Covid19_Final

2023-08-14

Work from class

All of the following code was taken from the class lectures, skip to the “New Analysis” section to see my visualizations and analysis.

```
library(ggplot2)
library(lessR)
library(lubridate)
library("tidyverse")
library(readr)
```

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("time_series_covid19_confirmed_US.csv",
                "time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_US.csv",
                "time_series_covid19_deaths_global.csv",
                "time_series_covid19_recovered_global.csv")
urls <- str_c(url_in, file_names)
```

```
US_cases <- read_csv(urls[1])
global_cases <- read_csv(urls[2])
US_deaths <- read_csv(urls[3])
global_deaths <- read_csv(urls[4])
global_recovered <- read_csv(urls[5])
```

```
global_cases <- global_cases %>%
    pivot_longer(cols=-c("Province/State", "Country/Region", "Lat", "Long"),
    names_to="date", values_to="cases") %>% select(-c(Lat,Long))
```

```
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
## # ... with 330,317 more rows
```

```
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
## # ... with 330,317 more rows
```

```
global_recovered <- global_recovered %>%
    pivot_longer(cols=-c("Province/State", "Country/Region", "Lat", "Long"),
    names_to="date", values_to="recovered") %>% select(-c(Lat,Long))
```

```
global_recovered
```

```
## # A tibble: 313,182 x 4
##    ‘Province/State‘ ‘Country/Region‘ date    recovered
##    <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
## # ... with 313,172 more rows
```

```
global <- global_cases %>%
    full_join(global_recovered) %>%
    full_join(global_deaths) %>%
    rename(Country_Region="Country/Region", Province_State="Province/State") %>%
    mutate(date=mdy(date))
```

```
## Joining, by = c("Province/State", "Country/Region", "date")
## Joining, by = c("Province/State", "Country/Region", "date")
```

```
global
```

```
## # A tibble: 331,470 x 6
##    Province_State Country_Region date       cases recovered deaths
##    <chr>          <chr>          <date>     <dbl>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-01-22     0         0      0
##  2 <NA>           Afghanistan    2020-01-23     0         0      0
##  3 <NA>           Afghanistan    2020-01-24     0         0      0
##  4 <NA>           Afghanistan    2020-01-25     0         0      0
##  5 <NA>           Afghanistan    2020-01-26     0         0      0
##  6 <NA>           Afghanistan    2020-01-27     0         0      0
##  7 <NA>           Afghanistan    2020-01-28     0         0      0
##  8 <NA>           Afghanistan    2020-01-29     0         0      0
##  9 <NA>           Afghanistan    2020-01-30     0         0      0
## 10 <NA>           Afghanistan    2020-01-31     0         0      0
## # ... with 331,460 more rows
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases          
##  Length:331470      Length:331470      Min.   :2020-01-22   Min.   :        0  
##  Class :character   Class :character   1st Qu.:2020-11-02   1st Qu.:      680  
##  Mode  :character   Mode  :character   Median :2021-08-15   Median :    14429  
##                                        Mean   :2021-08-15   Mean   :   959384  
##                                        3rd Qu.:2022-05-28   3rd Qu.:   228517  
##                                        Max.   :2023-03-09   Max.   :103802702  
##                                                             NA's   :1143       
##    recovered             deaths       
##  Min.   :      -1   Min.   :      0  
##  1st Qu.:       0   1st Qu.:      3  
##  Median :       0   Median :    150  
##  Mean   :   75009   Mean   :  13380  
##  3rd Qu.:     934   3rd Qu.:   3032  
##  Max.   :30974748   Max.   :1123836  
##  NA's   :18288      NA's   :1143     
```

```
global <- global %>% filter(cases > 0)
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  
##                                                                                
##    recovered             deaths       
##  Min.   :      -1   Min.   :      0  
##  1st Qu.:       0   1st Qu.:      7
##  Median :       0   Median :    214
##  Mean   :   79865   Mean   :  14405
##  3rd Qu.:    1235   3rd Qu.:   3665
##  Max.   :30974748   Max.   :1123836
##  NA’s   :16010
```

```
US_cases <- US_cases %>%
    pivot_longer(cols=-(UID:Combined_Key),
    names_to="date", values_to="cases") %>%
    select(Admin2:cases) %>%
    mutate(date=mdy(date)) %>%
    select(-c(Lat, Long_))
```

```
US_deaths <- US_deaths %>%
    pivot_longer(cols=-(UID:Combined_Key),
    names_to="date", values_to="deaths") %>%
    select(Admin2:deaths) %>%
    mutate(date=mdy(date)) %>%
    select(-c(Lat, Long_))
```

