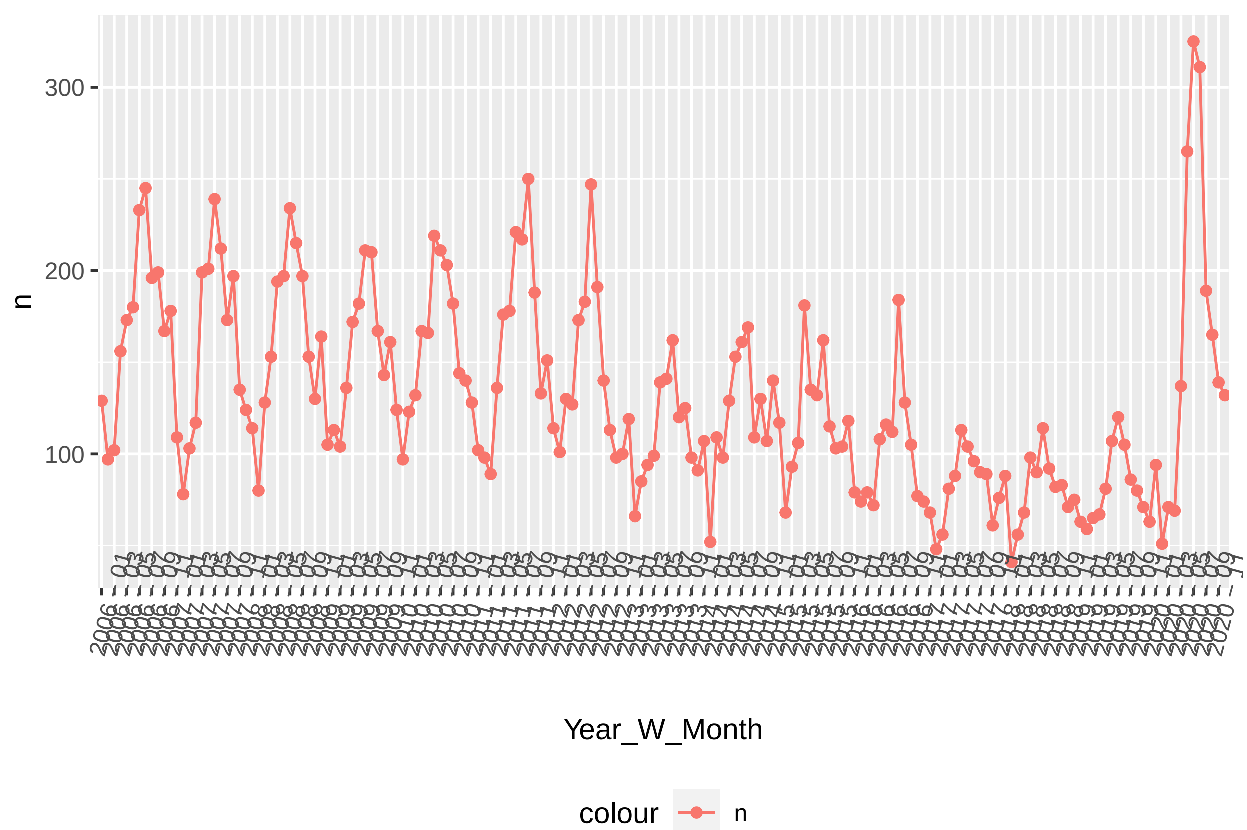
```
## 4       2006 - 04 156
## 5       2006 - 05 173
## 6       2006 - 06 180
## 7       2006 - 07 233
## 8       2006 - 08 245
## 9       2006 - 09 196
## 10      2006 - 10 199
## 11      2006 - 11 167
## 12      2006 - 12 178
## 13      2007 - 01 109
## 14      2007 - 02  78
## 15      2007 - 03 103
## 16      2007 - 04 117
## 17      2007 - 05 199
## 18      2007 - 06 201
## 19      2007 - 07 239
## 20      2007 - 08 212
## 21      2007 - 09 173
## 22      2007 - 10 197
## 23      2007 - 11 135
## 24      2007 - 12 124
## 25      2008 - 01 114
## 26      2008 - 02  80
## 27      2008 - 03 128
## 28      2008 - 04 153
## 29      2008 - 05 194
## 30      2008 - 06 197
## 31      2008 - 07 234
## 32      2008 - 08 215
## 33      2008 - 09 197
## 34      2008 - 10 153
## 35      2008 - 11 130
## 36      2008 - 12 164
## 37      2009 - 01 105
## 38      2009 - 02 113
## 39      2009 - 03 104
## 40      2009 - 04 136
## 41      2009 - 05 172
## 42      2009 - 06 182
## 43      2009 - 07 211
## 44      2009 - 08 210
## 45      2009 - 09 167
## 46      2009 - 10 143
## 47      2009 - 11 161
## 48      2009 - 12 124
## 49      2010 - 01  97
## 50      2010 - 02 123
## 51      2010 - 03 132
## 52      2010 - 04 167
## 53      2010 - 05 166
## 54      2010 - 06 219
## 55      2010 - 07 211
## 56      2010 - 08 203
## 57      2010 - 09 182
```

```
## 58     2010 - 10 144
## 59     2010 - 11 140
## 60     2010 - 12 128
## 61     2011 - 01 102
## 62     2011 - 02  98
## 63     2011 - 03  89
## 64     2011 - 04 136
## 65     2011 - 05 176
## 66     2011 - 06 178
## 67     2011 - 07 221
## 68     2011 - 08 217
## 69     2011 - 09 250
## 70     