Covid Data Example

D. ODay

7/19/2021

Packages

We will only need the following packages when working this example. Please import using 'library()' * tidyverse * lubridate

```
library(tidyverse)
```

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

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Tidying and Loading Data

It's important we learn the skills to clean and load our data.

Loading Data

I will start by reading in the data from the four main csv files.

```
## Get current Data in the four files
#They all begin in the same way
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi

filenames <- c("time_series_covid19_confirmed_global.csv",
               "time_series_covid19_deaths_global.csv",
               "time_series_covid19_confirmed_US.csv",
               "time_series_covid19_deaths_US.csv"
               )
urls <- str_c(url_in, filenames)
```

```
global_cases <- read_csv(urls[1])
global_deaths <- read_csv(urls[2])
us_cases <- read_csv(urls[3])
us_deaths <- read_csv(urls[4])
```

After looking at global_cases and global_death, I would like to tidy those datasets and put each variable (date, cases, deaths) in their own column. Also, I don't need Lat and Long for the analysis I'm planning; so, I will get rid of those and rename Region and State to be more R friendly.

```
#takes global cases (doing the same thing to global_deaths), makes everything but Province/State, Count
# and Long into a row. The names (was the column headings) are now going to be
# date. The values goes to cases. Select everything except Lat and Long
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_cases <- global_cases %>%
  pivot_longer( cols = - c('Province/State',
                           'Country/Region', Lat, Long),
                names_to = "date",
                values_to = "cases") %>%
  select(-c(Lat,Long))
```

Now We're going to work on transforming and tidying the data

We'll start by combining the case and death per date into one variable called 'global'; we will do this by joining the cases with the deaths and the renaming our 'country region' and 'Province State' to get rid of the slash. Also, need to notice that our date was not a date object. We will need the *lubridate* package for this.

```
global <- global_cases %>%
  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")
```

Look at the data and see if there's any problems. We think there are a lot of rows with 0. So we also want to filter those out.

```
global <- global %>% filter(cases > 0)
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:137232      Length:137232      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-07-08   1st Qu.:     299
##  Mode  :character   Mode  :character   Median :2020-11-14   Median :    3237
##                                        Mean   :2020-11-12   Mean   :  263876
##                                        3rd Qu.:2021-03-20   3rd Qu.:   56350
##                                        Max.   :2021-07-23   Max.   :34400655
##      deaths
##  Min.   :     0
##  1st Qu.:     3
##  Median :    54
##  Mean   :  6276
##  3rd Qu.:   929
##  Max.   :610720
```

Looking at the summary table, there are a few concerns. First we want to see if the maximum number of cases should really be that large. We can see the data is valid because there are numerous cases with this amount of data.

```
global %>% filter(cases > 28000000)
```

```
## # A tibble: 210 x 5
##    Province_State Country_Region date          cases deaths
##    <chr>          <chr>          <date>        <dbl>  <dbl>
##  1 <NA>           India          2021-05-30 28047534 329100
##  2 <NA>           India          2021-05-31 28175044 331895
##  3 <NA>           India          2021-06-01 28307832 335102
##  4 <NA>           India          2021-06-02 28441986 337989
##  5 <NA>           India          2021-06-03 28574350 340702
##  6 <NA>           India          2021-06-04 28694879 344082
##  7 <NA>           India          2021-06-05 28809339 346759
##  8 <NA>           India          2021-06-06 28909975 349186
##  9 <NA>           India          2021-06-07 28996473 351309
## 10 <NA>           India          2021-06-08 29089069 353528
## # ... with 200 more rows
```

Now we want to take a look at the US cases. When looking at the dataset we can see some weird codes/data types. We want to pivot the dates, keep Admin2, Province/State, Country/Region and Lat/Long.. Select all the important variables. Make date a date object. Select all the remaining columns except Latitude and Longitude.

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

We want to do the same for us_deaths. We can't always assume the two data sets will be in similar formats. But we are for this situation.

```
us_deaths <- us_deaths %>%
  pivot_longer(cols = -(UID:Population), #got rif of
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
```

We will join the two different US datasets

```
US <- us_cases %>%
full_join(us_deaths)
```

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

We want to combine the state and country_region of global dataset

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

Importing csv to grab population variable

```
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/U

uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   UID = col_double(),
##   iso2 = col_character(),
##   iso3 = col_character(),
##   code3 = col_double(),
##   FIPS = col_character(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Lat = col_double(),
##   Long_ = col_double(),
##   Combined_Key = col_character(),
##   Population = col_double()
## )
```

Adding uid csv to global dataset to add the population as a column

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

Visualization

Visualization is a key aspect to help everyone understand the data.

