COVD-19 Data Analysis

Francesca Iova

2025-08-27

1. Load libraries and read in data

```
#Load packages
library(tidyverse)
library(lubridate)

#Note: we did not use these in class but I needed them for my own analysis
library(rvest)
library(usmap)

#Read in COVID-19 data
us.cases <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_

global.cases <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/cs

us.deaths <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_

global.deaths <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/
```

```
#I like to see the data I'm working with first
us.cases
```

```
## # A tibble: 3,342 x 1,154
##         UID iso2  iso3  code3  FIPS Admin2   Province_State Country_Region   Lat
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>          <chr>          <dbl>
##  1 84001001 US    USA     840  1001 Autauga  Alabama        US              32.5
##  2 84001003 US    USA     840  1003 Baldwin  Alabama        US              30.7
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # i 1,145 more variables: Long_ <dbl>, Combined_Key <chr>, `1/22/20` <dbl>,
## #   `1/23/20` <dbl>, `1/24/20` <dbl>, `1/25/20` <dbl>, `1/26/20` <dbl>,
## #   `1/27/20` <dbl>, `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>,
## #   `1/31/20` <dbl>, `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>,
```

```
## #   `2/4/20` <dbl>, `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>,
## #   `2/8/20` <dbl>, `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, ...
```

```
us.deaths
```

```
## # A tibble: 3,342 x 1,155
##         UID iso2  iso3  code3  FIPS Admin2   Province_State Country_Region   Lat
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>          <chr>          <dbl>
##  1 84001001 US    USA     840  1001 Autauga  Alabama        US              32.5
##  2 84001003 US    USA     840  1003 Baldwin  Alabama        US              30.7
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # i 1,146 more variables: Long_ <dbl>, Combined_Key <chr>, Population <dbl>,
## #   `1/22/20` <dbl>, `1/23/20` <dbl>, `1/24/20` <dbl>, `1/25/20` <dbl>,
## #   `1/26/20` <dbl>, `1/27/20` <dbl>, `1/28/20` <dbl>, `1/29/20` <dbl>,
## #   `1/30/20` <dbl>, `1/31/20` <dbl>, `2/1/20` <dbl>, `2/2/20` <dbl>,
## #   `2/3/20` <dbl>, `2/4/20` <dbl>, `2/5/20` <dbl>, `2/6/20` <dbl>,
## #   `2/7/20` <dbl>, `2/8/20` <dbl>, `2/9/20` <dbl>, `2/10/20` <dbl>, ...
```

```
global.cases
```

```
## # A tibble: 289 x 1,147
##    `Province/State`  `Country/Region`   Lat   Long `1/22/20` `1/23/20` `1/24/20`
##    <chr>             <chr>            <dbl>  <dbl>     <dbl>     <dbl>     <dbl>
##  1 <NA>              Afghanistan       33.9  67.7          0         0         0
##  2 <NA>              Albania           41.2  20.2          0         0         0
##  3 <NA>              Algeria           28.0   1.66         0         0         0
##  4 <NA>              Andorra           42.5   1.52         0         0         0
##  5 <NA>              Angola           -11.2  17.9          0         0         0
##  6 <NA>              Antarctica       -71.9  23.3          0         0         0
##  7 <NA>              Antigua and Bar~  17.1 -61.8          0         0         0
##  8 <NA>              Argentina        -38.4 -63.6          0         0         0
##  9 <NA>              Armenia           40.1  45.0          0         0         0
## 10 Australian Capit~ Australia        -35.5 149.           0         0         0
## # i 279 more rows
## # i 1,140 more variables: `1/25/20` <dbl>, `1/26/20` <dbl>, `1/27/20` <dbl>,
## #   `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>, `1/31/20` <dbl>,
## #   `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>, `2/4/20` <dbl>,
## #   `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>, `2/8/20` <dbl>,
## #   `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, `2/12/20` <dbl>,
## #   `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, `2/16/20` <dbl>, ...
```

```
global.deaths
```

```
## # A tibble: 289 x 1,147
```

```
##    `Province/State`  `Country/Region`   Lat   Long `1/22/20` `1/23/20` `1/24/20`
##    <chr>             <chr>            <dbl>  <dbl>     <dbl>     <dbl>     <dbl>
##  1 <NA>              Afghanistan       33.9  67.7          0         0         0
##  2 <NA>              Albania           41.2  20.2          0         0         0
##  3 <NA>              Algeria           28.0   1.66         0         0         0
##  4 <NA>              Andorra           42.5   1.52         0         0         0
##  5 <NA>              Angola           -11.2  17.9          0         0         0
##  6 <NA>              Antarctica       -71.9  23.3          0         0         0
##  7 <NA>              Antigua and Bar~  17.1 -61.8          0         0         0
##  8 <NA>              Argentina        -38.4 -63.6          0         0         0
##  9 <NA>              Armenia           40.1  45.0          0         0         0
## 10 Australian Capit~ Australia        -35.5 149.           0         0         0
## # i 279 more rows
## # i 1,140 more variables: `1/25/20` <dbl>, `1/26/20` <dbl>, `1/27/20` <dbl>,
## #   `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>, `1/31/20` <dbl>,
## #   `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>, `2/4/20` <dbl>,
## #   `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>, `2/8/20` <dbl>,
## #   `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, `2/12/20` <dbl>,
## #   `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, `2/16/20` <dbl>, ...
```