2011 - 10 188
## 71     2011 - 11 133
## 72     2011 - 12 151
## 73     2012 - 01 114
## 74     2012 - 02 101
## 75     2012 - 03 130
## 76     2012 - 04 127
## 77     2012 - 05 173
## 78     2012 - 06 183
## 79     2012 - 07 247
## 80     2012 - 08 191
## 81     2012 - 09 140
## 82     2012 - 10 113
## 83     2012 - 11  98
## 84     2012 - 12 100
## 85     2013 - 01 119
## 86     2013 - 02  66
## 87     2013 - 03  85
## 88     2013 - 04  94
## 89     2013 - 05  99
## 90     2013 - 06 139
## 91     2013 - 07 141
## 92     2013 - 08 162
## 93     2013 - 09 120
## 94     2013 - 10 125
## 95     2013 - 11  98
## 96     2013 - 12  91
## 97     2014 - 01 107
## 98     2014 - 02  52
## 99     2014 - 03 109
## 100    2014 - 04  98
## 101    2014 - 05 129
## 102    2014 - 06 153
## 103    2014 - 07 161
## 104    2014 - 08 169
## 105    2014 - 09 109
## 106    2014 - 10 130
## 107    2014 - 11 107
## 108    2014 - 12 140
## 109    2015 - 01 117
## 110    2015 - 02  68
## 111    2015 - 03  93
```

```
## 112     2015 - 04 106
## 113     2015 - 05 181
## 114     2015 - 06 135
## 115     2015 - 07 132
## 116     2015 - 08 162
## 117     2015 - 09 115
## 118     2015 - 10 103
## 119     2015 - 11 104
## 120     2015 - 12 118
## 121     2016 - 01  79
## 122     2016 - 02  74
## 123     2016 - 03  79
## 124     2016 - 04  72
## 125     2016 - 05 108
## 126     2016 - 06 116
## 127     2016 - 07 112
## 128     2016 - 08 184
## 129     2016 - 09 128
## 130     2016 - 10 105
## 131     2016 - 11  77
## 132     2016 - 12  74
## 133     2017 - 01  68
## 134     2017 - 02  48
## 135     2017 - 03  56
## 136     2017 - 04  81
## 137     2017 - 05  88
## 138     2017 - 06 113
## 139     2017 - 07 104
## 140     2017 - 08  96
## 141     2017 - 09  90
## 142     2017 - 10  89
## 143     2017 - 11  61
## 144     2017 - 12  76
## 145     2018 - 01  88
## 146     2018 - 02  41
## 147     2018 - 03  56
## 148     2018 - 04  68
## 149     2018 - 05  98
## 150     2018 - 06  90
## 151     2018 - 07 114
## 152     2018 - 08  92
## 153     2018 - 09  82
## 154     2018 - 10  83
## 155     2018 - 11  71
## 156     2018 - 12  75
## 157     2019 - 01  63
## 158     2019 - 02  59
## 159     2019 - 03  65
## 160     2019 - 04  67
## 161     2019 - 05  81
## 162     2019 - 06 107
## 163     2019 - 07 120
## 164     2019 - 08 105
## 165     2019 - 09  86
```

```
## 166    2019 - 10  80
## 167    2019 - 11  71
## 168    2019 - 12  63
## 169    2020 - 01  94
## 170    2020 - 02  51
## 171    2020 - 03  71
## 172    2020 - 04  69
## 173    2020 - 05 137
## 174    2020 - 06 265
## 175    2020 - 07 325
## 176    2020 - 08 311
## 177    2020 - 09 189
## 178    2020 - 10 165
## 179    2020 - 11 139
## 180    2020 - 12 132
```

```r
ggplot(by_year_n_month_all, aes(x= Year_W_Month, y= n)) +
geom_point(aes(color="n")) +
geom_line(aes(y= n, color="n", group=1)) +
theme(legend.position = "bottom",
      axis.text.x = element_text(angle = 75)) +
scale_x_discrete(breaks=by_year_n_month_all$Year_W_Month[seq(1,length(by_year_n_month_all$Year_W_Month)
```




```r
labs(title = "Total Incidents by month", y = NULL)
```

```
## $y
```

```
## NULL
##
## $title
## [1] "Total Incidents by month"
##
## attr(,"class")
## [1] "labels"
```

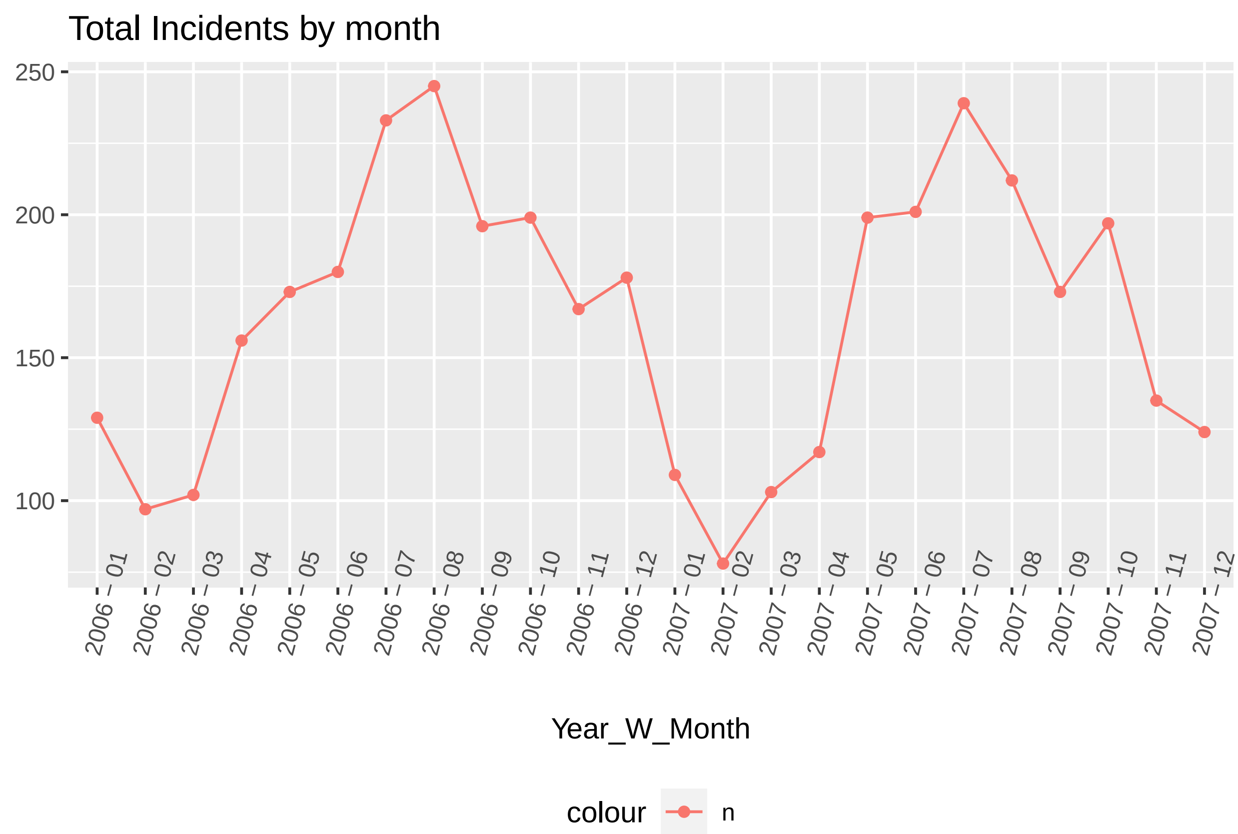
Let's analyze a smaller time frame to get a paired down view of how warmer months are tied to a higher incident count.

