NY Shooting Data

Antony Sequeira

2/9/2022

Following script installs the required libraries in Mac OSX

This section can be copied to a file or input into R console.
You could also download it from my repo at https://github.com/asequeir-edu-2022/dtsa5301final

```
#!/usr/bin/env Rscript
r = getOption("repos")
r["CRAN"] = "http://cran.us.r-project.org"
options(repos = r)

print("Installing R libraries")
install.packages("chron")
install.packages("tidyverse")
install.packages("tinytex")

tinytex::install_tinytex()
```

Overview

For fulfillment of DTSA-5301 finals *NYPD Shooting Incident Data Report* part of the assignment.

The data is free government data about shooting incidents in New York city. The goal is to get, cleanup, analyse, and present the NYPD shooting data. The main goal of the analysis is to find out how the incidents relate to the different factors.

Data source

Data is downloaded from CSV link in
https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

```r
ny_url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
ny_data_raw <- read_csv(ny_url)
```

Clean up data

We clean up the data mainly through the following three operations:
- variables to factor for appropriate columns

- date types from strings
- remove unneeded columns

```
ny_data <- ny_data_raw %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
  mutate(OCCUR_TIME = chron(times=OCCUR_TIME)) %>%
  mutate(BORO = factor(BORO)) %>%
  mutate(PRECINCT = factor(PRECINCT)) %>%
  mutate(PERP_AGE_GROUP = factor(PERP_AGE_GROUP)) %>%
  mutate(PERP_SEX = factor(PERP_SEX)) %>%
  mutate(PERP_RACE = factor(PERP_RACE)) %>%
  mutate(VIC_AGE_GROUP = factor(VIC_AGE_GROUP)) %>%
  mutate(VIC_SEX = factor(VIC_SEX)) %>%
  mutate(VIC_RACE = factor(VIC_RACE)) %>%
  select (-c(JURISDICTION_CODE, LOCATION_DESC, X_COORD_CD, Y_COORD_CD, Latitude, Longitude, INCIDENT_KE
```

Missing data columns and plans to handle them

There is missing data in the following columns:

```
names(which(colSums(is.na(ny_data_raw)) > 0))
```

```
## [1] "JURISDICTION_CODE" "LOCATION_DESC"     "PERP_AGE_GROUP"
## [4] "PERP_SEX"          "PERP_RACE"
```

Missing data in factor columns `PERP_SEX` etc. are handled already as a factor.
I do not plan to use `JURISDICTION_CODE` and `LOCATION_DESC`.
So, for this data, nothing more needs to be done for missing data handling.

Summary of the cleaned up data

```
summary(ny_data)
```

```
##    OCCUR_DATE           OCCUR_TIME                  BORO         PRECINCT
##  Min.   :2006-01-01   Min.   :00:00:00   BRONX        :6701   75     : 1375
##  1st Qu.:2008-12-31   1st Qu.:03:20:00   BROOKLYN     :9734   73     : 1284
##  Median :2012-02-27   Median :15:00:00   MANHATTAN    :2922   67     : 1101
##  Mean   :2012-10-05   Mean   :12:33:07   QUEENS       :3532   79     :  921
##  3rd Qu.:2016-03-02   3rd Qu.:20:45:00   STATEN ISLAND: 696   44     :  841
##  Max.   :2020-12-31   Max.   :23:59:00                        47     :  818
##                                                               (Other):17245
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX              PERP_RACE
##  Mode :logical           18-24  :5508   F   :  335   BLACK         :10025
##  FALSE:19085             25-44  :4714   M   :13490   WHITE HISPANIC: 1988
##  TRUE :4500              UNKNOWN:3148   U   : 1499   UNKNOWN       : 1836
##                          <18    :1368   NA's: 8261   BLACK HISPANIC: 1096
##                          45-64  : 495                WHITE         :  255
##                          (Other):  57                (Other)       :  124
##                          NA's   :8295                NA's          : 8261
##  VIC_AGE_GROUP   VIC_SEX                          VIC_RACE
##  <18    : 2525   F: 2204   AMERICAN INDIAN/ALASKAN NATIVE:    9
##  18-24  : 9003   M:21370   ASIAN / PACIFIC ISLANDER      :  327
##  25-44  :10303   U:   11   BLACK                         :16869
##  45-64  : 1541             BLACK HISPANIC                : 2245
##  65+    :  154             UNKNOWN                       :   65
##  UNKNOWN:   59             WHITE                         :  620
##                             WHITE HISPANIC                : 3450
##    Lon_Lat
##  Length:23585
##  Class :character
##  Mode  :character
##
##
##
##
```