```
US_cases
```

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

```
US <- US_cases %>% full_join(US_deaths)
```

```
## Joining, by = c("Admin2", "Province_State", "Country_Region", "Combined_Key",
## "date")
```

```
global <- global %>%
    unite("Combined_Key",
          c(Province_State, Country_Region),
          sep=", ",
          na.rm=TRUE,
          remove=FALSE)
```

```
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/
uid <- read_csv(uid_lookup_url) %>% select(-c(Lat, Long_, code3, iso2, iso3, Admin2))
```

```
## Rows: 4321 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
global <- global %>%
    left_join(uid, by=c("Province_State", "Country_Region")) %>%
    select(-c(UID, FIPS)) %>%
    select(Province_State, Country_Region, date, cases, deaths, Population)
```

```
global
```

```
## # A tibble: 306,827 x 6
##    Province_State Country_Region date       cases deaths Population
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341
## # ... with 306,817 more rows
```

```
US <- US %>%
    left_join(uid, by=c("Province_State", "Country_Region", "Combined_Key")) %>%
    select(-c(UID, FIPS)) %>%
    select(Province_State, Country_Region, date, cases, deaths, Population, Combined_Key)
```

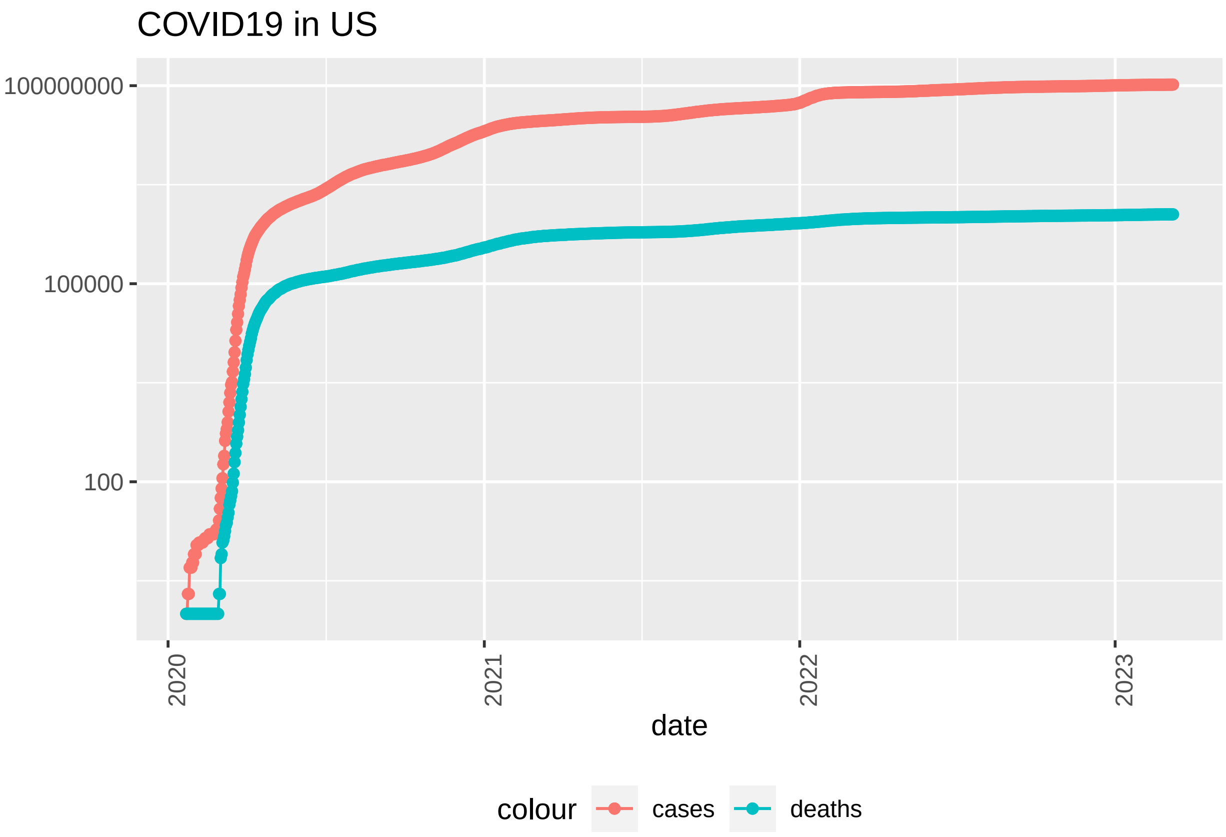
```
US_by_state <- US %>%
    group_by(Province_State, Country_Region, date) %>%
    summarize(cases=sum(cases), deaths=sum(deaths), Population=sum(Population, na.rm=TRUE)) %>%
    mutate(deaths_per_mill=deaths*1000000/Population) %>%
    select(Province_State, Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

```
## ‘summarise()‘ has grouped output by ’Province_State’, ’Country_Region’. You can
## override using the ‘.groups‘ argument.
```

