Looking at Police Misconduct Incidents in NYC

Becca Tramel

Looking at the Data

The file *locationsstrat2* was built using FiveThirtyEight's data on police misconduct settlements (https://github.com/fivethirtyeight/police-settlements). It contains 100 random cases from each of the years from 2008 - 2018. These years were chosen because each has at least 1000 cases in the original data set.

We then cleaned the *location* column and used the package *ggmap* to obtain the latitude and longitude of each incident.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
## Warning: package 'tibble' was built under R version 3.6.3
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'stringr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
locationstrat2 <- read_csv(url("https://raw.githubusercontent.com/becca-t/policemisconductnyc/ma
in/locationsstrat2"))
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   incident_date = col_date(format = ""),
##   filed_date = col_date(format = ""),
##   closed_date = col_datetime(format = ""),
##   amount_awarded = col_double(),
##   calendar_year = col_double(),
##   incident_year = col_double(),
##   filed_year = col_double(),
##   other_expenses = col_logical(),
##   collection = col_logical(),
##   total_incurred = col_logical(),
##   court = col_logical(),
##   docket_number = col_logical(),
##   matter_name = col_logical(),
##   case_outcome = col_logical(),
##   lon = col_double(),
##   lat = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

Using ggmap() to Create a Map

We can use ggmap (http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf) to create a map of NYC. Then we'll use *ggplot2* to plot our locations on top of this map. At first we'll plot them together, then break them up by year.

I found this blog (https://www.jessesadler.com/post/geocoding-with-r/) incredibly helpful in learning about some of the features of the ggmap package.

First we'll enter our key for Google's API.

```
## Warning: package 'ggmap' was built under R version 3.6.3
```

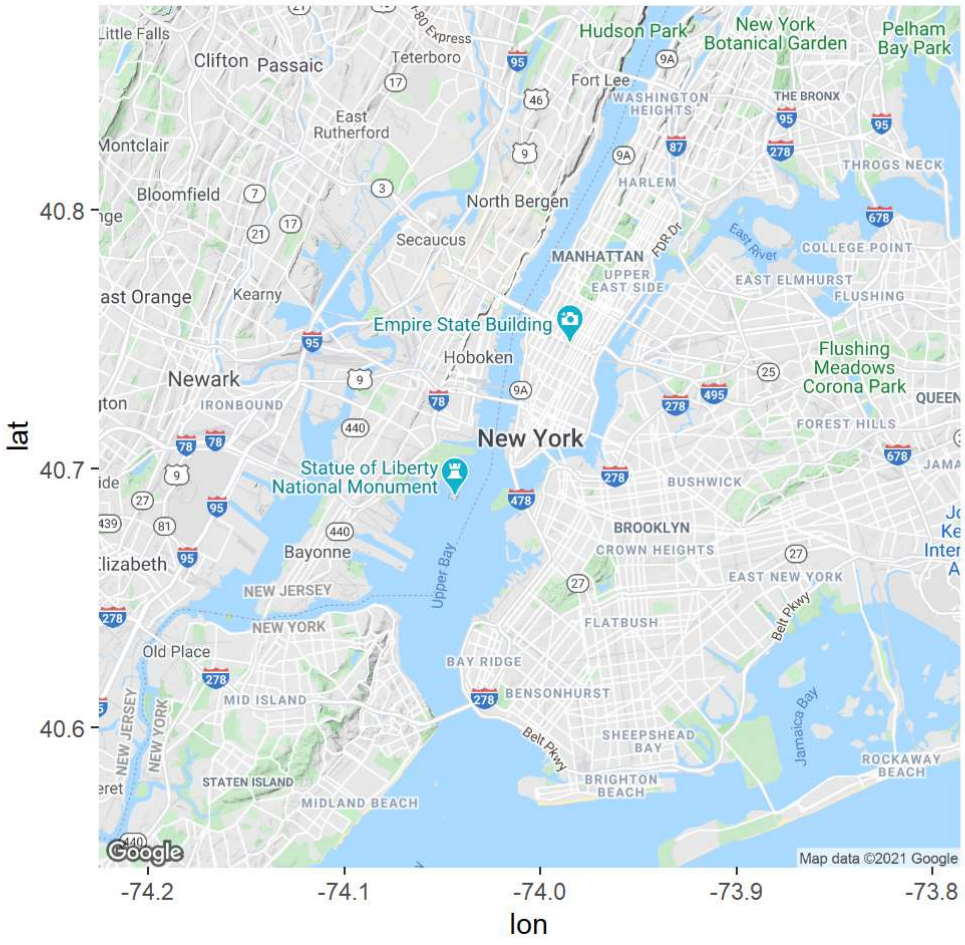
```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

Then we'll create our map of NYC.