Turning columns into factors shows total counts for columns such as `BORO`.
The summary also shows the breakdown of total incidents by factor type columns such as `PERP_AGE_GROUP`.

Visualization and Analysis

Generate necessary analysis ready data.

Prepare a few different slices of the data for visualizations.

```
ny_data_sum <- ny_data %>%
  group_by(BORO, VIC_AGE_GROUP) %>%
  mutate(BORO_BY_VIC_AGE = n()) %>%  # occurrences by victim age group by boro
  ungroup()  %>%
  group_by(month = lubridate::floor_date(OCCUR_DATE, "month")) %>%
  mutate(month_sum = sum(n())) %>% # occurrences by month
  ungroup() %>%
  group_by(month, BORO) %>%
  mutate(month_by_boro = sum(n())) %>%
  ungroup()
```

```
ny_data_sum2 <- ny_data %>%
  group_by(BORO, VIC_AGE_GROUP) %>%
  summarize(BORO_BY_VAG = n(), .groups = 'drop') %>%
  ungroup()
```

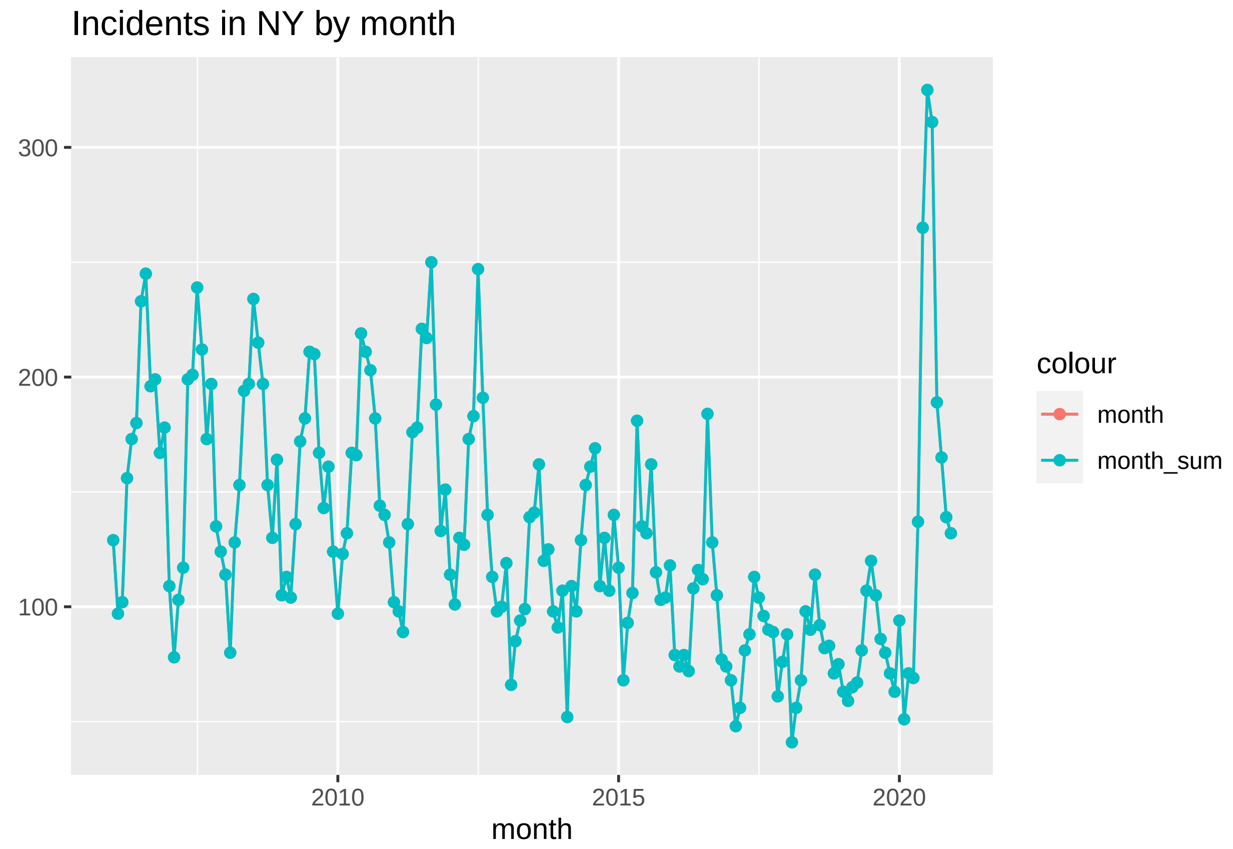
```
ny_data_sum3 <- ny_data %>%
  group_by(month = lubridate::floor_date(OCCUR_DATE, "month")) %>%
  summarise(month_sum = sum(n())) %>% # occurrences by month
  ungroup()
```

