Covid-19 Global, US, and State Trends

K.Smith

1/30/2022

Is the Covid-19 Virus Becoming Less Deadly?

Goals

- Analyze recent Covid-19 cases and death's data from recent years to present.
  - Look at global, US, and individual states.

- Find reliable, current sources for data through scraping (scraping allows realtime results).
  - Search open research forums and maybe see which organizations are cited more frequently.
- Look for any notable trends either way and present them.
  - Transform, clean, and tidy the data.
  - Build models, charts and graphs.

```
# import necessary packages
library(tidyselect)
library(tidyverse)
library(tidyr)
library(lubridate)
library(dplyr)
library(ggplot2)
```

Data Sourcing

- The Data for this analysis was scraped from the Johns Hopkins research centers public data repository. It seemed to be a very reliable source and they update it daily so it is extremely relevant.
- It can be found for download here.
- Or copy and paste this link into your browser https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series.

```
## Get Data needed by scraping the current, raw data from Johns Hopkins Github Repository
url_in <- ("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_co

## We will be using several sources of raw data but they all start with the same beginning url link. So
file_names <- c("time_series_covid19_confirmed_US.csv",
                "time_series_covid19_deaths_US.csv",
                "time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_global.csv")

## Concantenate them together and store them in a vector of string variables
urls <- str_c(url_in, file_names)
```

Gather, Clean, Tidy, and Transform Data

Global Data

```
## Read data into a DataFrame from files in urls
global_cases  <- read.csv(urls[3])
global_deaths <- read.csv(urls[4])

## Preview dataframes column names:
names(global_cases[1:6])
```

```
## [1] "Province.State" "Country.Region" "Lat"            "Long"
## [5] "X1.22.20"       "X1.23.20"
```

- It looks as if the individual dates are presented as a single column each. They are also of a type that we are not interested in working with (no offense to int's).
- In order to analyze changes over time we are going to have to transform these datasets. If we would like to see counts per date so we can answer some questions, then each date will be a single observation.
- This means we must arrange the data so that each 'row' is a single day and the variables ('columns') are either discarded if they offer no help with our analysis or transformed into usefull versions of themselves or combined with other variables to achieve this result.

```
## Transfrom global cases Dataframe
global_cases <- global_cases %>%
  pivot_longer(cols = -c('Province.State',
                         'Country.Region', Lat, Long),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat, Long))

## Transfrom global deaths Dataframe
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province.State',
                         'Country.Region', Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long))

## Join the two together to make a Global DataFrame
global <- global_cases %>%
  full_join(global_deaths)
```

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

```
## Rename columns
global <- global %>%
  rename("Country_Region" = 'Country.Region',
         "Province_State" = 'Province.State')


## Reformat the date column
```

```
### Drop the "X" off the front of every date string
global$date <- substring(global$date, 2)

### Convert them all to datetime objects and format to MDY
global <- global %>%
  mutate(date = mdy(date))

## Remove zero cases to shrink our data set a bit.
global <- global %>% filter(cases > 0)
summary(global)
```

```
##    Province_State          Country_Region        date
##           :136067   China         : 25236   Min.   :2020-01-22
##  Anhui    :   765   Canada        : 10333   1st Qu.:2020-09-03
##  Beijing  :   765   United Kingdom:  9056   Median :2021-03-05
##  Chongqing:   765   France        :  8393   Mean   :2021-03-02
##  Fujian   :   765   Australia     :  5920   3rd Qu.:2021-08-31
##  Guangdong:   765   Netherlands   :  3557   Max.   :2022-02-24
##  (Other)  : 58136   (Other)       :135533
##      cases              deaths
##  Min.   :       1   Min.   :     0
##  1st Qu.:     574   1st Qu.:     4
##  Median :    7380   Median :   105
##  Mean   :  479059   Mean   :  9861
##  3rd Qu.:  121061   3rd Qu.:  2005
##  Max.   :78798989   Max.   :944830
##
```