```
by_year_n_month <- by_year_n_month_orig %>%
  filter(filter_year < 2008) %>%
  select(Year_W_Month) %>%
  count(Year_W_Month) %>%
  mutate(n = as.numeric(as.character(n))) %>%
  arrange(Year_W_Month)

by_year_n_month
```

```
##    Year_W_Month   n
## 1     2006 - 01 129
## 2     2006 - 02  97
## 3     2006 - 03 102
## 4     2006 - 04 156
## 5     2006 - 05 173
## 6     2006 - 06 180
## 7     2006 - 07 233
## 8     2006 - 08 245
## 9     2006 - 09 196
## 10    2006 - 10 199
## 11    2006 - 11 167
## 12    2006 - 12 178
## 13    2007 - 01 109
## 14    2007 - 02  78
## 15    2007 - 03 103
## 16    2007 - 04 117
## 17    2007 - 05 199
## 18    2007 - 06 201
## 19    2007 - 07 239
## 20    2007 - 08 212
## 21    2007 - 09 173
## 22    2007 - 10 197
## 23    2007 - 11 135
## 24    2007 - 12 124
```

```
ggplot(by_year_n_month, aes(x= Year_W_Month, y= n)) +
geom_point(aes(color="n")) +
geom_line(aes(y= n, color="n", group=1)) +
theme(legend.position = "bottom",
      axis.text.x = element_text(angle = 75)) +
labs(title = "Total Incidents by month", y = NULL)
```



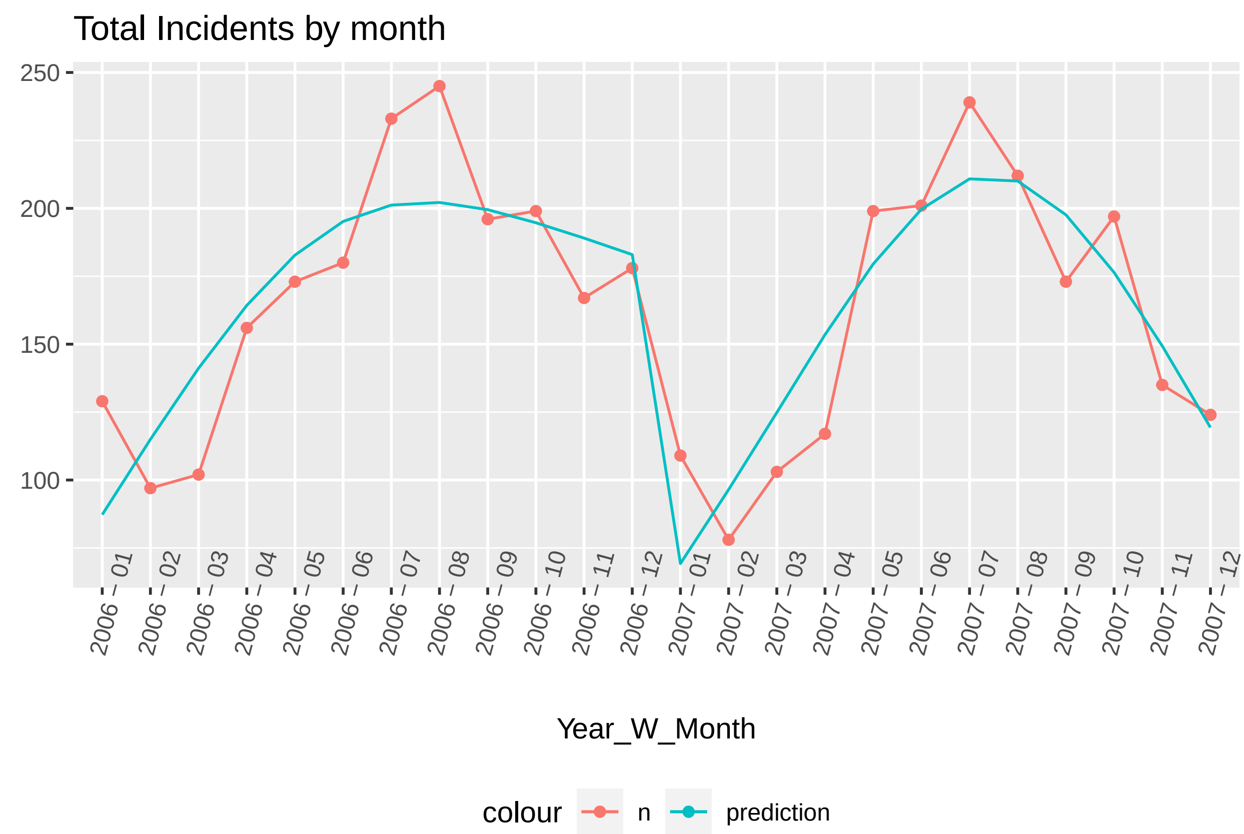
Next, let's see if a model can be created from the data that fits the trend for higher incident counts during the warmer months. Using a splines model with a degree of 5. As shown, a model can be fitted to the data with an acceptable degree of accuracy.