2. Tidy Data

```
#First, we need to pivot global.cases
global.cases <- global.cases %>%
  pivot_longer(cols = -c(`Province/State`, #condense all but these cols from global.deaths (this way is
                         `Country/Region`,#note we need to use backticks here to escape the slash since
                         Lat, Long),
               names_to = "date", #the col names (the date) will go under new col called "date"
               values_to = "cases") %>% #the col values (# of reported deaths) will go into new col "ca
  select(-c(Lat, Long)) #remove unneeded cols

global.cases
```

```
## # A tibble: 330,327 x 4
##    `Province/State` `Country/Region` date    cases
##    <chr>            <chr>            <chr>   <dbl>
##  1 <NA>             Afghanistan      1/22/20     0
##  2 <NA>             Afghanistan      1/23/20     0
##  3 <NA>             Afghanistan      1/24/20     0
##  4 <NA>             Afghanistan      1/25/20     0
##  5 <NA>             Afghanistan      1/26/20     0
##  6 <NA>             Afghanistan      1/27/20     0
##  7 <NA>             Afghanistan      1/28/20     0
##  8 <NA>             Afghanistan      1/29/20     0
##  9 <NA>             Afghanistan      1/30/20     0
## 10 <NA>             Afghanistan      1/31/20     0
## # i 330,317 more rows
```

```
#Now, using the same method we'll pivot global.deaths
global.deaths <- global.deaths %>%
  pivot_longer(cols = -c(`Province/State`,
                         `Country/Region`,Lat, Long),
```

```
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long))

global.deaths
```

```
## # A tibble: 330,327 x 4
##    `Province/State` `Country/Region` date    deaths
##    <chr>            <chr>            <chr>    <dbl>
##  1 <NA>             Afghanistan      1/22/20      0
##  2 <NA>             Afghanistan      1/23/20      0
##  3 <NA>             Afghanistan      1/24/20      0
##  4 <NA>             Afghanistan      1/25/20      0
##  5 <NA>             Afghanistan      1/26/20      0
##  6 <NA>             Afghanistan      1/27/20      0
##  7 <NA>             Afghanistan      1/28/20      0
##  8 <NA>             Afghanistan      1/29/20      0
##  9 <NA>             Afghanistan      1/30/20      0
## 10 <NA>             Afghanistan      1/31/20      0
## # i 330,317 more rows
```

```
#Combining global cases & deaths into "global"
global <- global.cases %>%
  full_join(global.deaths) %>% #join global.cases & global.deaths
  rename(Country_Region = 'Country/Region', #getting rid of the slashes
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date)) %>% #change date from dbl to date type
  filter(cases > 0) #filter out rows with 0 cases

global
```

```
## # A tibble: 306,827 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-02-24     5      0
##  2 <NA>           Afghanistan    2020-02-25     5      0
##  3 <NA>           Afghanistan    2020-02-26     5      0
##  4 <NA>           Afghanistan    2020-02-27     5      0
##  5 <NA>           Afghanistan    2020-02-28     5      0
##  6 <NA>           Afghanistan    2020-02-29     5      0
##  7 <NA>           Afghanistan    2020-03-01     5      0
##  8 <NA>           Afghanistan    2020-03-02     5      0
##  9 <NA>           Afghanistan    2020-03-03     5      0
## 10 <NA>           Afghanistan    2020-03-04     5      0
## # i 306,817 more rows
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:306827      Length:306827      Min.   :2020-01-22   Min.   :     1
##  Class :character   Class :character   1st Qu.:2020-12-12   1st Qu.:  1316
##  Mode  :character   Mode  :character   Median :2021-09-16   Median :  20365
```

```
##                                       Mean   :2021-09-11   Mean   :  1032863
##                                       3rd Qu.:2022-06-15   3rd Qu.:   271281
##                                       Max.   :2023-03-09   Max.   :103802702
##      deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    214
##  Mean   :  14405
##  3rd Qu.:   3665
##  Max.   :1123836
```

```
#Tidy up US cases using pivot_longer
us.cases <- us.cases %>%
  pivot_longer(cols= -(UID:Combined_Key), #Here I'm selecting all cols EXCEPT UID thru Combined_Key
               names_to = "date",
               values_to = "cases") %>%
  select(FIPS:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))

us.cases
```

```
## # A tibble: 3,819,906 x 7
##     FIPS Admin2  Province_State Country_Region Combined_Key     date       cases
##    <dbl> <chr>   <chr>          <chr>          <chr>            <date>     <dbl>
##  1  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-22     0
##  2  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-23     0
##  3  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-24     0
##  4  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-25     0
##  5  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-26     0
##  6  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-27     0
##  7  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-28     0
##  8  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-29     0
##  9  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-30     0
## 10  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-31     0
## # i 3,819,896 more rows
```

```
#Tidy up US deaths using the same method
us.deaths <- us.deaths %>%
  pivot_longer(cols= -(UID:Population), #Here I'm selecting all cols EXCEPT UID thru Combined_Key
               names_to = "date",
               values_to = "deaths") %>%
  select(FIPS:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))

us.deaths
```

```
## # A tibble: 3,819,906 x 8
##     FIPS Admin2 Province_State Country_Region Combined_Key Population date
##    <dbl> <chr>  <chr>          <chr>          <chr>             <dbl> <date>
##  1  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-22
##  2  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-23
```

```
##  3  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-24
##  4  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-25
##  5  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-26
##  6  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-27
##  7  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-28
##  8  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-29
##  9  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-30
## 10  1001 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-31
## # i 3,819,896 more rows
## # i 1 more variable: deaths <dbl>
```

```
#Combining US cases & deaths into "us"
us <- us.cases %>%
  full_join(us.deaths)

us
```

```
## # A tibble: 3,819,906 x 9
##     FIPS Admin2  Province_State Country_Region Combined_Key     date       cases
##    <dbl> <chr>   <chr>          <chr>          <chr>           <date>     <dbl>
##  1  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-22     0
##  2  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-23     0
##  3  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-24     0
##  4  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-25     0
##  5  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-26     0
##  6  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-27     0
##  7  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-28     0
##  8  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-29     0
##  9  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-30     0
## 10  1001 Autauga Alabama        US             Autauga, Alabam~ 2020-01-31     0
## # i 3,819,896 more rows
## # i 2 more variables: Population <dbl>, deaths <dbl>
```

```
#We need to make the global(more complex) dataset mirror the US(simpler) dataset, so we can compare the
global <- global %>%
  unite("Combined_Key", #combine Province_State & Country_region to Combined_Key
        c(Province_State, Country_Region),
        sep = ", ", #they'll be separated by a comma & space
        na.rm = TRUE,
        remove = FALSE)

#And we want to add global population info, which is found in a diff dataset I'll load in here
uid <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2)) #remove unnecessary cols

#Join the uid & global datasets to add a pop col
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>% #left_join to join the datasets on the
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths, Population, Combined_Key)

global
```

```
## # A tibble: 306,827 x 7
```

```
##    Province_State Country_Region date       cases deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341 Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341 Afghanistan
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341 Afghanistan
## # i 306,817 more rows
```

3. Visualize Data

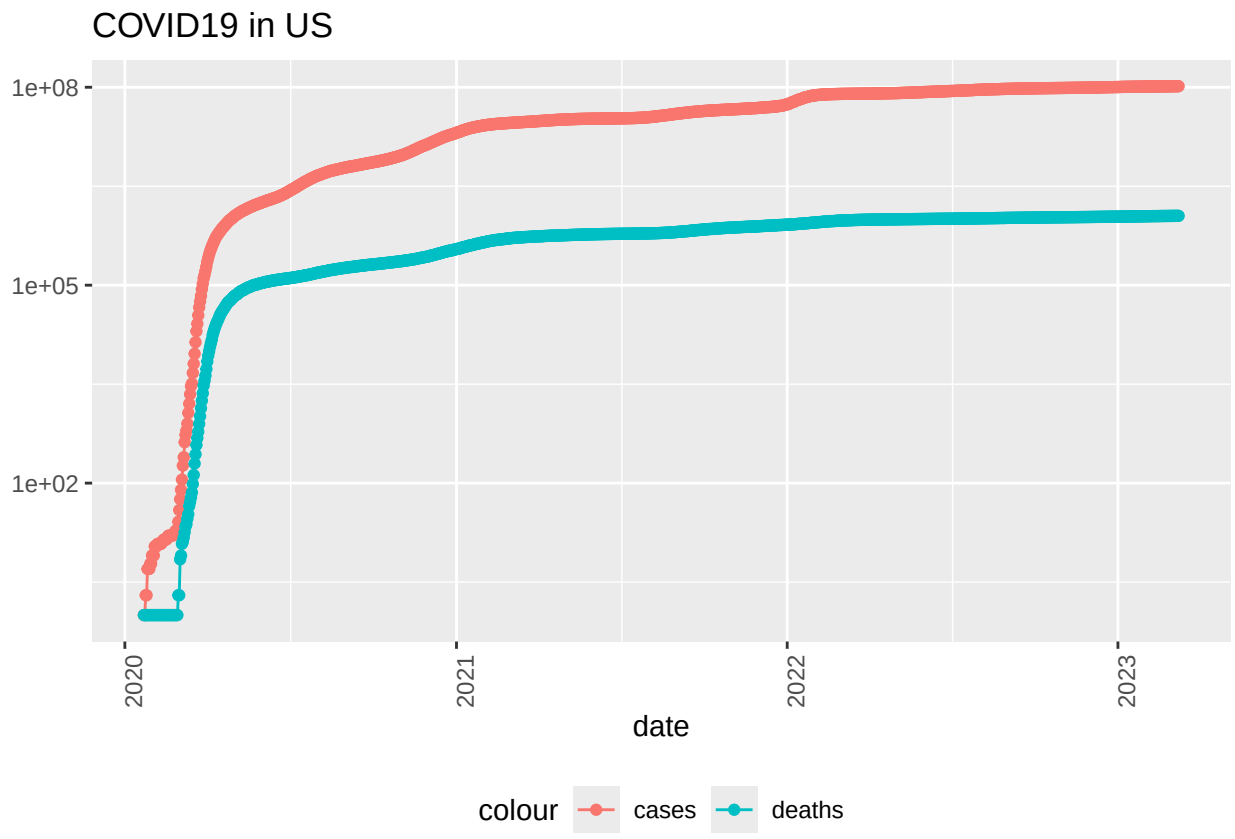
```
us.by.state <- us %>%
  group_by(Province_State, Country_Region, date) %>% #group the dataset by state
  summarize(cases = sum(cases),
            deaths = sum(deaths),
            Population = sum(Population)) %>% #summarize the total cases, deaths, & pop by state
  mutate(deaths_per_mill = deaths*1000000 / Population) %>% #add deaths_per_mill col
  select(Province_State, Country_Region, date, cases, deaths, deaths_per_mill, Population) %>% #select
  ungroup()

us.by.state
```

```
## # A tibble: 66,294 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
##  1 Alabama        US             2020-01-22     0      0               0
##  2 Alabama        US             2020-01-23     0      0               0
##  3 Alabama        US             2020-01-24     0      0               0
##  4 Alabama        US             2020-01-25     0      0               0
##  5 Alabama        US             2020-01-26     0      0               0
##  6 Alabama        US             2020-01-27     0      0               0
##  7 Alabama        US             2020-01-28     0      0               0
##  8 Alabama        US             2020-01-29     0      0               0
##  9 Alabama        US             2020-01-30     0      0               0
## 10 Alabama        US             2020-01-31     0      0               0
## # i 66,284 more rows
## # i 1 more variable: Population <dbl>
```