First, we will analyze the COVID data for each U.S. state

Mutate creates a columns calculation based off of arithmetic of other columns Make sure to double check the data for population with a source online

```
us_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
    Population = sum(Population)) %>%
    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 `
```

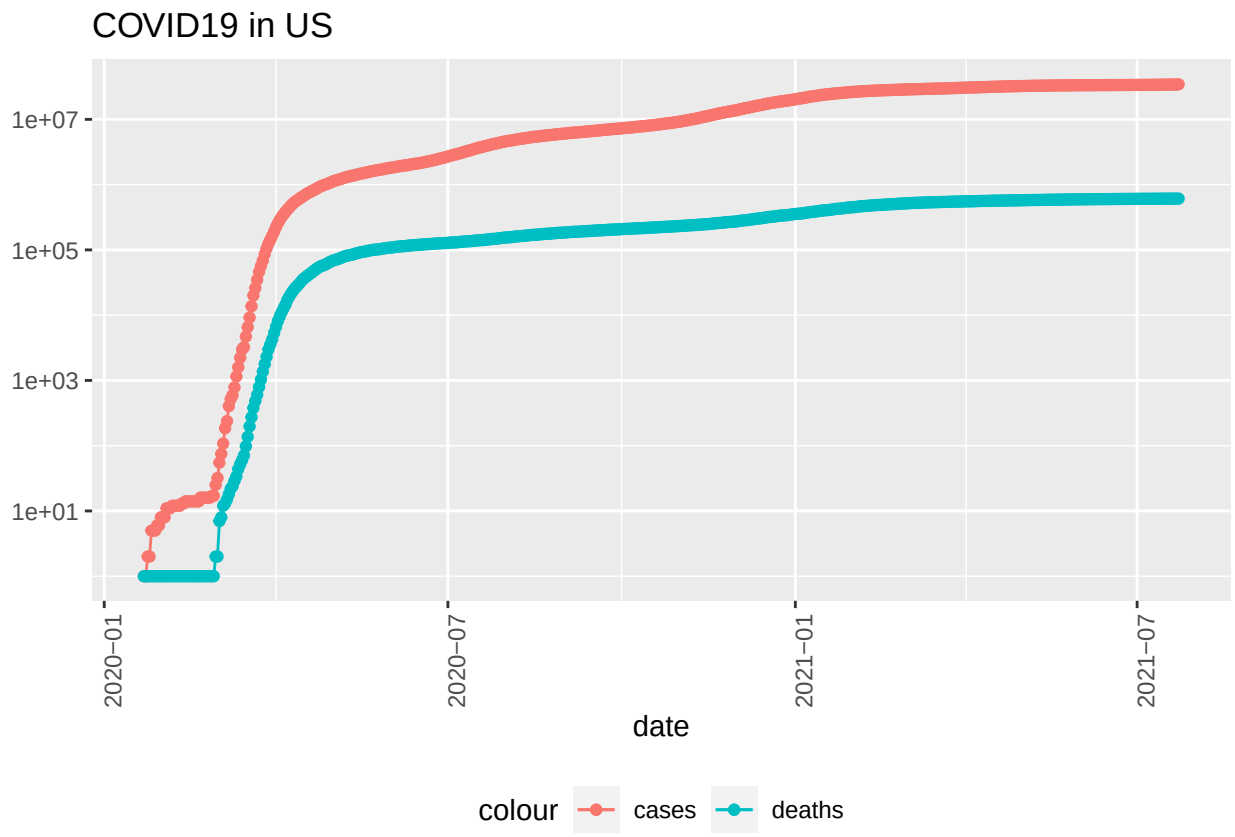
Look at total amount for U.S.

```
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.
```

Want to visualize the US total data

```
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)
```