```
updated_with_numeric <- by_year_n_month %>%
  mutate(Year_W_Month_Num = gsub(" - ","",as.character(Year_W_Month))) %>%
  mutate(Year_W_Month_Num = as.numeric(Year_W_Month_Num))


# make a model with spline degree 5
mod5 <- lm(n ~ ns(Year_W_Month_Num, 5), data = updated_with_numeric)


updated_with_numeric <- updated_with_numeric %>%
  mutate(prediction = predict(mod5))

ggplot(updated_with_numeric, aes(x= Year_W_Month, y= n)) +
geom_point(aes(color="n")) +
geom_line(aes(y= n, color="n", group=1)) +
geom_line(aes(y= prediction, color="prediction", group=1)) +
theme(legend.position = "bottom",
      axis.text.x = element_text(angle = 75)) +
labs(title = "Total Incidents by month", y = NULL)
```



Summary

The plot showing the shooting incidents for each borough per year shows a lessening of incidents around 2016 with a large jump in 2020. One might conclude that the Covid-19 pandemic and associated economic issues caused the jump in incidents. The addition of the unemployment data does appear to support that the unemployment during the start of the pandemic did correlate highly with the number of shooting incidents. The original plot shows that Brooklyn has the highest number of shooting incidents. However, normalized for population size, the Bronx has the highest number. The original plot for Staten Island might lead one to believe that Staten Island is considerably safer, however, when normalized for population size, Staten Island, Manhatten, and Queens have similar trends.

It is often said that warmer or hotter months have a strong correlation to increased violent crime and this data does support that. First a plot of all year/month combinations was generated and there was an obvious pattern of seasonal increases. Next a smaller range was selected in order for a model to be fitted. Using splines of degree five the model shows a good fit to the data,

Bias concerns

One bias I had at first was that crime was simply higher in Brooklyn and the Bronx. Originally, I hadn't thought about population size. The 2010 census data from https://www1.nyc.gov/assets/planning/download/pdf/planning-level/nyc-population/historical-population/nyc_total_pop_1900-2010.pdf was used to normalize the numbers. Each borough has had fairly steady, but moderate population growth over the last twenty years so although I only used the population from 2010 to normalize, it is indicative of the population size.

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] scales_1.1.1    forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     tidyr_1.1.4     tibble_3.1.6    ggplot2_3.3.5
##  [9] tidyverse_1.3.1 readr_2.1.1     lubridate_1.8.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1 xfun_0.29        haven_2.4.3      colorspace_2.0-2
##  [5] vctrs_0.3.8      generics_0.1.1   htmltools_0.5.2  yaml_2.2.1
##  [9] utf8_1.2.2       rlang_0.4.12     pillar_1.6.5     glue_1.6.0
## [13] withr_2.4.3      DBI_1.1.2        dbplyr_2.1.1     modelr_0.1.8
## [17] readxl_1.3.1     lifecycle_1.0.1  munsell_0.5.0    gtable_0.3.0
## [21] cellranger_1.1.0 rvest_1.0.2      evaluate_0.14    labeling_0.4.2
## [25] knitr_1.37       tzdb_0.2.0       fastmap_1.1.0    fansi_1.0.2
## [29] highr_0.9        broom_0.7.12     Rcpp_1.0.8       backports_1.4.1
## [33] jsonlite_1.7.3   farver_2.1.0     fs_1.5.2         hms_1.1.1
## [37] digest_0.6.29    stringi_1.7.6    grid_4.1.2       cli_3.1.0
## [41] tools_4.1.2      magrittr_2.0.1   crayon_1.4.2     pkgconfig_2.0.3
## [45] ellipsis_0.3.2   xml2_1.3.3       reprex_2.0.1     assertthat_0.2.1
## [49] rmarkdown_2.11   httr_1.4.2       rstudioapi_0.13  R6_2.5.1
## [53] compiler_4.1.2
```