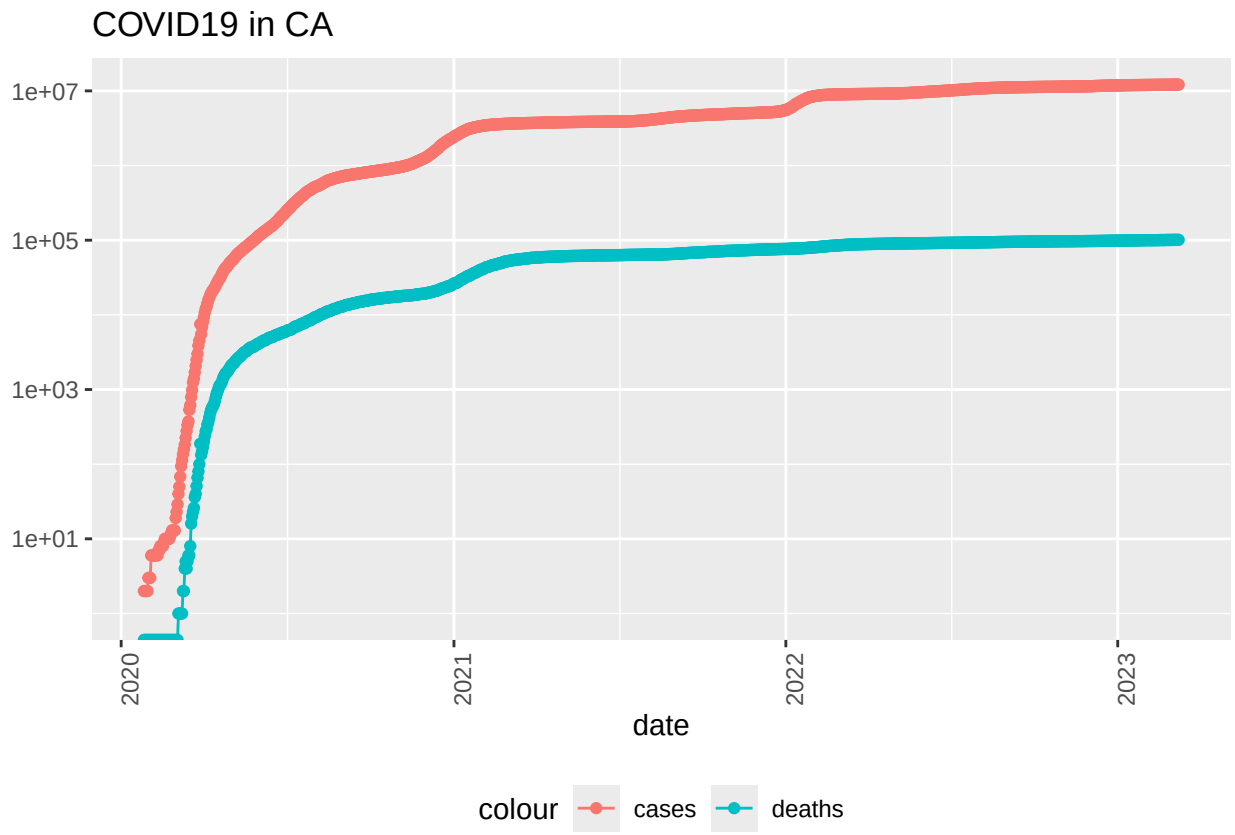
```
#I also want the US totals by date using the same method
us.totals <- us.by.state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()

us.totals
```

```
## # A tibble: 1,143 x 6
##    Country_Region date       cases deaths deaths_per_mill Population
##    <chr>          <date>     <dbl>  <dbl>           <dbl>      <dbl>
##  1 US             2020-01-22     1      1         0.00300  332875137
##  2 US             2020-01-23     1      1         0.00300  332875137
##  3 US             2020-01-24     2      1         0.00300  332875137
##  4 US             2020-01-25     2      1         0.00300  332875137
##  5 US             2020-01-26     5      1         0.00300  332875137
##  6 US             2020-01-27     5      1         0.00300  332875137
##  7 US             2020-01-28     5      1         0.00300  332875137
##  8 US             2020-01-29     6      1         0.00300  332875137
##  9 US             2020-01-30     6      1         0.00300  332875137
## 10 US             2020-01-31     8      1         0.00300  332875137
## # i 1,133 more rows
```

```r
#Let's plot the US totals!
us.totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() + #scale the y axis logarithmically
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```



```r
#I'll look at CA specifically
us.by.state %>%
  filter(Province_State == "California") %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() + #scale the y axis logarithmically
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in CA", y = NULL)
```




```r
#Let's see the date with the most COVID related deaths
max(us.totals$date)
```

```
## [1] "2023-03-09"
```

```r
max(us.totals$deaths)
```

```
## [1] 1123836
```

4. Analyze Data

```
#We'll add new variables conveying the new cases/deaths each day
us.by.state <- us.by.state %>%
  mutate(new_cases = cases - lag(cases),#lag() shifts the time (here that means date) one back (source:
         new_deaths = deaths - lag(deaths))

us.totals <- us.totals %>%
  arrange(date) %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