US Data

- Above we can see that removing rows with 0 values in them cleans our data up quite a bit.
- For this analysis this is going to be fine, but many times you would not want to remove 0 values so quickly without thinking about what consequences this will have on your analysis.
- Now we will repeat our previous import, clean, and tidy process from the global data with the US datasets.

```
## Create DataFrame from first file in urls
US_cases      <- read_csv(urls[1])
```

```
## Rows: 3342 Columns: 776
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (770): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20,...
```

```
##
## 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.
```

```
US_deaths     <- read.csv(urls[2])

## Now let's repeat our previous tidy with global over the US datasets

## Preview DataFrames
#US_cases
#US_deaths

## Lets transform our US_cases to look more like our US_deaths so we may join them

## Pivot on date with cases as counter, Format date from chr to datetime
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) %>%
  select(-c(Lat, Long_))

## Reformat the date column

### Drop the "X" off the front of every date string
US_deaths$date <- substring(US_deaths$date, 2)
#US_deaths

### Convert them all to datetime objects and format to MDY
US_deaths <- US_deaths %>%
  mutate(date = mdy(date))
```

```
## Warning: 3342 failed to parse.
```

```
#US_deaths

## We have transformed our US_cases dataframe to look like our US_deaths so we can join them
US <- US_cases %>%
  full_join(US_deaths)
```

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

```
#US
## Lets get rid of rows where cases are 0
US <- US %>% filter(cases > 0)
sample_n(US, 10)
```

```
## # A tibble: 10 x 7
```

```
##    Admin2     Province_State Country_Region Combined_Key date       cases deaths
##    <chr>      <chr>          <chr>          <chr>        <date>     <dbl>  <int>
##  1 Callaway   Missouri       US             Callaway, M~ 2021-12-18  8455     97
##  2 Trempeale~ Wisconsin      US             Trempealeau~ 2020-11-13  1672      7
##  3 Hamilton   Ohio           US             Hamilton, O~ 2021-03-14 74636   1187
##  4 Wapello    Iowa           US             Wapello, Io~ 2021-04-03  4226    119
##  5 Marshall   South Dakota   US             Marshall, S~ 2021-07-26   375      6
##  6 Cassia     Idaho          US             Cassia, Ida~ 2021-12-09  4020     43
##  7 Petersburg Virginia       US             Petersburg,~ 2021-07-19  3926     89
##  8 Limestone  Texas          US             Limestone, ~ 2020-06-23    29      1
##  9 Catawba    North Carolina US             Catawba, No~ 2021-07-08 19863    311
## 10 Gallatin   Illinois       US             Gallatin, I~ 2020-09-22    67      2
```

- If we are interested in finding out rates per-capita and similar variables that require population data, then we must add this to our data as it doesn't include it. Luckily there is a population look up table with country keys we can scrape and join with our current datasets.

```
## Create a Combined Key variable in our global dataset to match our US
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ', ',
        na.rm = TRUE,
        remove = FALSE)

## Source our population data:
uid_lookup_url <- c("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/",
                    "master/csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv")
uid_lookup_url <- str_c(uid_lookup_url[1], uid_lookup_url[2])

## Read in URL
uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, code3, iso2, iso3, Admin2, UID, FIPS))
```

```
## Rows: 4218 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.
```

```
sample_n(uid, 10)
```

```
## # A tibble: 10 x 4
##    Province_State      Country_Region Combined_Key                Population
##    <chr>               <chr>          <chr>                            <dbl>
##  1 Virginia            US             Patrick, Virginia, US            17608
##  2 Maryland            US             Out of MD, Maryland, US             NA
```

```
##  3 Jamtland Harjedalen Sweden         Jamtland Harjedalen, Sweden     130810
##  4 Rio de Janeiro      Brazil         Rio de Janeiro, Brazil        17264943
##  5 Arkansas            US             Crawford, Arkansas, US           63257
##  6 Pennsylvania        US             Dauphin, Pennsylvania, US       278299
##  7 Quintana Roo        Mexico         Quintana Roo, Mexico           1723259
##  8 Georgia             US             Webster, Georgia, US              2607
##  9 Florida             US             Clay, Florida, US               219252
## 10 Missouri            US             Texas, Missouri, US              25398
```

```
## Join with our global dataset
global <- global %>%
  right_join(uid, by = c("Province_State", "Country_Region", "Combined_Key")) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, Population,
         Combined_Key)
## Lets see what we have so far:
sample_n(global, 10)
```

```
## # A tibble: 10 x 7
##    Province_State             Country_Region date        cases deaths Population
##    <chr>                      <chr>          <date>      <int>  <int>      <dbl>
##  1 Tibet                      China          2020-11-06      1      0    3648100
##  2 Saint Barthelemy           France         2020-05-31      6      0       9885
##  3 Indiana                    US             NA             NA     NA      16581
##  4 Jiangxi                    China          2021-02-10    935      1   45188635
##  5 New South Wales            Australia      2021-10-10  69205    493    8118000
##  6 British Columbia           Canada         2021-09-29 186245   1953    5214805
##  7 Bonaire, Sint Eustatius a~ Netherlands    2021-02-10    196      3      26221
##  8 Macau                      China          2021-09-09     63      0     649342
##  9 Australian Capital Territ~ Australia      2021-11-14   1953     14     428100
## 10 Guernsey                   United Kingdom 2020-11-07    278     16      63000
## # ... with 1 more variable: Combined_Key <chr>
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:64934       Length:64934       Min.   :2020-01-22   Min.   :      1
##  Class :character   Class :character   1st Qu.:2020-08-26   1st Qu.:    160
##  Mode  :character   Mode  :character   Median :2021-02-27   Median :    676
##                                        Mean   :2021-02-24   Mean   :  17363
##                                        3rd Qu.:2021-08-28   3rd Qu.:   2423
##                                        Max.   :2022-02-24   Max.   :1292781
##                                        NA's   :4132         NA's   :4132
##      deaths          Population        Combined_Key
##  Min.   :    0.0   Min.   :8.600e+01   Length:64934
##  1st Qu.:    1.0   1st Qu.:6.572e+04   Class :character
##  Median :    4.0   Median :1.384e+06   Mode  :character
##  Mean   :  310.6   Mean   :1.790e+07
##  3rd Qu.:   23.0   3rd Qu.:2.585e+07
##  Max.   :13931.0   Max.   :1.412e+09
##  NA's   :4132      NA's   :900
```