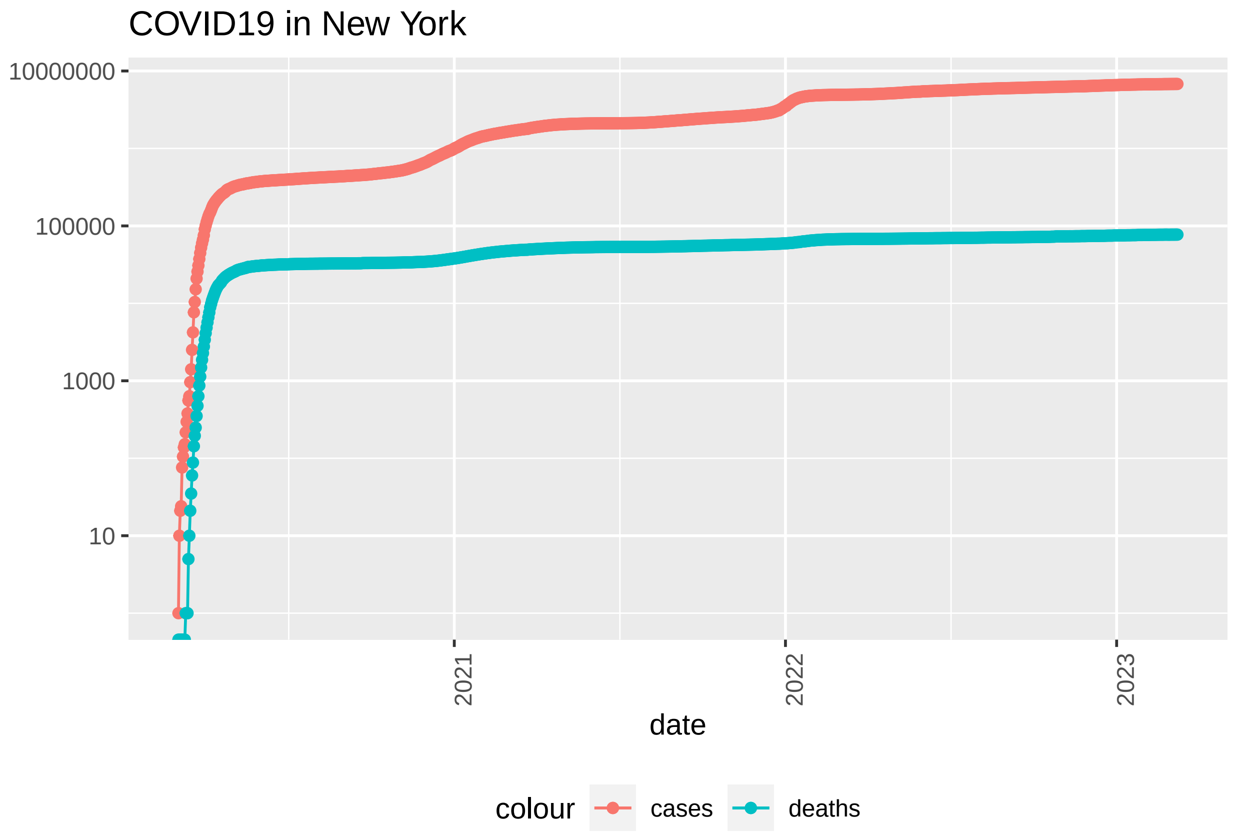
```
US_totals <- US_by_state %>% group_by(Country_Region, date) %>%
    summarize(cases=sum(cases), deaths=sum(deaths), Population=sum(Population)) %>%
    mutate(deaths_per_mill=deaths*1000000/Population) %>%
    select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
    ungroup()
```

```
## 'summarise()' has grouped output by 'Country_Region'. You can override using
## the '.groups' argument.
```

```
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() +
    theme(legend.position="bottom", axis.text.x=element_text(angle=90)) +
    labs(title="COVID19 in US", y=NULL)
```