us.by.state
```

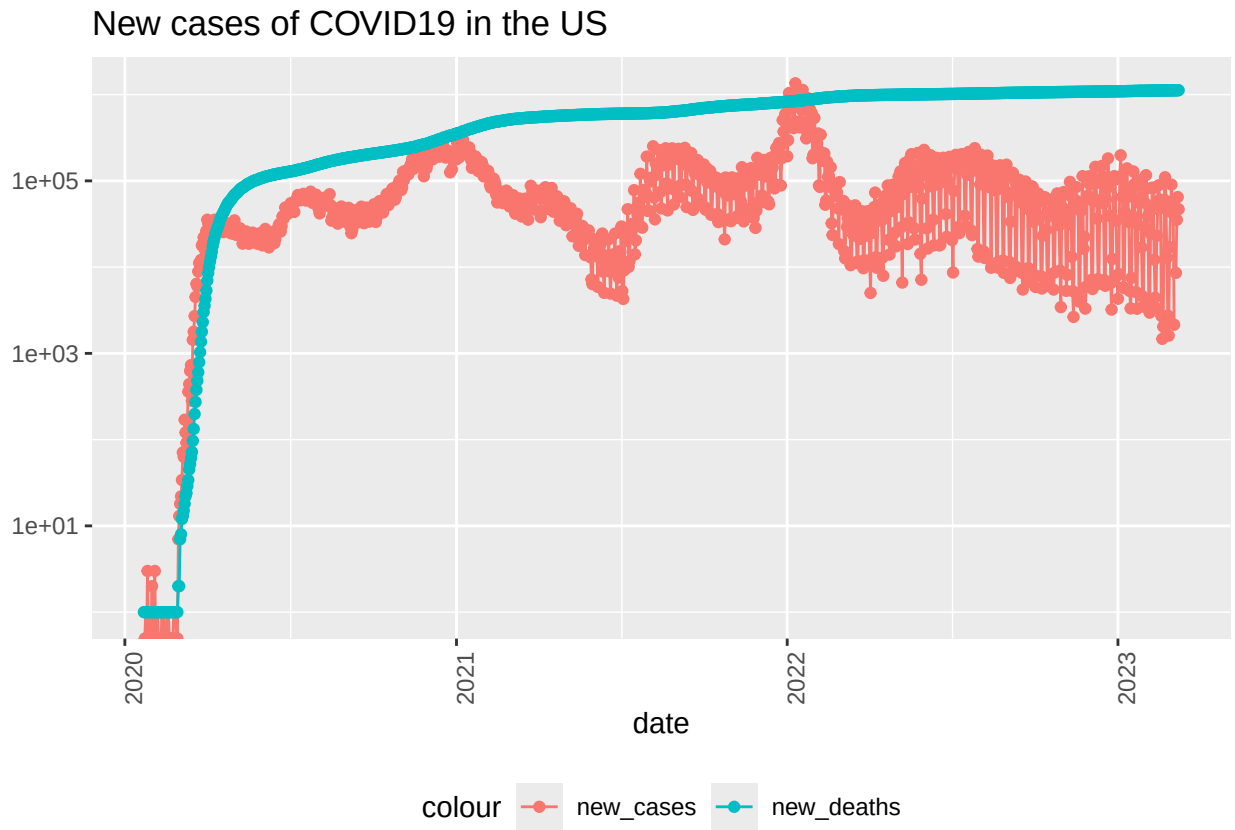
```
## # A tibble: 66,294 x 9
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
##  1 Alabama        US             2020-01-22     0      0               0
##  2 Alabama        US             2020-01-23     0      0               0
##  3 Alabama        US             2020-01-24     0      0               0
##  4 Alabama        US             2020-01-25     0      0               0
##  5 Alabama        US             2020-01-26     0      0               0
##  6 Alabama        US             2020-01-27     0      0               0
##  7 Alabama        US             2020-01-28     0      0               0
##  8 Alabama        US             2020-01-29     0      0               0
##  9 Alabama        US             2020-01-30     0      0               0
## 10 Alabama        US             2020-01-31     0      0               0
## # i 66,284 more rows
## # i 3 more variables: Population <dbl>, new_cases <dbl>, new_deaths <dbl>
```

```
us.totals
```

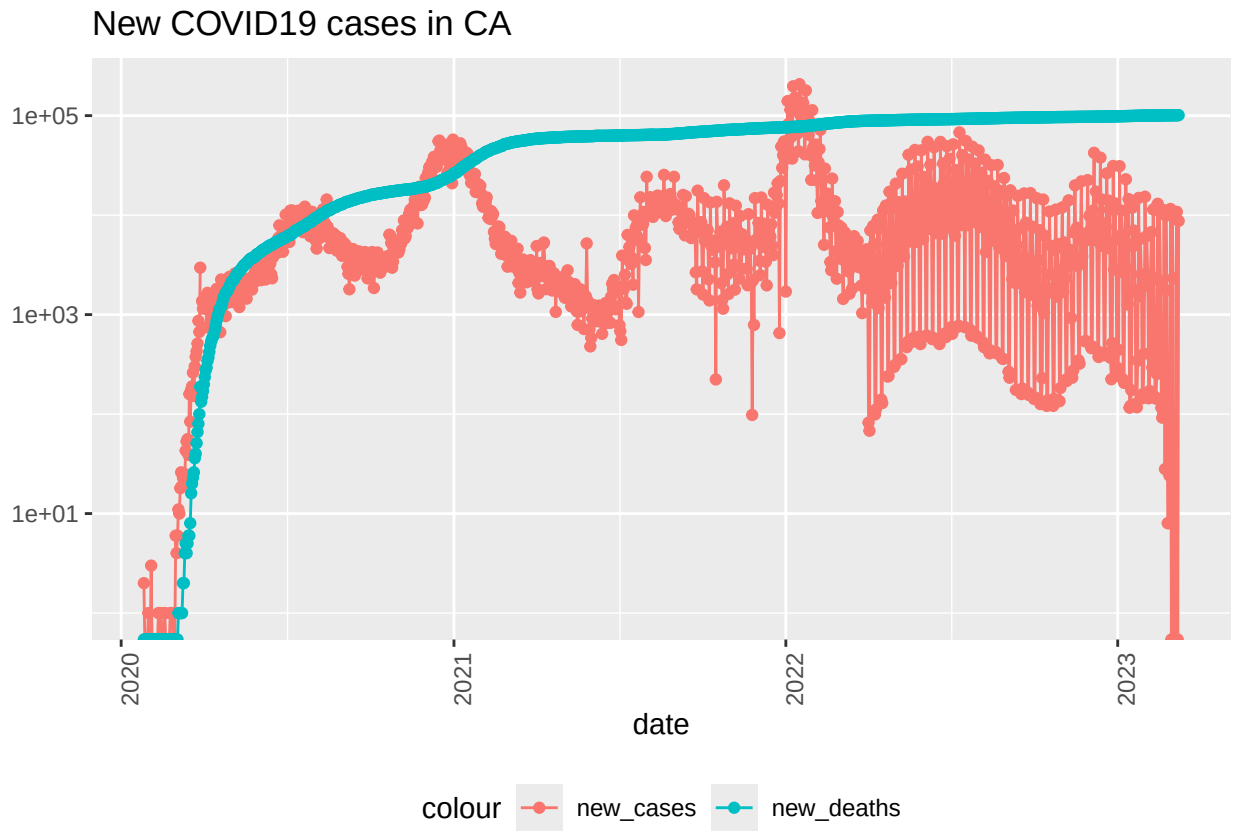
```
## # A tibble: 1,143 x 8
##    Country_Region date       cases deaths deaths_per_mill Population new_cases
##    <chr>          <date>     <dbl>  <dbl>           <dbl>      <dbl>     <dbl>
##  1 US             2020-01-22     1      1         0.00300  332875137        NA
##  2 US             2020-01-23     1      1         0.00300  332875137         0
##  3 US             2020-01-24     2      1         0.00300  332875137         1
##  4 US             2020-01-25     2      1         0.00300  332875137         0
##  5 US             2020-01-26     5      1         0.00300  332875137         3
##  6 US             2020-01-27     5      1         0.00300  332875137         0
##  7 US             2020-01-28     5      1         0.00300  332875137         0
##  8 US             2020-01-29     6      1         0.00300  332875137         1
##  9 US             2020-01-30     6      1         0.00300  332875137         0
## 10 US             2020-01-31     8      1         0.00300  332875137         2
## # i 1,133 more rows
## # i 1 more variable: new_deaths <dbl>
```

```
#Let's graph the new cases across the US
us.totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
```

```
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "New cases of COVID19 in the US", y = NULL)
```




```
#and now CA specifically
us.by.state %>%
  filter(Province_State == "California") %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "New COVID19 cases in CA", y = NULL)
```