Before we continue we have to do something with our na values in the data so it does not effect our analysis.

```
## Oh dear it looks as though we have some na values to remove so lets take care of that:
global <- na.omit(global)
#summary(global)

## Join with our US dataset
US <- US %>%
  right_join(uid, by = c("Province_State", "Country_Region", "Combined_Key")) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, Population,
         Combined_Key)

## Lets see what we have so far:
#sample_n(US, 10)
#tail(US)

## Some na values here as well to remove so lets take care of that
US <- na.omit(US)
#tail(US)
sample_n(US, 10)
```

```
## # A tibble: 10 x 7
##    Province_State Country_Region date       cases deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <int>      <dbl> <chr>
##  1 Kentucky       US             2022-02-19  3816     46      12942 Webster, Ke~
##  2 Illinois       US             2020-04-15     9      0      37684 Jefferson, ~
##  3 Georgia        US             2020-08-21   611      7      22383 Burke, Geor~
##  4 Nebraska       US             2021-03-07   481      4       5003 Thayer, Neb~
##  5 Nebraska       US             2020-08-17    21      1       6298 Antelope, N~
##  6 Arkansas       US             2021-02-24  3059     75      23528 Poinsett, A~
##  7 Texas          US             2020-06-17  1306     27      72971 Walker, Tex~
##  8 Washington     US             2021-07-26  1048     14      22471 Pacific, Wa~
##  9 Iowa           US             2021-05-10  1124     23       8886 Palo Alto, ~
## 10 Illinois       US             2021-03-10   441      4       4828 Gallatin, I~
```

Visualizations

- Ok we have scraped and wrangled, cleaned and tidied, transformed and merged... I think it is high time we build some visualizations!

```
## Lets start in the US by state
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  select(Country_Region, Province_State, date,
         cases, deaths, Population) %>%
  ungroup()

## A US totals
US_totals <- US_by_state %>%
```

```
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  select(Country_Region, date,
         cases, deaths, Population) %>%
  ungroup()
```

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

```
## Create a new column variable for deaths per million
US_totals$deaths_per_mill <- US_totals$deaths*1000000
US_totals$deaths_per_mill <- round(US_totals$deaths_per_mill / US_totals$Population)
#sample_n(US_by_state, 10)
tail(US_totals)
```

```
## # A tibble: 6 x 6
##   Country_Region date          cases deaths Population deaths_per_mill
##   <chr>          <date>        <dbl>  <int>      <dbl>           <dbl>
## 1 US             2022-02-19 77342204 915448  329940452            2775
## 2 US             2022-02-20 77363467 915826  329940452            2776
## 3 US             2022-02-21 77439877 916584  329940452            2778
## 4 US             2022-02-22 77522819 918409  329940452            2784
## 5 US             2022-02-23 77633662 923178  329940452            2798
## 6 US             2022-02-24 77696557 925319  329940452            2805
```

```
## Create a new column variable for proportion of deaths to cases
US_totals$death_rate_percent <- round((US_totals$deaths / US_totals$cases)*100, digits=2)
sample_n(US_totals, 100)
```