Visualizations

Variations by the month of the year

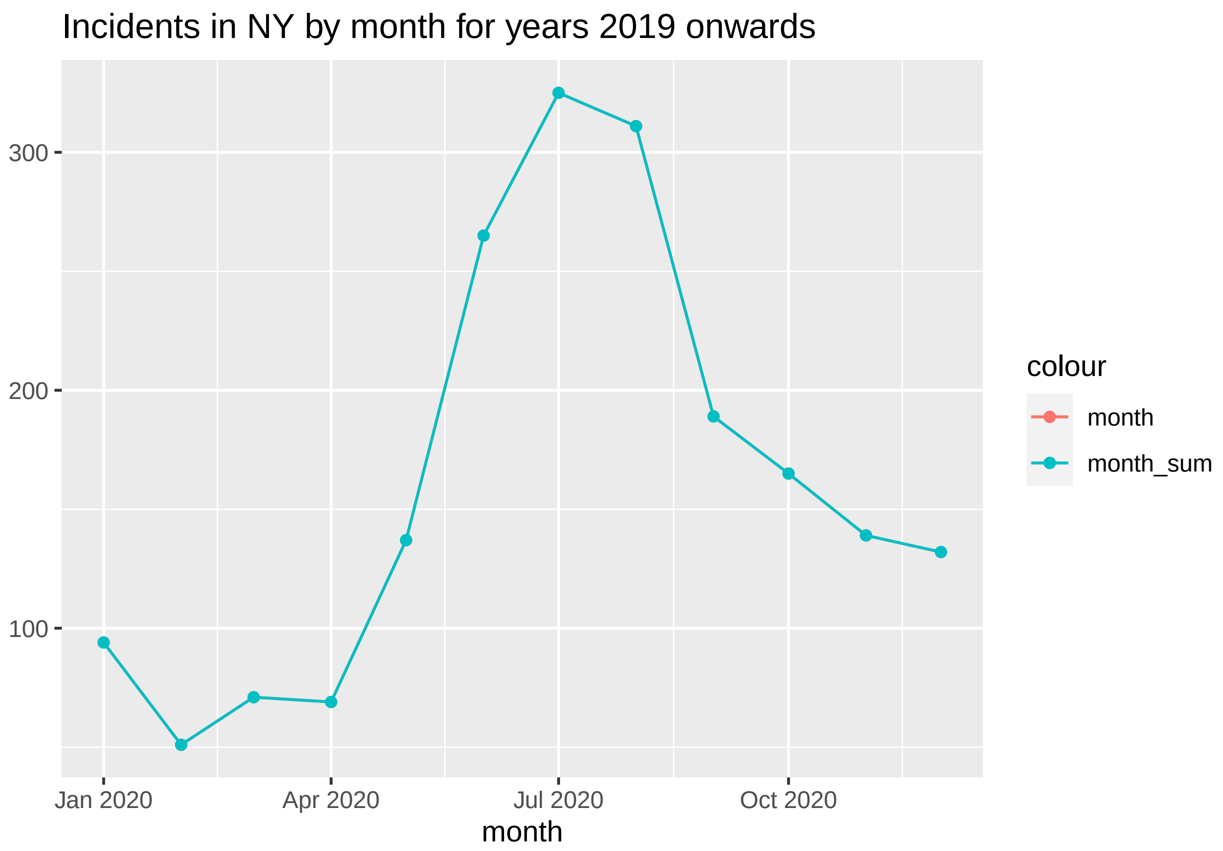
We plot the total incidents by month.

```
ny_data_sum3 %>%
ggplot(aes(x = month, y = month_sum)) +
geom_line(aes(color = "month")) +
geom_point(aes(color = "month")) +
geom_line(aes(y=month_sum, color = "month_sum")) +
geom_point(aes(y=month_sum, color = "month_sum")) +
labs(title = "Incidents in NY by month", y = NULL)
```