```
#Here, I'll add columns calculating deaths per thousand by state
us.state.totals <- us.by.state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths),
            cases = max(cases),
            population = max(Population),
            cases_per_thou = 1000 * cases / population,
            deaths_per_thou = 1000 * deaths / population) %>%
  filter(cases > 0, population > 0)

us.state.totals %>%
  slice_min(deaths_per_thou, n = 10) %>%#this will tell me the 10 states with the least deaths per thou
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State        deaths  cases population
##              <dbl>          <dbl> <chr>                  <dbl>  <dbl>      <dbl>
##  1           0.611           150. American Samoa            34 8.32e3      55641
##  2           0.744           248. Northern Mariana Isl~     41 1.37e4      55144
##  3           1.21            231. Virgin Islands           130 2.48e4     107268
##  4           1.30            269. Hawaii                  1841 3.81e5    1415872
##  5           1.49            245. Vermont                  929 1.53e5     623989
##  6           1.55            293. Puerto Rico             5823 1.10e6    3754939
##  7           1.65            340. Utah                    5298 1.09e6    3205958
##  8           2.01            415. Alaska                  1486 3.08e5     740995
```

```
##  9           2.03            252. District of Columbia    1432 1.78e5     705749
## 10           2.06            253. Washington             15683 1.93e6    7614893
```

```
us.state.totals %>%
  slice_max(deaths_per_thou, n = 10) %>% #and conversely, the 10 states with the most deaths per thousa
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.55           336. Arizona         33102 2443514    7278717
##  2            4.54           326. Oklahoma        17972 1290929    3956971
##  3            4.49           333. Mississippi     13370  990756    2976149
##  4            4.44           359. West Virginia    7960  642760    1792147
##  5            4.32           320. New Mexico       9061  670929    2096829
##  6            4.31           334. Arkansas        13020 1006883    3017804
##  7            4.29           335. Alabama         21032 1644533    4903185
##  8            4.28           368. Tennessee       29263 2515130    6829174
##  9            4.23           307. Michigan        42205 3064125    9986857
## 10            4.06           385. Kentucky        18130 1718471    4467673
```

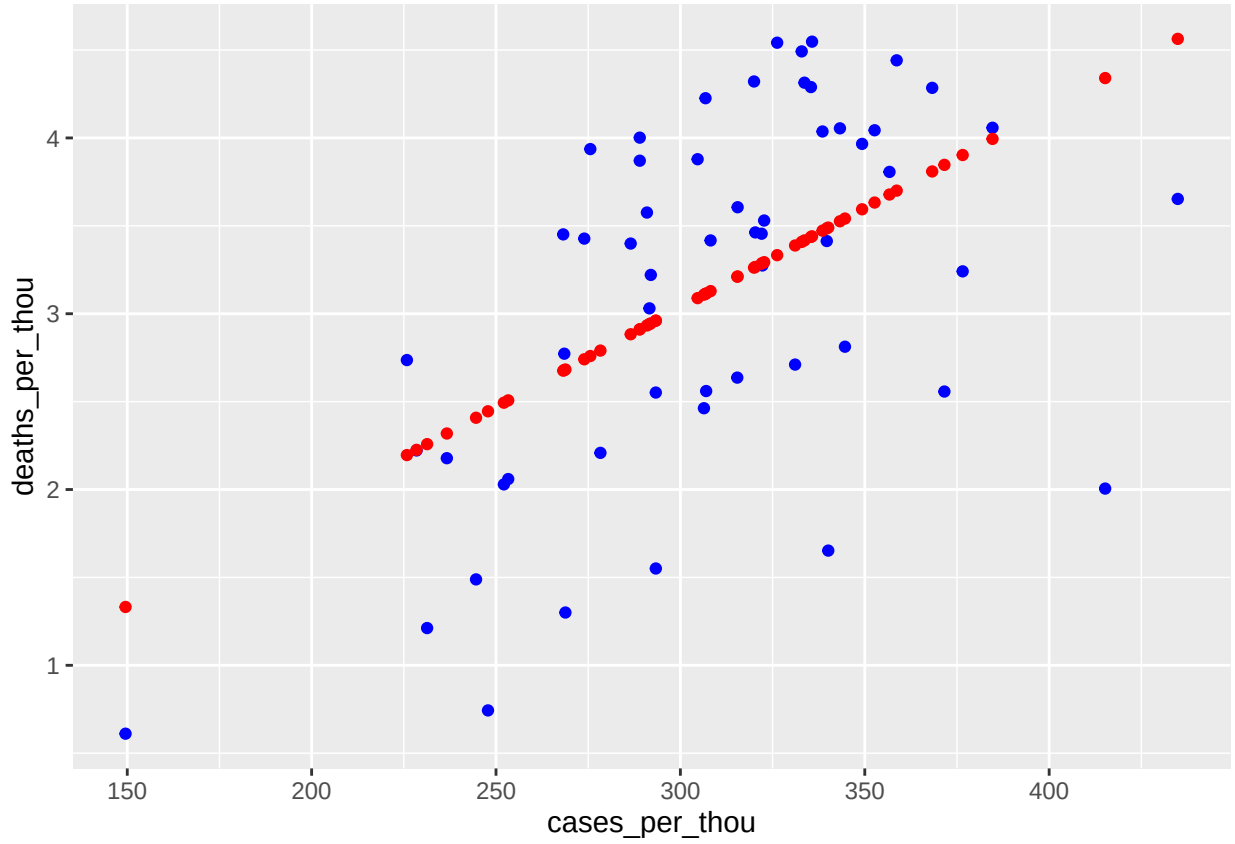
5. Modeling Data

```
#Let's use the linear model to see the deaths per thousand by state as a function of cases per thousand
lmdl <- lm(deaths_per_thou ~ cases_per_thou, data = us.state.totals)

summary(lmdl)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = us.state.totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.36167    0.72480  -0.499     0.62
## cases_per_thou  0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8615 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06
```

```
#Now I'll add the linear model to the state totals data
us.tot.w.preds <- us.state.totals %>%
  mutate(pred = predict(lmdl))
```

```r
#Finally, I want t oplot this relationship

us.tot.w.preds %>%
  ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



6. Bias

My bias comes in further along in this project. It is that I believe the poorest communities are harmed the most in a majority of the cases where our entire nation is subject to some disaster. Of course I believe there will be anomalies, but the trend at large does not escape me, nor anyone else who finds this pattern troubling. Admittedly, the idea that the poorer are worse off in these situations has become somewhat of an assumption to me, one I try to avoid but can't completely hide from. Of course that is only 1 of the 2 factors I will consider in my further analysis, the other being population density. That comes from my BS in biology: this virus is an airborne contagion – the more contact one has with others, the higher their odds of contracting it; and the more densely a community is populated, the harder isolation becomes.

7. My Unique Visual & Analysis

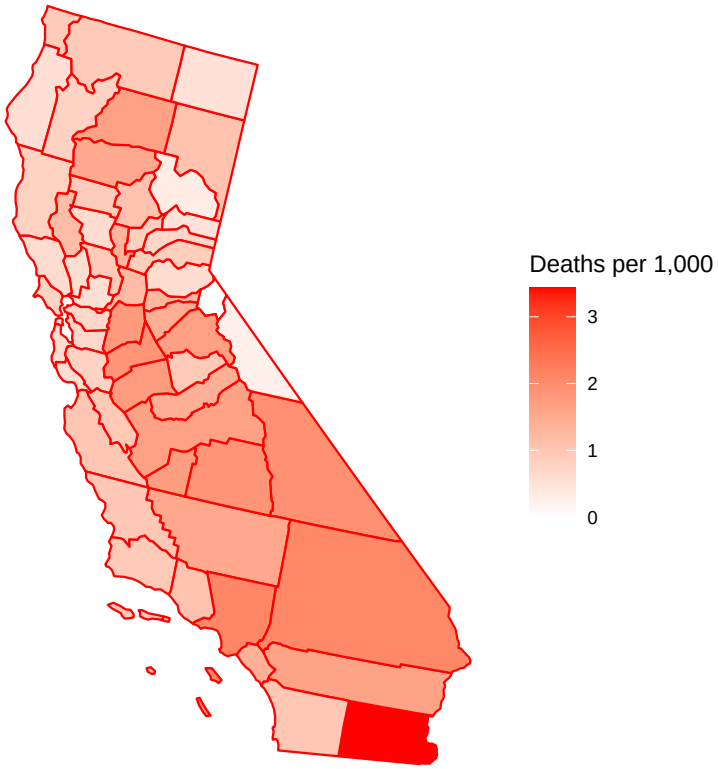
As the above material was led step-by-step in class, I want to leverage my new skills in a different direction here. As a Southern Californian, I'd like to see the spread of COVID-19 by county in my home state, the most populous and 17th most densely populated state in the US. Please note that I'm considering "US Territories" & DC as states here, since the dataset treats them as such, and it also aligns with what I believe

is fair & inclusive treatment of these regions. Unfortunately, these regions are not included in the usmap package, so my currently limited R skills and I will stick with analyzing CA rather than the entirety of the US as I had originally planned.