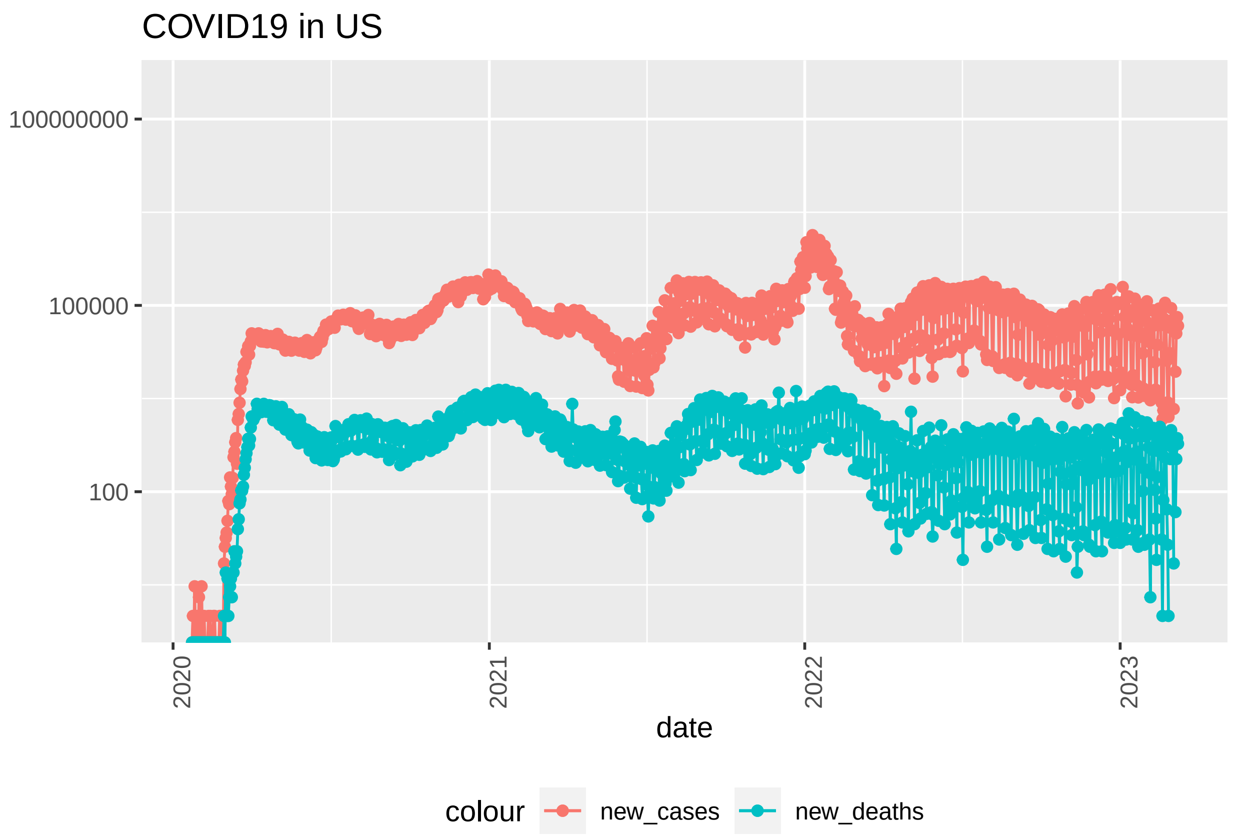


```
state <- "New York"
US_by_state %>%
    filter(Province_State==state) %>%
    filter(cases > 0) %>%
    ggplot(aes(x=date, y=cases)) +
    geom_line(aes(color="cases")) +
    geom_point(aes(color="cases")) +
    geom_line(aes(y=deaths, color="deaths")) +
    geom_point(aes(y=deaths, color="deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom", axis.text.x=element_text(angle=90)) +
    labs(title=str_c("COVID19 in ", state), y=NULL)
```