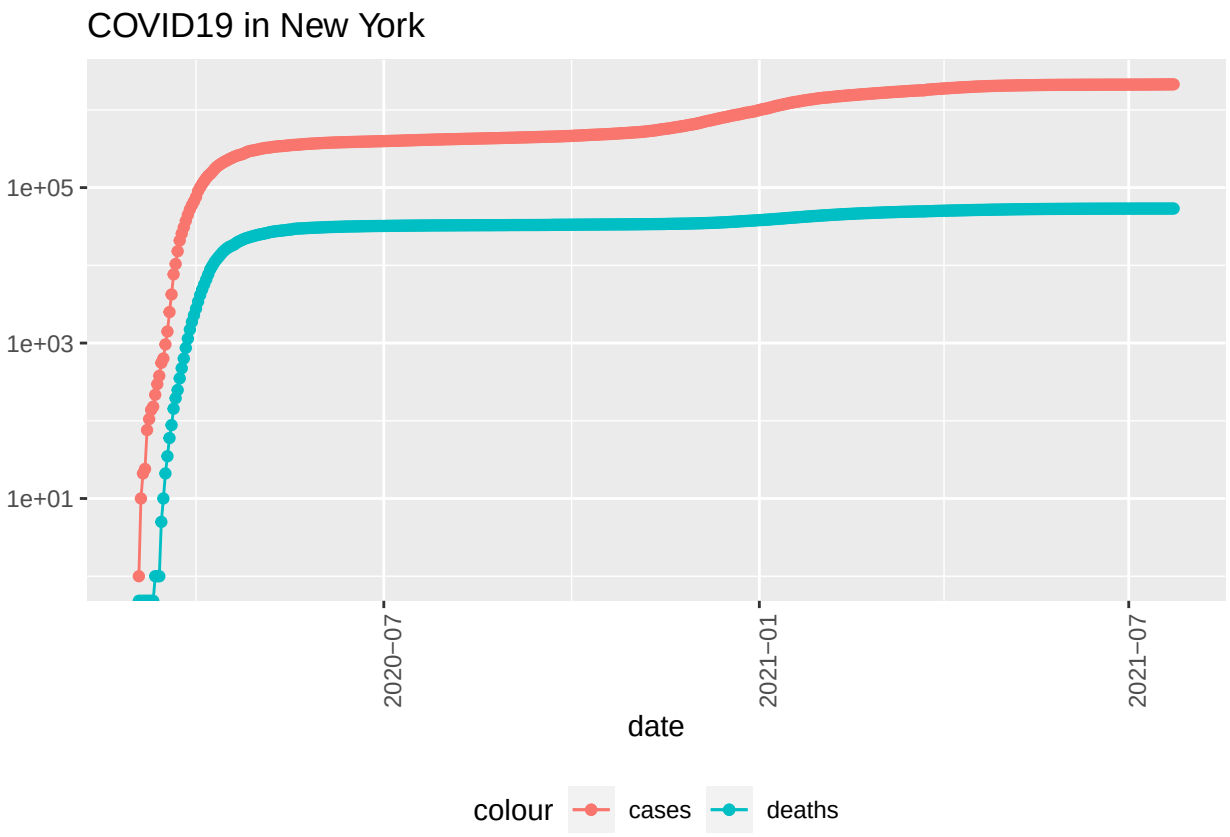


Now we'll look at cases in New York

```
us_by_state %>%
  filter(Province_State == "New York") %>%
  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 New York", y = NULL)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```



We can look at the date with the maximum number of cases and date. Should be the date worked and cases to the daty working

```
max(us_totals$date)
```

```
## [1] "2021-07-23"
```

```
max(us_totals$deaths)
```

```
## [1] 610720
```

Analyze the data

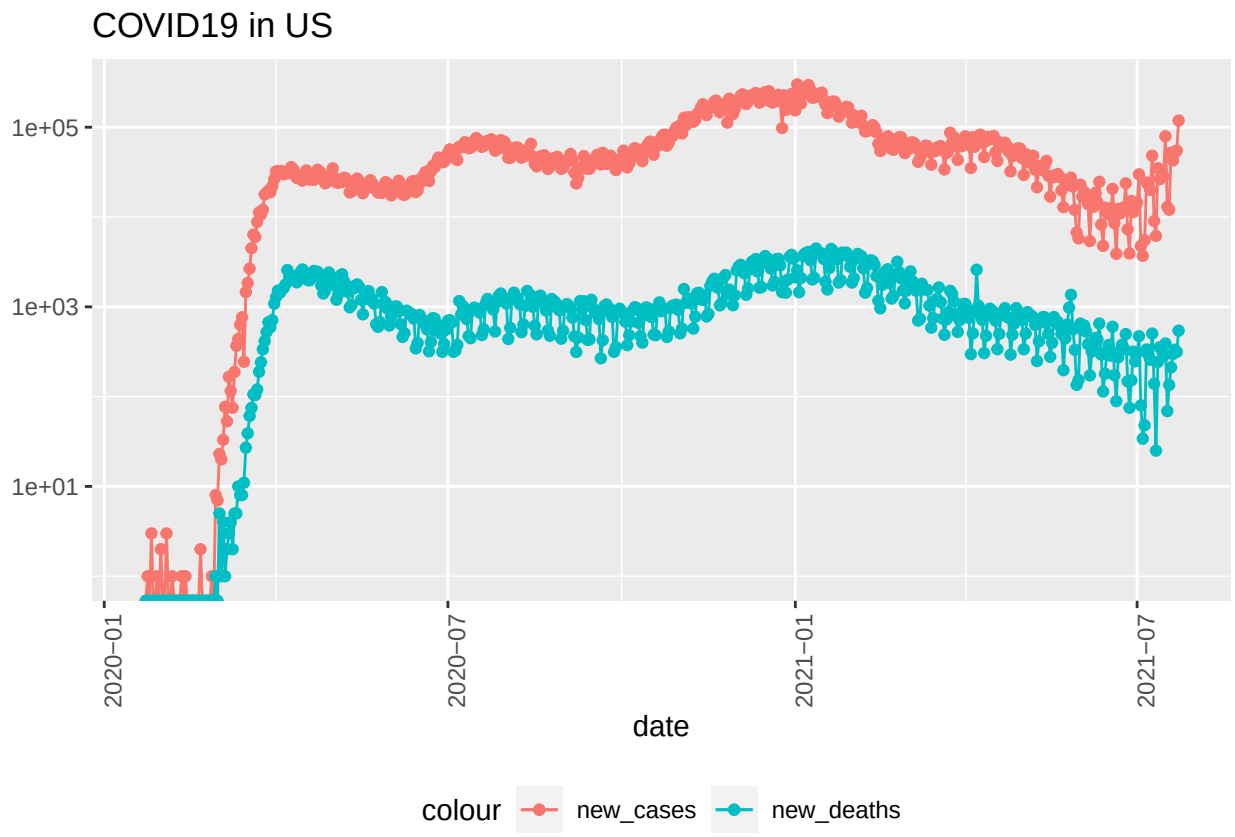
We are going to add some variables to our data so we can look at the change from day to day