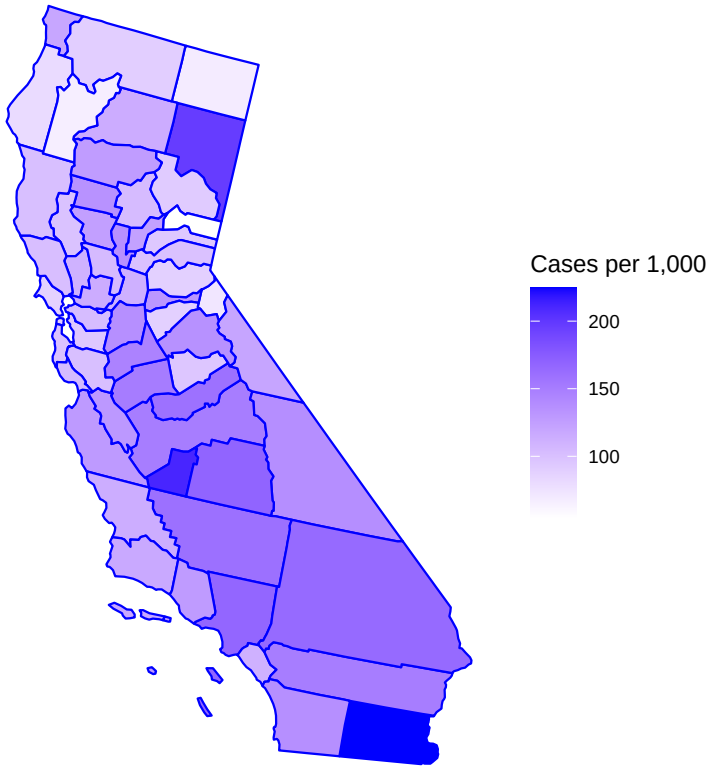
```
#First, using methods learned from these class examples, I will create a CA-specific dataset grouped by
CA <- us %>%
  rename(state = Province_State,
         county = Admin2,
         fips = FIPS) %>% #I renamed these cols to my liking
  filter(state == "California") %>% #single out CA
  group_by(state, county, fips) %>% #group the dataset by county; not I need to use FIPS for the follow
  summarize(total_cases = sum(cases, na.rm = TRUE),
            total_deaths = sum(deaths, na.rm = TRUE),
            pop = sum(Population)) %>% #summarize the total cases, deaths, & pop by county
  mutate(deaths_per_1k = total_deaths*1000 / pop,#add deaths_per_1k col
         cases_per_1k = total_cases * 1000 / pop) %>% #add cases_per_1k col
  filter(county != "Unassigned", county != "Out of CA") %>% #There are a few rows I want to eliminate s
  mutate(fips = paste0(0, fips)) %>% #source for this line of code: <https://stackoverflow.com/question
  select(fips, county, total_cases, total_deaths, deaths_per_1k, cases_per_1k, pop) %>% #select desired
  ungroup()

CA
```

```
## # A tibble: 58 x 8
##    state fips  county total_cases total_deaths deaths_per_1k cases_per_1k    pop
##    <chr> <chr> <chr>        <dbl>        <dbl>         <dbl>        <dbl>  <dbl>
##  1 Cali~ 06001 Alame~   182250215      1357082         0.710         95.4 1.91e9
##  2 Cali~ 06003 Alpine       93941            0         0             72.8 1.29e6
##  3 Cali~ 06005 Amador     5999897        58876         1.30         132.  4.54e7
##  4 Cali~ 06007 Butte     26179854       273706         1.09         104.  2.51e8
##  5 Cali~ 06009 Calav~     4692416        77047         1.47          89.4 5.25e7
##  6 Cali~ 06011 Colusa     3090957        15745         0.639        126.  2.46e7
##  7 Cali~ 06013 Contr~   137155292       930862         0.706        104.  1.32e9
##  8 Cali~ 06015 Del N~     3826227        29352         0.923        120.  3.18e7
##  9 Cali~ 06017 El Do~    19610886       138169         0.627         89.0 2.20e8
## 10 Cali~ 06019 Fresno   172083673      1902474         1.67         151.  1.14e9
## # i 48 more rows
```

```
plot_usmap(regions = "county", include="California", data = CA, values = "deaths_per_1k", color = "red"
  scale_fill_continuous(low = "white", high = "red", name = "Deaths per 1,000") +
  theme(legend.position = "right")
```




```
plot_usmap(regions = "county", include="California", data = CA, values = "cases_per_1k", color = "blue"
  scale_fill_continuous(low = "white", high = "blue", name = "Cases per 1,000") +
  theme(legend.position = "right")
```




```
CA %>% slice_max(cases_per_1k, n = 3)
```

```
## # A tibble: 3 x 8
##   state  fips  county total_cases total_deaths deaths_per_1k cases_per_1k    pop
##   <chr>  <chr> <chr>        <dbl>        <dbl>         <dbl>        <dbl>  <dbl>
## 1 Calif~ 06025 Imper~    46591814       712501          3.44         225. 2.07e8
## 2 Calif~ 06031 Kings     36915826       302652          1.73         211. 1.75e8
## 3 Calif~ 06035 Lassen     6899164        37396          1.07         197. 3.49e7
```

```
CA %>% slice_max(deaths_per_1k, n = 3)
```

```
## # A tibble: 3 x 8
##   state fips  county total_cases total_deaths deaths_per_1k cases_per_1k     pop
##   <chr> <chr> <chr>        <dbl>        <dbl>         <dbl>        <dbl>   <dbl>
## 1 Cali~ 06025 Imper~    46591814       712501          3.44         225. 2.07e 8
## 2 Cali~ 06037 Los A~  1919132962     24114001          2.10         167. 1.15e10
## 3 Cali~ 06071 San B~   410418540      5187328          2.08         165. 2.49e 9
```

Why was Imperial County so disproportionately affected? I was honestly expecting SF, Orange, & LA to top the list of cases and deaths per capita, especially since this is an airborne pathogen and they're the most densely populated counties in all of CA. Especially LA, since it's also the most populous county in the whole US. It's also surprising to see San Bernardino, Kings, and Lassen make the list, since their densities are relatively low.