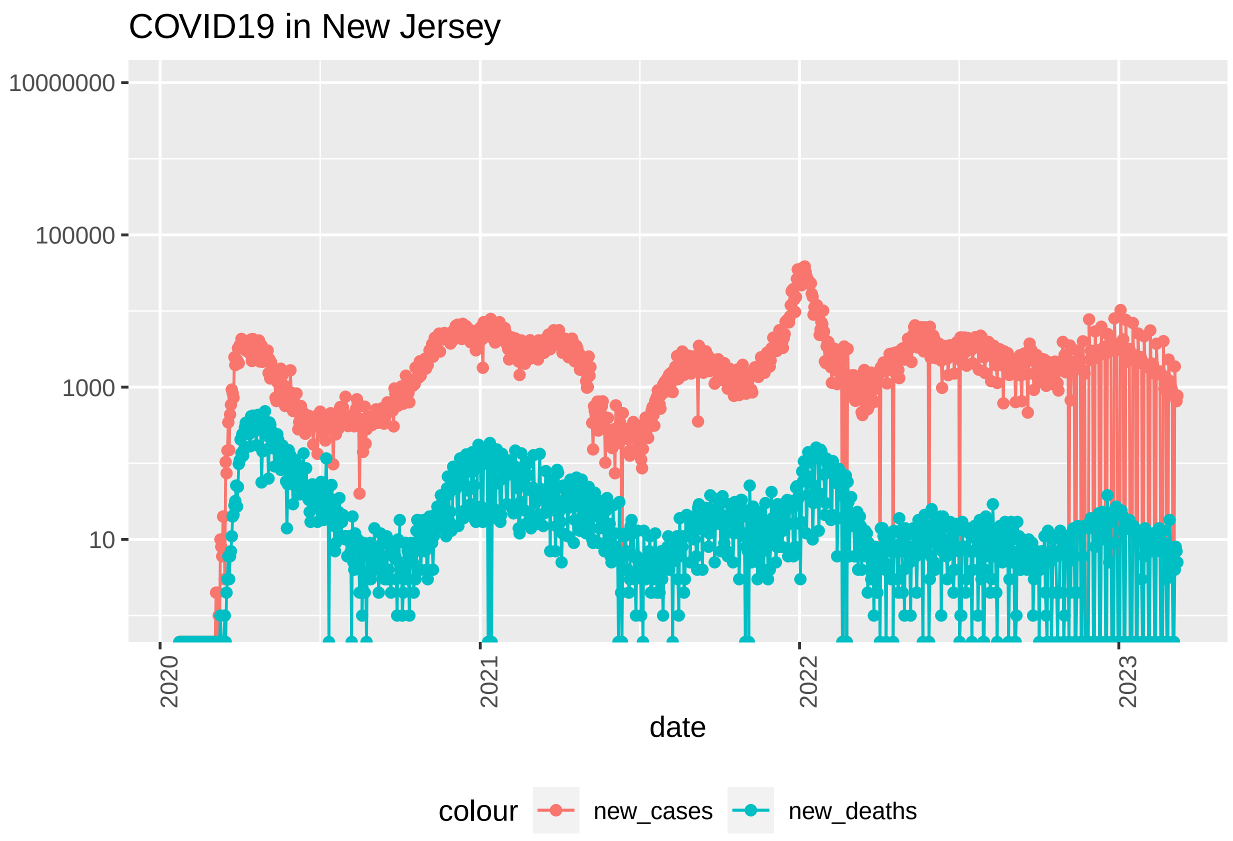


```
US_by_state <- US_by_state %>%
    mutate(new_cases=cases-lag(cases),
           new_deaths=deaths-lag(deaths))
US_totals <- US_totals %>%
    mutate(new_cases=cases-lag(cases),
           new_deaths=deaths-lag(deaths))
```

```
US_totals %>%
    ggplot(aes(x=date, y=new_cases)) +
    geom_line(aes(color="new_cases")) +
    geom_point(aes(color="new_cases")) +
    geom_line(aes(y=new_deaths, color="new_deaths")) +
    geom_point(aes(y=new_deaths, color="new_deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom", axis.text.x=element_text(angle=90)) +
    labs(title="COVID19 in US", y=NULL)
```



```r
state <- "New Jersey"
US_by_state %>%
    filter(Province_State==state) %>%
    ggplot(aes(x=date, y=new_cases)) +
    geom_line(aes(color="new_cases")) +
    geom_point(aes(color="new_cases")) +
    geom_line(aes(y=new_deaths, color="new_deaths")) +
    geom_point(aes(y=new_deaths, color="new_deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom", axis.text.x=element_text(angle=90)) +
    labs(title=str_c("COVID19 in ", state), y=NULL)
```



```
US_state_totals <- US_by_state %>%
    group_by(Province_State) %>%
    summarize(deaths=deaths, cases=cases,
              population=max(Population),
              cases_per_thou=1000*cases/population,
              deaths_per_thou=1000*deaths/population) %>%
    filter(cases > 0)
```

```
## ‘summarise()‘ has grouped output by ’Province_State’. You can override using
## the ‘.groups‘ argument.
```

```
US_state_totals
```

```
## # A tibble: 63,216 x 6
## # Groups:   Province_State [58]
##    Province_State deaths cases population cases_per_thou deaths_per_thou
##    <chr>           <dbl> <dbl>      <dbl>          <dbl>           <dbl>
##  1 Alabama             0     3    4903185       0.000612               0
##  2 Alabama             0     4    4903185       0.000816               0
##  3 Alabama             0     8    4903185       0.00163                0
##  4 Alabama             0    15    4903185       0.00306                0
##  5 Alabama             0    28    4903185       0.00571                0
##  6 Alabama             0    36    4903185       0.00734                0
##  7 Alabama             0    51    4903185       0.0104                 0
##  8 Alabama             0    61    4903185       0.0124                 0
##  9 Alabama              0    88    4903185       0.0179                   0
## 10 Alabama              0   115    4903185       0.0235                   0
## # ... with 63,206 more rows
```

```
US_state_totals_no_nan <- US_state_totals
US_state_totals_no_nan[is.na(US_state_totals_no_nan) | US_state_totals_no_nan == "Inf"] <- NA
mod <- lm(deaths_per_thou ~ cases_per_thou, data=US_state_totals_no_nan)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals_no_nan)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3312 -0.3965 -0.0424  0.4800  1.4893
##
## Coefficients:
##                  Estimate Std. Error t value            Pr(>|t|)
## (Intercept)    0.43305133 0.00434702   99.62 <0.0000000000000002 ***
## cases_per_thou 0.00920684 0.00002297  400.76 <0.0000000000000002 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.6665 on 59948 degrees of freedom
##   (3266 observations deleted due to missingness)
## Multiple R-squared:  0.7282, Adjusted R-squared:  0.7282
## F-statistic: 1.606e+05 on 1 and 59948 DF,  p-value: < 0.00000000000000022
```