The above plot shows that the number of incidents vary seasonally.
To see the 2020 peak more clearly, we plot a shorter time span.

```
ny_data_sum3 %>%
  filter(year(month) > 2019) %>%
  ggplot(aes(x = month, y = month_sum)) +
  geom_line(aes(color = "month")) +
  geom_point(aes(color = "month")) +
  geom_line(aes(y=month_sum, color = "month_sum")) +
  geom_point(aes(y=month_sum, color = "month_sum")) +
  labs(title = "Incidents in NY by month for years 2019 onwards", y = NULL)
```

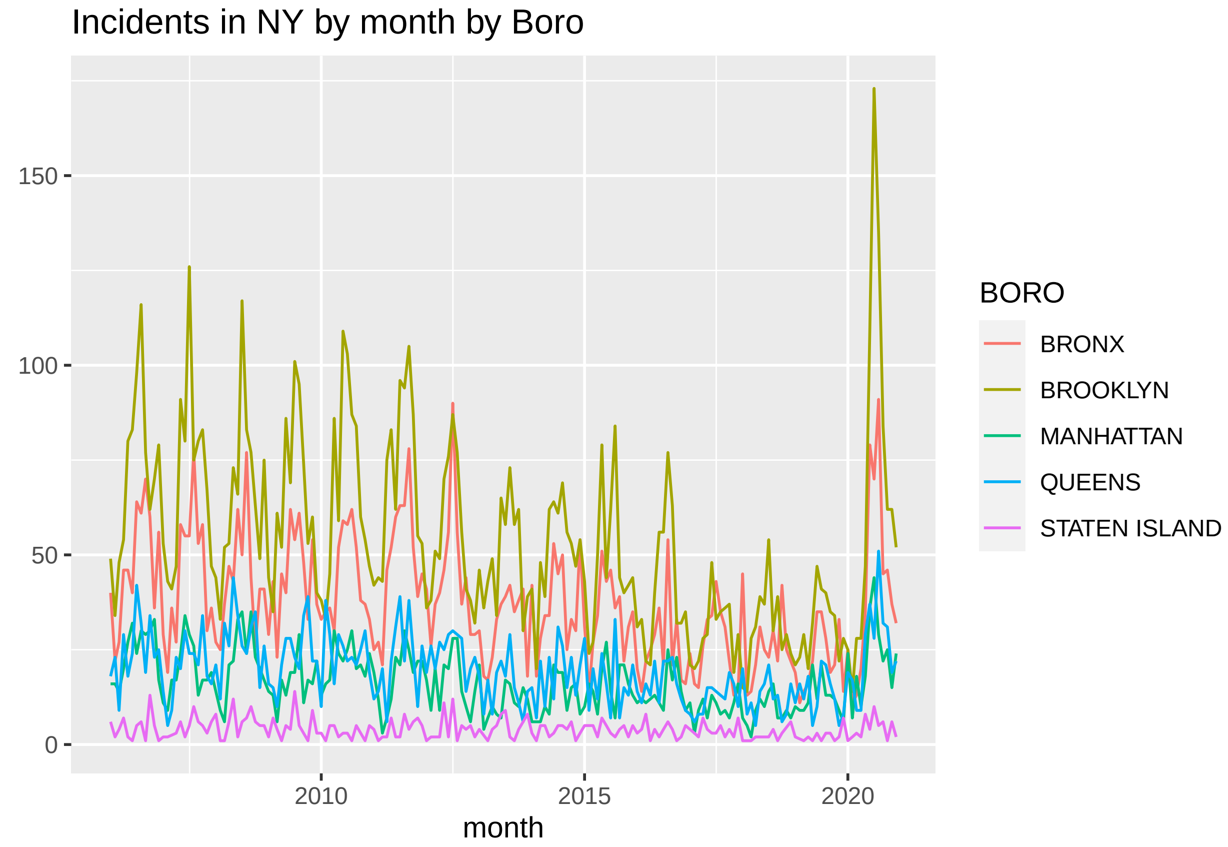


This shows a clear peak in July of 2020.

Variations by borough

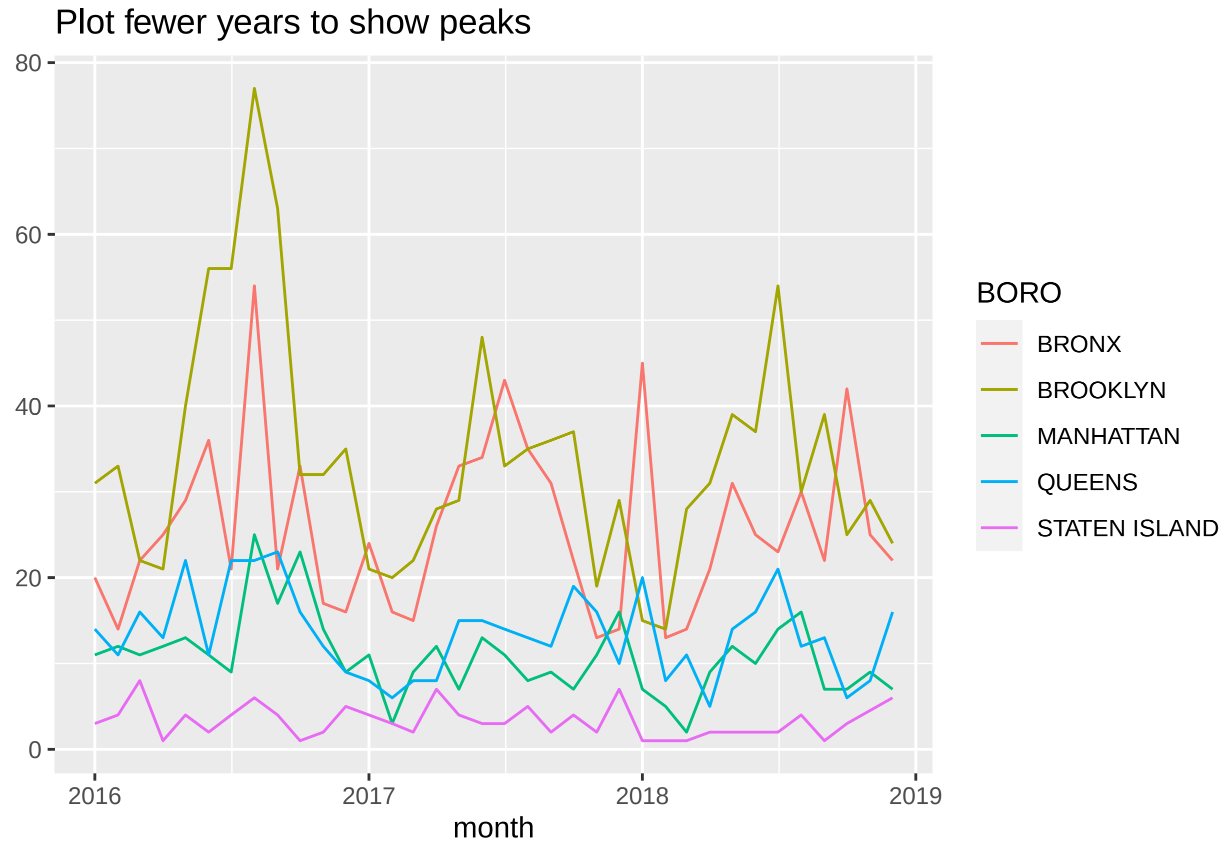
Generate the counts by borough.

```
ny_data_sum %>%
  group_by(month_by_boro) %>%
  ggplot(aes(x=month, y=month_by_boro, group=BORO, color=BORO)) +
  geom_line() +
  labs(title = "Incidents in NY by month by Boro", y = NULL)
```