```
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))
```

Now we want to visualize the new cases and new deaths in the U.S. At first, the US was looking at the total number of cases. Soon, this lost value so we started looking at new cases and new deaths.

```r
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 = 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)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).
```



Now we want to know what are the best and/or worst states? We'll see the 10 states with smallest/largest deaths per thousands. Remembe rto always ask questions when looking at the data.

```
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) %>%
  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          0.0363           3.32 Northern Mariana Isl~     2    183      55144
##  2          0.326           40.4  Virgin Islands           35   4336     107268
##  3          0.372           28.3  Hawaii                  527  40125    1415872
##  4          0.415           39.5  Vermont                 259  24676     623989
##  5          0.518           99.5  Alaska                  384  73753     740995
##  6          0.663           52.0  Maine                   891  69835    1344212
##  7          0.672           50.9  Oregon                 2836 214869    4217737
##  8          0.683           37.9  Puerto Rico            2566 142359    3754939
```

```
##  9          0.756          133.   Utah                   2425 426418    3205958
## 10          0.797           61.2  Washington             6066 466099    7614893
```

```
us_state_totals %>%
  slice_max(deaths_per_thou, n=10) %>%
  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            2.99          116.  New Jersey      26568 1032255    8882190
##  2            2.77          110.  New York        53801 2133264   19453561
##  3            2.62          104.  Massachusetts   18046  715180    6892503
##  4            2.59          145.  Rhode Island     2739  153447    1059361
##  5            2.52          112.  Mississippi      7502  333180    2976149
##  6            2.49          126.  Arizona         18144  914132    7278717
##  7            2.34          110.  Louisiana       10900  512843    4648794
##  8            2.34          116.  Alabama         11472  567243    4903185
##  9            2.32           98.7 Connecticut      8286  352037    3565287
## 10            2.31          141.  South Dakota     2043  124960     884659
```

Modeling Data

We're going to do a very simple model and make predictions