```
ny_map <- get_googlemap(center = c(lon = -74.0060, lat = 40.7128), zoom = 11)
```

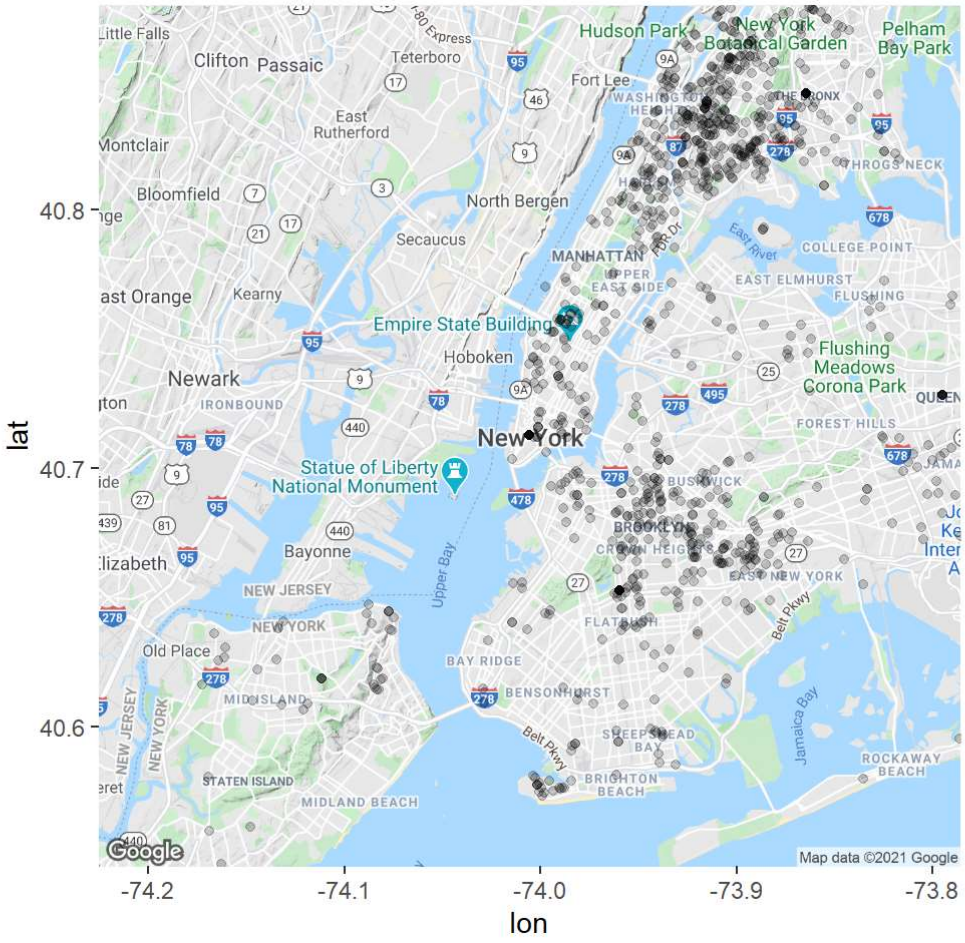
```
ggmap(ny_map)
```



Now we can use *ggplot2* to add our data to the map. I'm going to make two versions below. One will include all of the data in one map, and the other will create a map for each year.

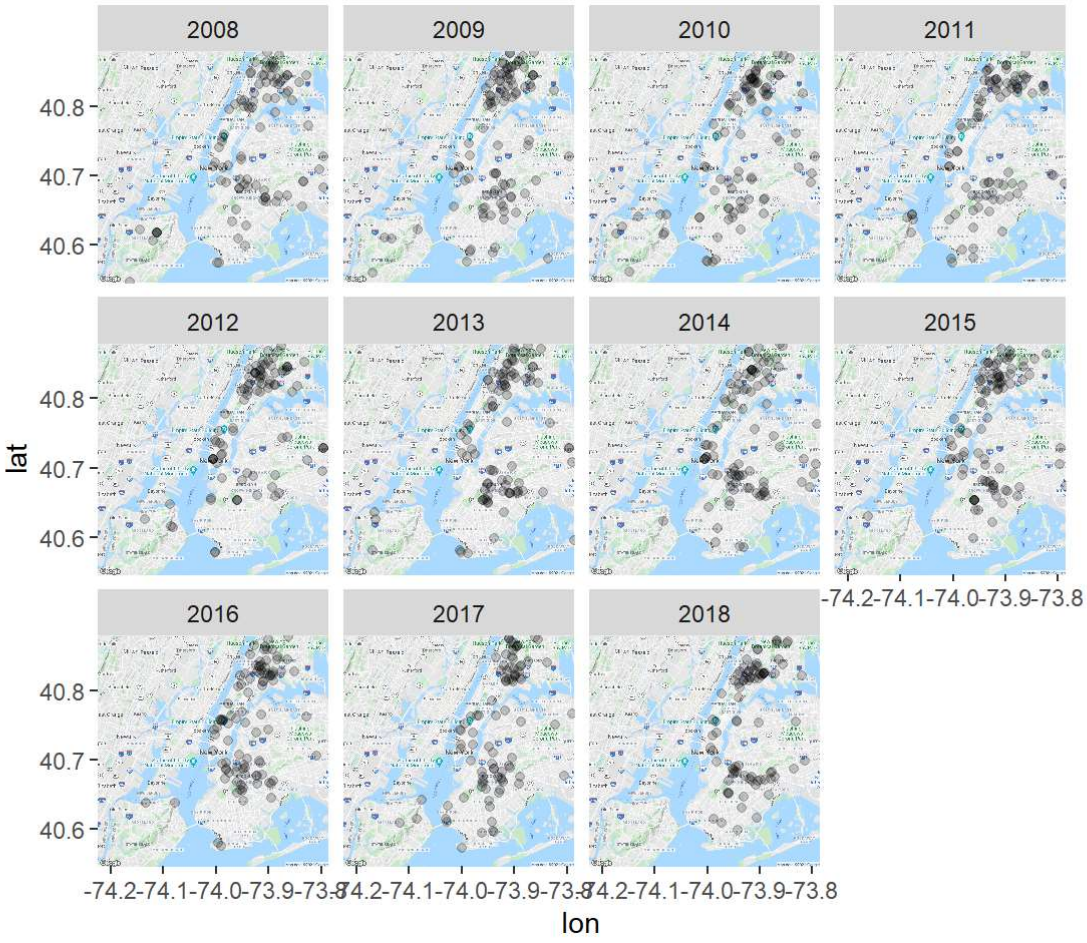
```
# Making a single map:
ggmap(ny_map) +
  geom_point(data = locationstrat2, aes(x = lon, y = lat), alpha = 0.2)
```

```
## Warning: Removed 93 rows containing missing values (geom_point).
```




```
# Making a map for each year:
ggmap(ny_map) +
  geom_point(data = locationstrat2, aes(x = lon, y = lat), alpha = 0.2) +
  facet_wrap(~incident_year)
```

```
## Warning: Removed 93 rows containing missing values (geom_point).
```


2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
lat
40.8
40.7
40.6
-74.2-74.1-74.0-73.9-73.8
lon