Get a smaller time span to see a zoomed in view.

```
ny_data_sum %>%
  filter(year(month) > 2015) %>%
  filter(year(month) < 2019) %>%
  group_by(month_by_boro) %>%
  ggplot(aes(x=month, y=month_by_boro, group=BORO, color=BORO)) +
  geom_line() +
  labs(title = "Plot fewer years to show peaks", y = NULL)
```



Analysis

The plots show the following:

- seasonal peaks mostly in summer
- higher levels of incidents based on the boro - Staten Island is lowest and Bronx and Brooklyn seem to be the higher end.
- the incidents show unusual higher numbers in first quarter of 2020

Questions raised by the visualization and analysis (to be investigated)

- population of boros (Staten Island might have much smaller population) may be too different
- factors not in data such as income
- number of police officers per person

Bias

There could be multiple sources of bias in the NY shooting data

- the data collection may be biased, it is possible that not all shootings are reported
- the standard race categories may not reflect the reality of the NY demographics

I have tried to focus on the boro and seasonality of the data to reduce bias.

Modeling

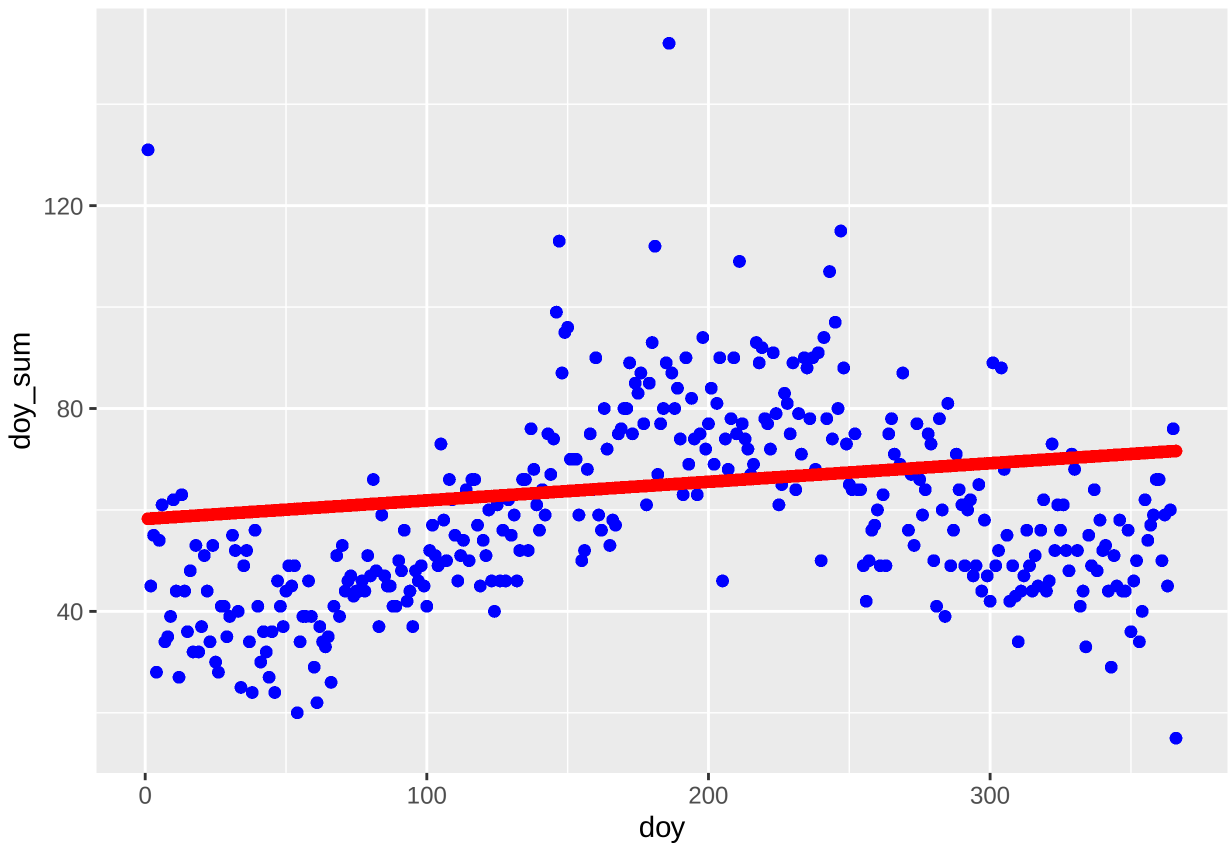
```
ny_data_doy <- ny_data_sum %>%
  filter(year(OCCUR_DATE)< 2020) %>% # avoiding covid years
  group_by(doy = yday(OCCUR_DATE)) %>%
  mutate(doy_sum = sum(n())) %>%
  ungroup()

mod <- lm(doy_sum ~ doy, data=ny_data_doy)
summary(mod)
```

```
##
## Call:
## lm(formula = doy_sum ~ doy, data = ny_data_doy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.619 -14.855  -2.395  11.545  86.971
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 58.219202   0.304236   191.4   <2e-16 ***
## doy          0.036611   0.001414    25.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.09 on 21635 degrees of freedom
## Multiple R-squared:  0.03006,    Adjusted R-squared:  0.03002
## F-statistic: 670.6 on 1 and 21635 DF,  p-value: < 2.2e-16
```

```
ny_data_pred <-  ny_data_doy %>% mutate(doy_sum_pred = predict(mod))

ny_data_pred %>% ggplot() +
  geom_point(aes( x = doy, y = doy_sum), color = "blue") +
  # compare with predicted
  geom_point(aes( x = doy, y = doy_sum_pred), color = "red")
```