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

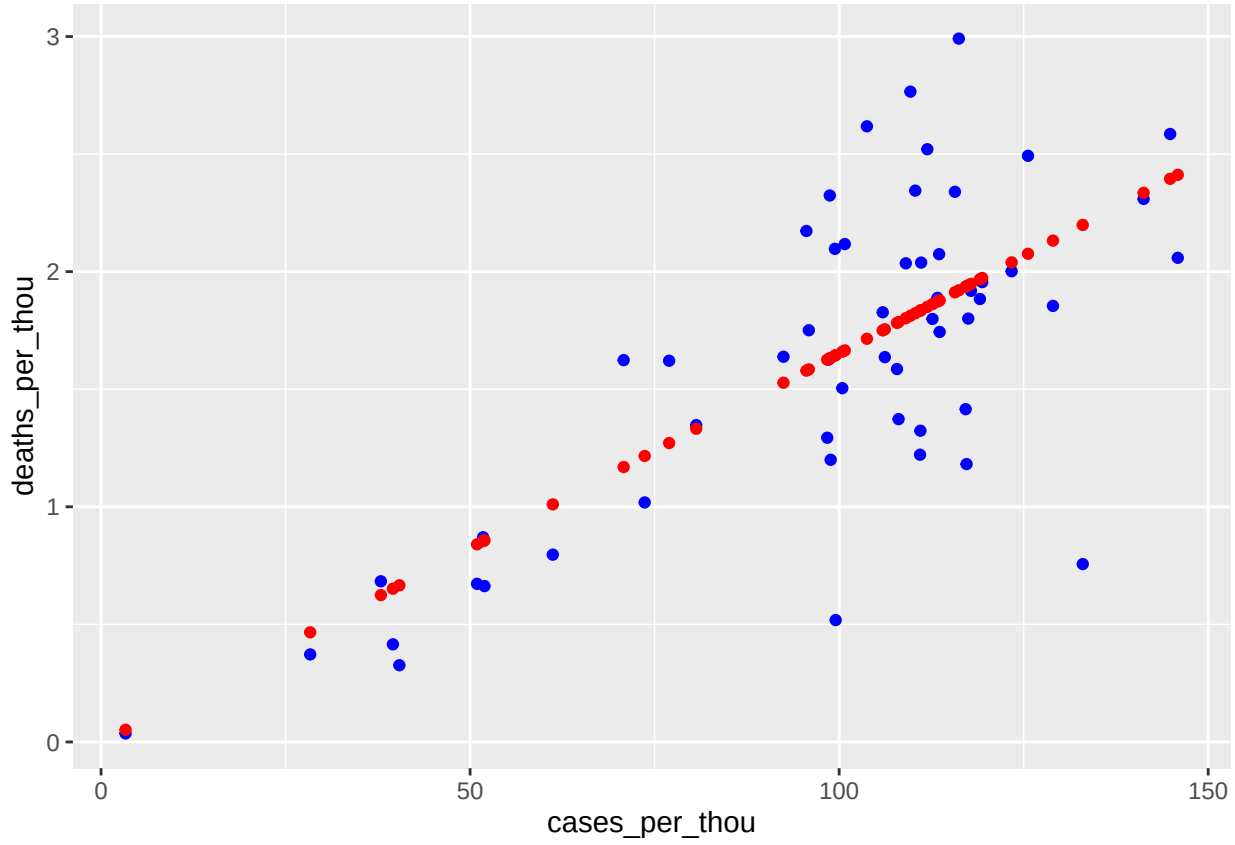
```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = us_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44245 -0.20572 -0.02596  0.21802  1.07027
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.002974   0.216309  -0.014    0.989
## cases_per_thou  0.016554   0.002111   7.841 2.01e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4699 on 53 degrees of freedom
## Multiple R-squared:  0.5371, Adjusted R-squared:  0.5283
## F-statistic: 61.48 on 1 and 53 DF,  p-value: 2.008e-10
```

```
us_tot_w_pred <- us_state_totals %>% mutate(pred = predict(mod))
us_tot_w_pred
```

```
## # A tibble: 55 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          11472 5.67e5    4903185          116.            2.34   1.91
##  2 Alaska             384 7.38e4     740995           99.5           0.518  1.64
##  3 Arizona          18144 9.14e5    7278717          126.            2.49   2.08
##  4 Arkansas          6041 3.72e5    3017804          123.            2.00   2.04
##  5 California       64235 3.89e6   39512223           98.6           1.63   1.63
##  6 Colorado          6910 5.69e5    5758736           98.9           1.20   1.63
##  7 Connecticut       8286 3.52e5    3565287           98.7           2.32   1.63
##  8 Delaware          1698 1.11e5     973764          114.            1.74   1.88
##  9 District of Co~   1146 5.00e4     705749           70.8           1.62   1.17
## 10 Florida          38670 2.52e6   21477737          117.            1.80   1.94
## # ... with 45 more rows
```

Let's plot these predictions with our real data. Other factors needed to predict what causes more deaths. Obviougly, the number of cases aren't a direct correlation. This introduces us to the cyclical process. We would now look further into the other reasons why this data looks like this

```
us_tot_w_pred %>% 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")
```



Now we have to look at bias

- Potenital bias:
  - Topic Chosen
    1. You chose the topic so you have different feelings about it
  - Variables Used
  - How you phrase questions in a survey
  - Clothing worn during the survey
  - How do you handle outliers
    1. A lot of times the most interesting data is in the outliers
  - Bias in the machine learning model?

Why does bias exist?

- It is not necessarily a bad thing
- Fear keeps us alive
- Tend to be afraid of those who are different than us
- If you're a left-wing political person, would be best to put on right-wing hat when working with data