New Analysis

Considering how much the US datasets were transformed I will mostly be looking at the global dataset. I will be applying similar transformations to the global dataset as we did for the US in class. Then, I will do an analysis on the cases and deaths for every country.

First, I will calculate cases per thousand and deaths per thousand for each country.

```
global_totals <- global %>%
    group_by(Country_Region) %>%
    summarize(deaths=deaths, cases=cases, date=date,
              population=max(Population),
              cases_per_thou=1000*cases/population,
              deaths_per_thou=1000*deaths/population) %>%
    filter(cases > 0)
```

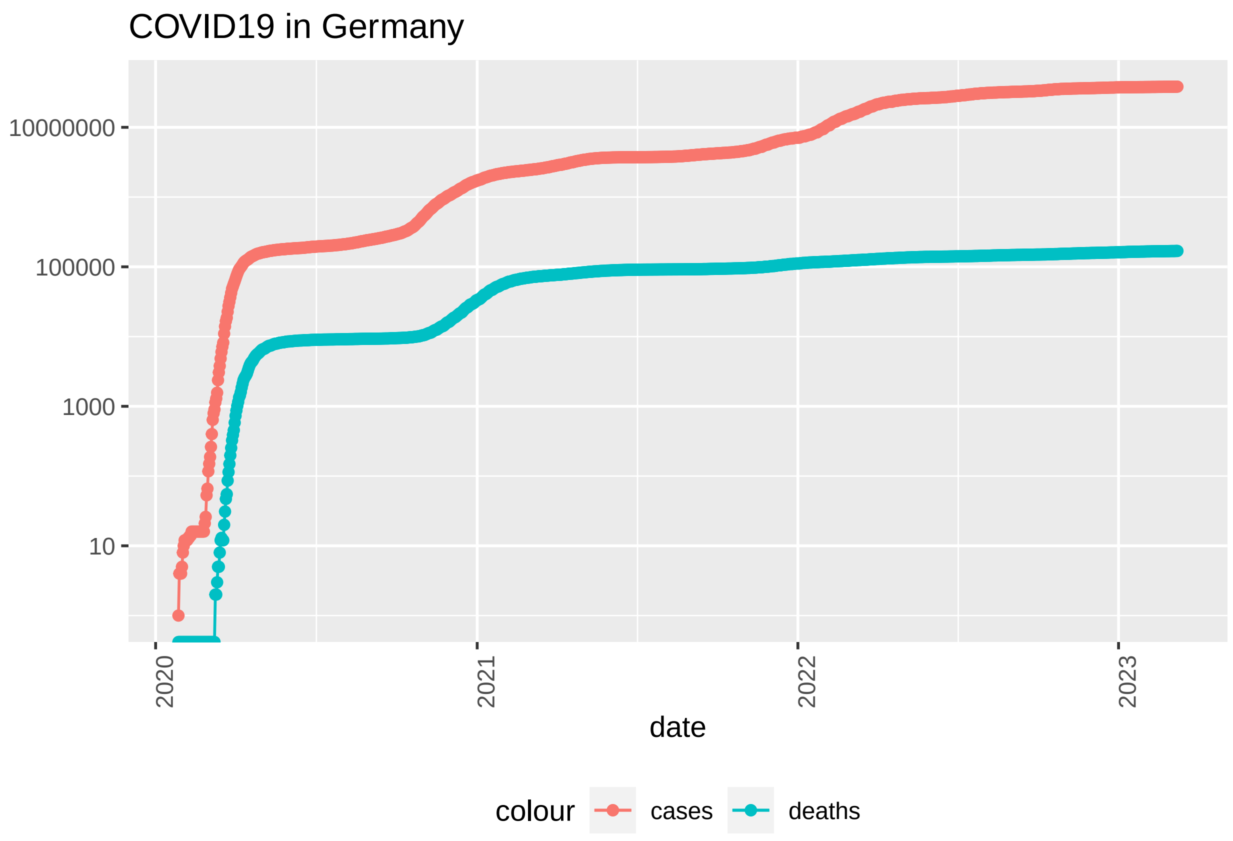
```
## ‘summarise()‘ has grouped output by ’Country_Region’. You can override using
## the ‘.groups‘ argument.
```

```
global_totals
```

```
## # A tibble: 306,827 x 7
## # Groups:   Country_Region [201]
##    Country_Region deaths cases date       population cases_per_thou deaths_per~1
##    <chr>           <dbl> <dbl> <date>          <dbl>          <dbl>        <dbl>
##  1 Afghanistan         0     5 2020-02-24   38928341       0.000128            0
##  2 Afghanistan         0     5 2020-02-25   38928341       0.000128            0
##  3 Afghanistan         0     5 2020-02-26   38928341       0.000128            0
##  4 Afghanistan         0     5 2020-02-27   38928341       0.000128            0
##  5 Afghanistan         0     5 2020-02-28   38928341       0.000128            0
##  6 Afghanistan         0     5 2020-02-29   38928341       0.000128            0
##  7 Afghanistan         0     5 2020-03-01   38928341       0.000128            0
##  8 Afghanistan         0     5 2020-03-02   38928341       0.000128            0
##  9 Afghanistan         0     5 2020-03-03   38928341       0.000128            0
## 10 Afghanistan         0     5 2020-03-04   38928341       0.000128            0
## # ... with 306,817 more rows, and abbreviated variable name 1: deaths_per_thou
```