I've been interested lately on the adverse effects of wealth inequality in the US, so let's start there. I found this dataset from ca.gov, and chose to use the 2022 income data, since my graph of CA earlier in this project showed a spike in 2022.

```
#Load the data
ca.median.income <- read_csv("https://data.ca.gov/dataset/d56fc70f-5566-4030-8854-1ce72c93e100/resource/

ca.median.income
```

```
## # A tibble: 58 x 42
##    County        AMI ALI_1 ALI_2 ALI_3 ALI_4 ALI_5 ALI_6 ALI_7 ALI_8 ELI_1 ELI_2
##    <chr>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Alameda    142800 15000 17100 19250 21400 23100 24800 26550 28250 30000 34300
##  2 Alpine      94900 10000 11400 12850 14250 15400 16550 17650 18800 19100 21800
##  3 Amador      86600  9100 10400 11700 13000 14050 15100 16100 17150 18200 20800
##  4 Butte       85000  8950 10200 11500 12750 13750 14800 15800 16850 16350 18700
##  5 Calaveras   90000  9450 10800 12150 13500 14600 15650 16750 17800 18900 21600
##  6 Colusa      80300  8450  9650 10850 12050 13000 14000 14950 15900 16350 18700
##  7 Contra Co~ 142800 15000 17100 19250 21400 23100 24800 26550 28250 30000 34300
##  8 Del Norte   80300  8450  9650 10850 12050 13000 14000 14950 15900 16350 18700
##  9 El Dorado  102200 10750 12300 13800 15350 16600 17800 19050 20250 21300 24350
## 10 Fresno      80300  8450  9650 10850 12050 13000 14000 14950 15900 16350 18700
## # i 48 more rows
## # i 30 more variables: ELI_3 <dbl>, ELI_4 <dbl>, ELI_5 <dbl>, ELI_6 <dbl>,
## #   ELI_7 <dbl>, ELI_8 <dbl>, VLI_1 <dbl>, VLI_2 <dbl>, VLI_3 <dbl>,
## #   VLI_4 <dbl>, VLI_5 <dbl>, VLI_6 <dbl>, VLI_7 <dbl>, VLI_8 <dbl>,
## #   LI_1 <dbl>, LI_2 <dbl>, LI_3 <dbl>, LI_4 <dbl>, LI_5 <dbl>, LI_6 <dbl>,
## #   LI_7 <dbl>, LI_8 <dbl>, MOD_1 <dbl>, MOD_2 <dbl>, MOD_3 <dbl>, MOD_4 <dbl>,
## #   MOD_5 <dbl>, MOD_6 <dbl>, MOD_7 <dbl>, MOD_8 <dbl>
```

After reading the dataset's dictionary https://data.ca.gov/dataset/income-limits-by-county/resource/a25962fc-8bdf-484e-afe2-73def7d01b4d, I'll only be keeping the column I'm seeking to work with (AMI)

```
#I'll eliminate the cols I'm not working with, and then join it to my CA data
ca.median.income <- ca.median.income %>%
  rename(county = County,
         median_income = AMI) %>%
  select(county, median_income)

CA <- CA %>%
  left_join(ca.median.income, by = "county")

CA
```

```
## # A tibble: 58 x 9
##    state fips  county total_cases total_deaths deaths_per_1k cases_per_1k    pop
##    <chr> <chr> <chr>        <dbl>        <dbl>         <dbl>        <dbl>  <dbl>
##  1 Cali~ 06001 Alame~   182250215      1357082         0.710         95.4 1.91e9
##  2 Cali~ 06003 Alpine       93941            0         0             72.8 1.29e6
##  3 Cali~ 06005 Amador     5999897        58876         1.30         132.  4.54e7
##  4 Cali~ 06007 Butte     26179854       273706         1.09         104.  2.51e8
##  5 Cali~ 06009 Calav~     4692416        77047         1.47          89.4 5.25e7
##  6 Cali~ 06011 Colusa     3090957        15745         0.639        126.  2.46e7
```

```
##  7 Cali~ 06013 Contr~   137155292       930862         0.706        104.  1.32e9
##  8 Cali~ 06015 Del N~     3826227        29352         0.923        120.  3.18e7
##  9 Cali~ 06017 El Do~    19610886       138169         0.627         89.0 2.20e8
## 10 Cali~ 06019 Fresno   172083673      1902474         1.67         151.  1.14e9
## # i 48 more rows
## # i 1 more variable: median_income <dbl>
```

I also think population density for an airborne pathogen is too big a factor to ignore, so I'll add that to my CA dataset. However, I don't want to introduce a dataset with new/possibly conflicting population values, so I'll load in the area of each county instead, and calculate density myself.

```
#Load in data
ca.county.area <- read_csv("https://cecgis-caenergy.opendata.arcgis.com/api/download/v1/items/ce721c35a

ca.county.area
```

```
## # A tibble: 58 x 8
##    OBJECTID NAME               STATE_NAME STATE_FIPS CNTY_FIPS FIPS  Shape__Area
##       <dbl> <chr>              <chr>      <chr>      <chr>     <chr>       <dbl>
##  1        1 Alameda County     California 06         001       06001     3.08e 9
##  2        2 Alpine County      California 06         003       06003     3.16e 9
##  3        3 Amador County      California 06         005       06005     2.56e 9
##  4        4 Butte County       California 06         007       06007     7.34e 9
##  5        5 Calaveras County   California 06         009       06009     4.36e 9
##  6        6 Colusa County      California 06         011       06011     4.99e 9
##  7        7 Contra Costa Coun~ California 06         013       06013     3.08e 9
##  8        8 Del Norte County   California 06         015       06015     4.72e 9
##  9        9 El Dorado County   California 06         017       06017     7.62e 9
## 10       10 Fresno County      California 06         019       06019     2.43e10
## # i 48 more rows
## # i 1 more variable: Shape__Length <dbl>
```

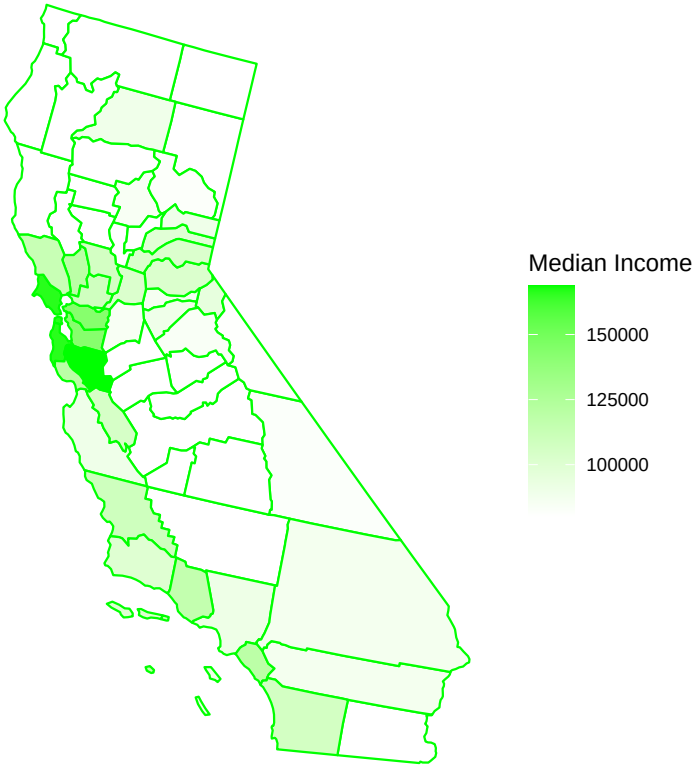
Note that the units of measure are m^2, and it looks like the data might be a bit distorted. Since I'm only using this data to make rough, relative comparisons, it will serve my purposes here.

```
#tidying up the data to only keep the area, then joining and creating a new column calculating pop dens
ca.county.area <- ca.county.area %>%
  mutate(area = Shape__Area / 1000) %>% #converting m^2 to km^2
  rename(fips = FIPS) %>%
  select(fips, area)

CA <- CA %>%
  left_join(ca.county.area, by = "fips") %>%
  mutate(pop_density = pop / area) %>%
  select(-c(total_cases, total_deaths, area))

CA
```