The plot clearly shows that the linear model does not fit the data of incidents given a day of the year. It looks like we need a different model (other than linear) to predict incidents given any day of the year.

I hope to learn more about statistical modelling in future data science courses so I can model such data better.

Session info

```
## R version 4.1.2 (2021-11-01)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/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 utils     datasets  methods   base
##
## other attached packages:
##  [1] chron_2.3-56    lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0
##  [5] dplyr_1.0.7     purrr_0.3.4     readr_2.1.1     tidyr_1.1.4
##  [9] tibble_3.1.6    ggplot2_3.3.5   tidyverse_1.3.1
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8       assertthat_0.2.1 digest_0.6.29    utf8_1.2.2
##  [5] R6_2.5.1         cellranger_1.1.0 backports_1.4.1  reprex_2.0.1
##  [9] evaluate_0.14    highr_0.9        httr_1.4.2       pillar_1.6.4
## [13] rlang_0.4.12     curl_4.3.2       readxl_1.3.1     rstudioapi_0.13
## [17] rmarkdown_2.11   labeling_0.4.2   bit_4.0.4        munsell_0.5.0
## [21] broom_0.7.11     compiler_4.1.2   modelr_0.1.8     xfun_0.29
## [25] pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.1 fansi_1.0.2
## [29] crayon_1.4.2     tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.3
## [33] grid_4.1.2       jsonlite_1.7.2   gtable_0.3.0     lifecycle_1.0.1
## [37] DBI_1.1.2        magrittr_2.0.1   scales_1.1.1     cli_3.1.0
## [41] stringi_1.7.6    vroom_1.5.7      farver_2.1.0     fs_1.5.2
## [45] xml2_1.3.3       ellipsis_0.3.2   generics_0.1.1   vctrs_0.3.8
## [49] tools_4.1.2      bit64_4.0.5      glue_1.6.0       hms_1.1.1
## [53] parallel_4.1.2   fastmap_1.1.0    yaml_2.2.1       colorspace_2.0-2
## [57] rvest_1.0.2      knitr_1.37       haven_2.4.3
```