Similar to the visualization we had the US totals, I am using similar methods for displaying the total number of cases for individual countries. Feel free to replace the 'country' with and country in the dataset to view the total number of cases in that country over time.

```r
country <- "Germany"
global %>%
    filter(Country_Region==country) %>%
    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() +
    theme(legend.position="bottom", axis.text.x=element_text(angle=90)) +
    labs(title=str_c("COVID19 in ", country), y=NULL)
```



We can regroup the dataset further by summing the total number of cases in each country and then calcuating the cases per thousand. This will allow me to visualize which countries had the highest number of reported cases across the entire pandemic.

```
global_total_cases <- global %>%
    group_by(Country_Region) %>%
    summarize(deaths=max(deaths), cases=max(cases),
              population=max(Population),
              cases_per_thou=1000*cases/population)
global_total_cases
```

```
## # A tibble: 201 x 5
##    Country_Region      deaths    cases population cases_per_thou
##    <chr>                <dbl>    <dbl>      <dbl>          <dbl>
##  1 Afghanistan           7896   209451   38928341           5.38
##  2 Albania               3598   334457    2877800         116.
##  3 Algeria               6881   271496   43851043           6.19
##  4 Andorra                165    47890      77265         620.
##  5 Angola                1933   105288   32866268           3.20
##  6 Antarctica               0       11         NA          NA
##  7 Antigua and Barbuda    146     9106      97928          93.0
##  8 Argentina           130472 10044957   45195777         222.
##  9 Armenia               8727   447308    2963234         151.
## 10 Australia             7370  3915992    8118000         482.
## # ... with 191 more rows
```