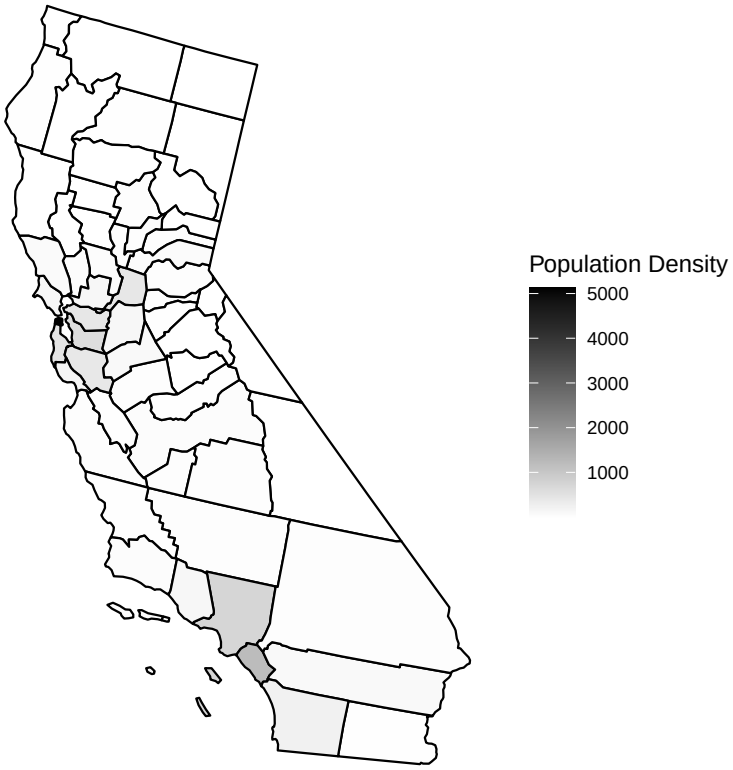
```
## # A tibble: 58 x 8
##    state      fips  county       deaths_per_1k cases_per_1k    pop median_income
##    <chr>      <chr> <chr>                <dbl>        <dbl>  <dbl>         <dbl>
##  1 California 06001 Alameda              0.710         95.4 1.91e9        142800
##  2 California 06003 Alpine               0             72.8 1.29e6         94900
```

```
##  3 California 06005 Amador              1.30         132.  4.54e7         86600
##  4 California 06007 Butte               1.09         104.  2.51e8         85000
##  5 California 06009 Calaveras           1.47          89.4 5.25e7         90000
##  6 California 06011 Colusa              0.639        126.  2.46e7         80300
##  7 California 06013 Contra Costa        0.706        104.  1.32e9        142800
##  8 California 06015 Del Norte           0.923        120.  3.18e7         80300
##  9 California 06017 El Dorado           0.627         89.0 2.20e8        102200
## 10 California 06019 Fresno              1.67         151.  1.14e9         80300
## # i 48 more rows
## # i 1 more variable: pop_density <dbl>
```

```
plot_usmap(regions = "county", include="California", data = CA, values = "median_income", color = "green
  scale_fill_continuous(low = "white", high = "green", name = "Median Income") +
  theme(legend.position = "right")
```




```
plot_usmap(regions = "county", include="California", data = CA, values = "pop_density", color = "black"
  scale_fill_continuous(low = "white", high = "black", name = "Population Density") +
  theme(legend.position = "right")
```



Interesting graphic, let's zoom in on those numbers

```
CA %>% slice_max(pop_density, n = 10)
```

```
## # A tibble: 10 x 8
##    state      fips  county      deaths_per_1k cases_per_1k     pop median_income
##    <chr>      <chr> <chr>               <dbl>        <dbl>   <dbl>         <dbl>
##  1 California 06075 San Franci~         0.627         90.4 1.01e 9        166000
##  2 California 06059 Orange              1.40         112.  3.63e 9        119100
##  3 California 06037 Los Angeles         2.10         167.  1.15e10         91100
##  4 California 06001 Alameda             0.710         95.4 1.91e 9        142800
##  5 California 06081 San Mateo           0.625         99.5 8.76e 8        166000
##  6 California 06013 Contra Cos~         0.706        104.  1.32e 9        142800
##  7 California 06067 Sacramento          1.21         113.  1.77e 9        102200
##  8 California 06085 Santa Clara         0.804        102.  2.20e 9        168500
##  9 California 06073 San Diego           1.00         137.  3.82e 9        106900
## 10 California 06087 Santa Cruz          0.613        105.  3.12e 8        119300
## # i 1 more variable: pop_density <dbl>
```

```
CA %>% slice_max(median_income, n = 10)
```

```
## # A tibble: 10 x 8
##    state      fips  county       deaths_per_1k cases_per_1k    pop median_income
##    <chr>      <chr> <chr>                <dbl>        <dbl>  <dbl>         <dbl>
##  1 California 06085 Santa Clara          0.804        102.  2.20e9        168500
```

```
##  2 California 06041 Marin               0.773         88.6 2.96e8        166000
##  3 California 06075 San Francis~        0.627         90.4 1.01e9        166000
##  4 California 06081 San Mateo           0.625         99.5 8.76e8        166000
##  5 California 06001 Alameda             0.710         95.4 1.91e9        142800
##  6 California 06013 Contra Costa        0.706        104.  1.32e9        142800
##  7 California 06055 Napa                0.615        111.  1.57e8        119400
##  8 California 06087 Santa Cruz          0.613        105.  3.12e8        119300
##  9 California 06059 Orange              1.40         112.  3.63e9        119100
## 10 California 06111 Ventura             1.08         128.  9.67e8        115400
## # i 1 more variable: pop_density <dbl>
```

```
CA %>% slice_min(pop_density, n = 10)
```

```
## # A tibble: 10 x 8
##    state      fips  county   deaths_per_1k cases_per_1k      pop median_income
##    <chr>      <chr> <chr>            <dbl>        <dbl>    <dbl>         <dbl>
##  1 California 06003 Alpine           0             72.8  1290447         94900
##  2 California 06027 Inyo             1.95         138.  20618577         82700
##  3 California 06049 Modoc            0.536         69.3 10105263         80300
##  4 California 06091 Sierra           0.496         55.1  3434715         90000
##  5 California 06105 Trinity          0.802         67.0 14041755         80300
##  6 California 06051 Mono             0.266        121.  16509492         81200
##  7 California 06035 Lassen           1.07         197.  34944939         80300
##  8 California 06093 Siskiyou         0.933         90.3 49765077         80300
##  9 California 06063 Plumas           0.348         92.5 21496401         82400
## 10 California 06043 Mariposa         0.965         96.3 19663029         80300
## # i 1 more variable: pop_density <dbl>
```

```
CA %>% slice_min(median_income, n = 21)
```

```
## # A tibble: 22 x 8
##    state      fips  county    deaths_per_1k cases_per_1k       pop median_income
##    <chr>      <chr> <chr>             <dbl>        <dbl>     <dbl>         <dbl>
##  1 California 06011 Colusa            0.639        126.     2.46e7         80300
##  2 California 06015 Del Norte         0.923        120.     3.18e7         80300
##  3 California 06019 Fresno            1.67         151.     1.14e9         80300
##  4 California 06021 Glenn             0.915        135.     3.25e7         80300
##  5 California 06023 Humboldt          0.573         81.4    1.55e8         80300
##  6 California 06025 Imperial          3.44         225.     2.07e8         80300
##  7 California 06029 Kern              1.55         160.     1.03e9         80300
##  8 California 06031 Kings             1.73         211.     1.75e8         80300
##  9 California 06033 Lake              1.18          99.5    7.36e7         80300
## 10 California 06035 Lassen            1.07         197.     3.49e7         80300
## # i 12 more rows
## # i 1 more variable: pop_density <dbl>
```

Well, there you have it: both population density and wealth inequality play a part – but it looks like the richer the county, the more the population density risk is mitigated. I also think it's a sad fact that the estimated median income of 21 counties is the same and makes up the lowest bound at a median income of 80,300 USD. A few of the most densely populated counties like SF, Alameda, & San Mateo make about double the income of the poorest counties like Imperial, Kings, and Lassen, which were hit the hardest per capita. A great demonstration of this wealth disparity is LA, which has a median income of 91,100 USD, and also the 3rd highest pop density. Here, COVID also hit hard, claiming the most lives total, and had the 2nd most deaths per capita.