Geocoding with FiveThirtyEight's Police Misconduct Data

Becca Tramel

FiveThirtyEight and The Marshall Project's Police Misconduct Data

FiveThirtyEight and the Marshall Project collected data about police misconduct cases in major US cities via a FOIA request. You can read about it here (https://github.com/fivethirtyeight/police-settlements). While many cities reported limited data on the locations of the incidents, New York City provided the adresses.

In total, there are 32,632 cases described in the data reported by New York City. In the rest of this file I will walk through the process of taking a stratified sample of this data and using Google's API to find the latitude and longitude for each case in the sample.

Taking a Sample Stratified by Year

Each case in the data from New York City is associated with three years: the year of the incident (*incident_year*), the year the case was filed (*filed_year*), and the year the case was closed (*calendar_year*). For my purposes I was interested in patterns in the year and location of the incidents.

The incident years range from 1980 (1 case) to 2019 (154 cases), but the large majority of cases occur between 2008 and 2018. For each year in this time range there are more than 1100 cases, with a maximum of 4039 from 2012.

To look for patterns by year, I used the *tidyverse* package to take a random sample of 100 cases from each year from 2008 to 2018.

```
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()
```

```
# To import the data from FiveThirtyEight's github page:
new_york_edited <- read_csv(url("https://raw.githubusercontent.com/fivethirtyeight/police-settle
ments/main/new_york_ny/final/new_york_edited.csv"))
```

```
## 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()
## )
```

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

```
# To restrict to the years 2008 to 2018 I used filter():
new_york_2008_to_2018 <- filter(new_york_edited, incident_year >= 2008 & incident_year <= 2018)

# To take the stratified sample:
set.seed(22)
stratified_ny_sample <- new_york_2008_to_2018 %>%
  group_by(incident_year) %>%
  sample_n(100)
```

Geocoding with Google's API

In order to obtain the latitude and longitude from Google's API, this data requires a small amount of cleaning.

- The location variable needs to be converted from integer to character.
- Although Google Maps is happy to accept intersections with an "&" in the middle of the two streets, this does not work in the API. If the "&" is left in the address, the latitudes and longitudes will be based only the addresses up until that symbol. We can solve this by replacing the "&" with the word "and" using the *stringr* package (part of the tidyverse).

```
# Fixing the variable type:
stratified_ny_sample$location <- as.character(stratified_ny_sample$location)

# Replacing the & with and in each address:
stratified_ny_sample$location <- str_replace(stratified_ny_sample$location, "&", "and")
```

Once this cleaning is performed, we can use the command *mutate_geocode()* to pull the location data from Google's API. In order to do this we'll need to have an API key and use *register_google()* to tell Google who we are. We'll use the package ggmap (https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf) to do this.

```
## 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.
```

```
stratified_ny_sample_locations <- mutate_geocode(stratified_ny_sample, location)
```

After inputting your own API key and running this code you will have a sample of 100 police misconduct cases per incident year from 2008 to 2018, along with latitude and longitude for each incident.

```
write_csv(stratified_ny_sample_locations, "locationsstrat2")
```