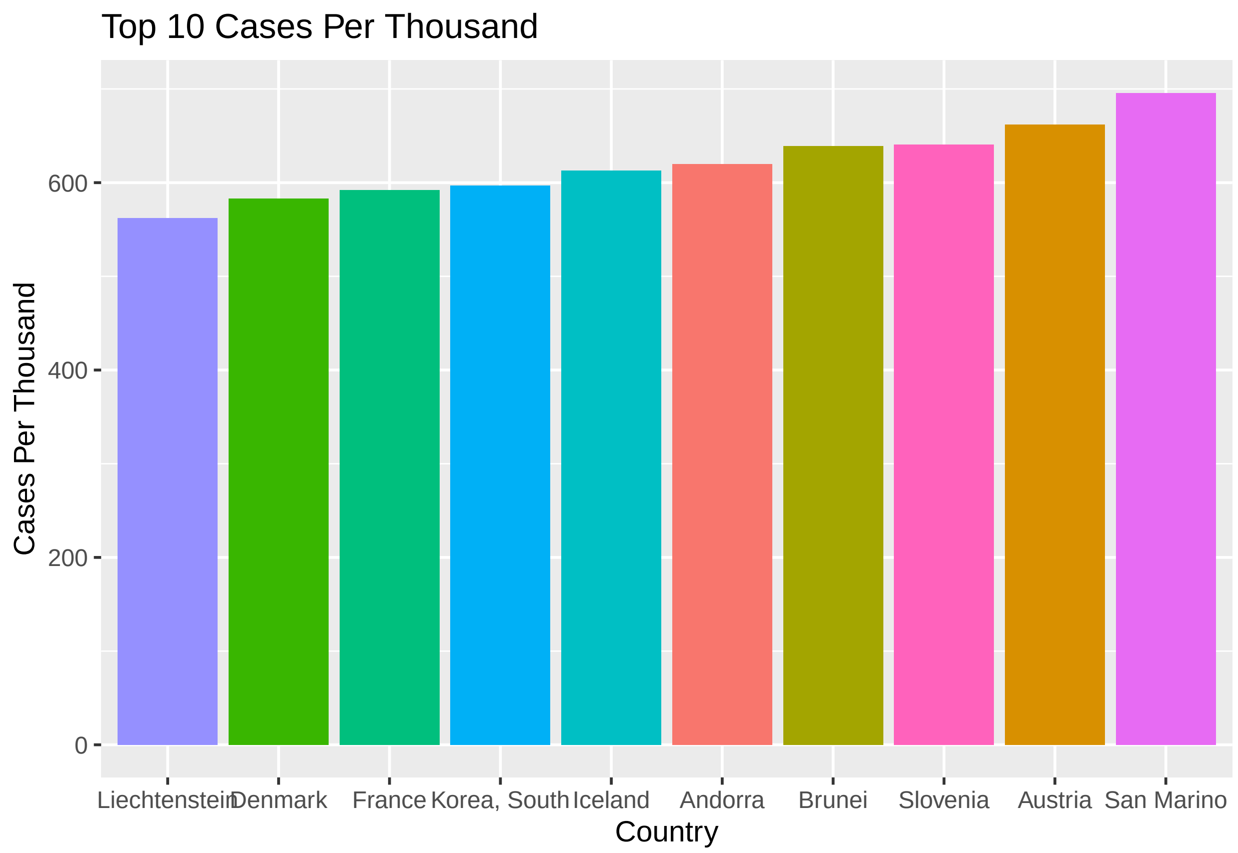
This sorts by cases per thousand.

```
global_total_cases <- global_total_cases[with(global_total_cases, order(-cases_per_thou)),]
global_total_cases <- global_total_cases[1:10,]
global_total_cases
```

```
## # A tibble: 10 x 5
##    Country_Region deaths    cases population cases_per_thou
##    <chr>           <dbl>    <dbl>      <dbl>          <dbl>
##  1 San Marino        122    23616      33938           696.
##  2 Austria         21970  5961143    9006400           662.
##  3 Slovenia         7078  1331707    2078932           641.
##  4 Brunei            225   279661     437483           639.
##  5 Andorra           165    47890      77265           620.
##  6 Iceland           263   209137     341250           613.
##  7 Korea, South    34093 30615522   51269183           597.
##  8 France         161512 38618509   65249843           592.
##  9 Denmark          8296  3404407    5837213           583.
## 10 Liechtenstein      89    21432      38137           562.
```

And here is the visualization for the countries that had the highest amount of cases per thousand.

```
plot1 <- global_total_cases %>% ggplot() +
  labs(title="Top 10 Cases Per Thousand", x="Country", y="Cases Per Thousand") +
  geom_bar(aes(x=reorder(Country_Region, cases_per_thou),
               y=cases_per_thou,
               fill=Country_Region),
           stat="identity",
           show.legend=FALSE)
plot1
```



I then create a model to predict the number of deaths per thousand using the number of cases per thousand.

```
mod <- lm(deaths_per_thou ~ cases_per_thou, data=global_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = global_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2241 -0.2720 -0.2514  0.0484  5.6484
##
## Coefficients:
##                  Estimate Std. Error t value            Pr(>|t|)
## (Intercept)    0.27198616 0.00175164   155.3 <0.0000000000000002 ***
## cases_per_thou 0.00542265 0.00001352   401.2 <0.0000000000000002 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.7713 on 248346 degrees of freedom
##   (58479 observations deleted due to missingness)
## Multiple R-squared:  0.3932, Adjusted R-squared:  0.3932
## F-statistic: 1.609e+05 on 1 and 248346 DF,  p-value: < 0.00000000000000022
```

Conclusion and Bias

Adding these visualizations and models to the global dataset provided a better understanding of how covid effected the rest of the world and shows how the US compares to other countries. From these visualizations we can see that the US was not in top 10 when it came to cases per thousand population.

I think the most likely source for bias is the graph where I showed the number of cases in Germany because the y-axis is on a logarithmic scale. It can be a little deceptive at first glance but logarithmic scales are useful for showing growth over time. Other than that I do not believe there is any other bias.