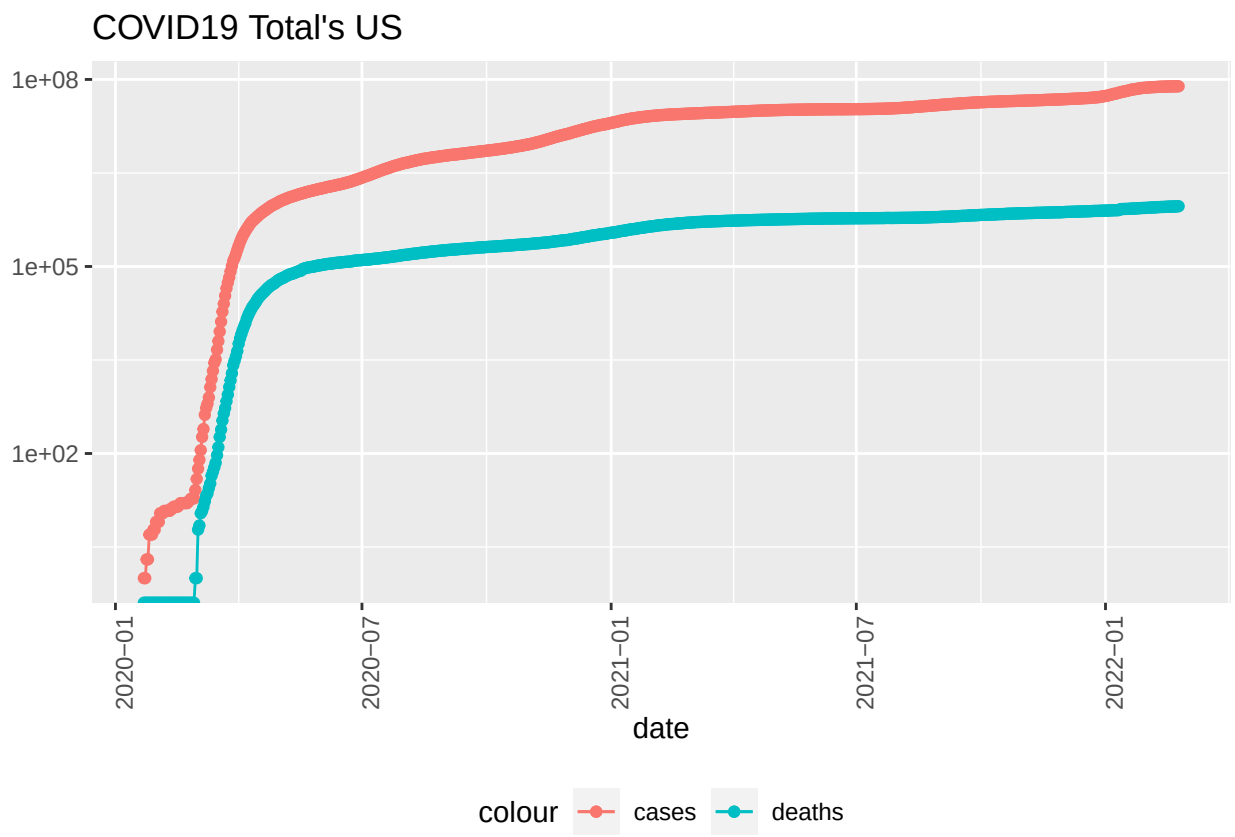
```
## # A tibble: 100 x 7
##    Country_Region date          cases deaths Population deaths_per_mill
##    <chr>          <date>        <dbl>  <int>      <dbl>           <dbl>
##  1 US             2020-06-19  2190392 118532  329507216             360
##  2 US             2021-03-22 29618635 533466  329940452            1617
##  3 US             2021-08-01 34729558 600964  329940452            1821
##  4 US             2020-11-18 11414203 248223  329940452             752
##  5 US             2020-03-29   138147   3086  309342168              10
##  6 US             2020-09-04  6131203 185249  329929338             561
##  7 US             2021-09-17 41532439 647820  329940452            1963
##  8 US             2021-06-03 33051138 585669  329940452            1775
##  9 US             2021-06-29 33377134 593167  329940452            1798
## 10 US             2020-03-18     9042    184  237620951               1
## # ... with 90 more rows, and 1 more variable: death_rate_percent <dbl>
```

```
## First visualization
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 Total's US", y = NULL)
```

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

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



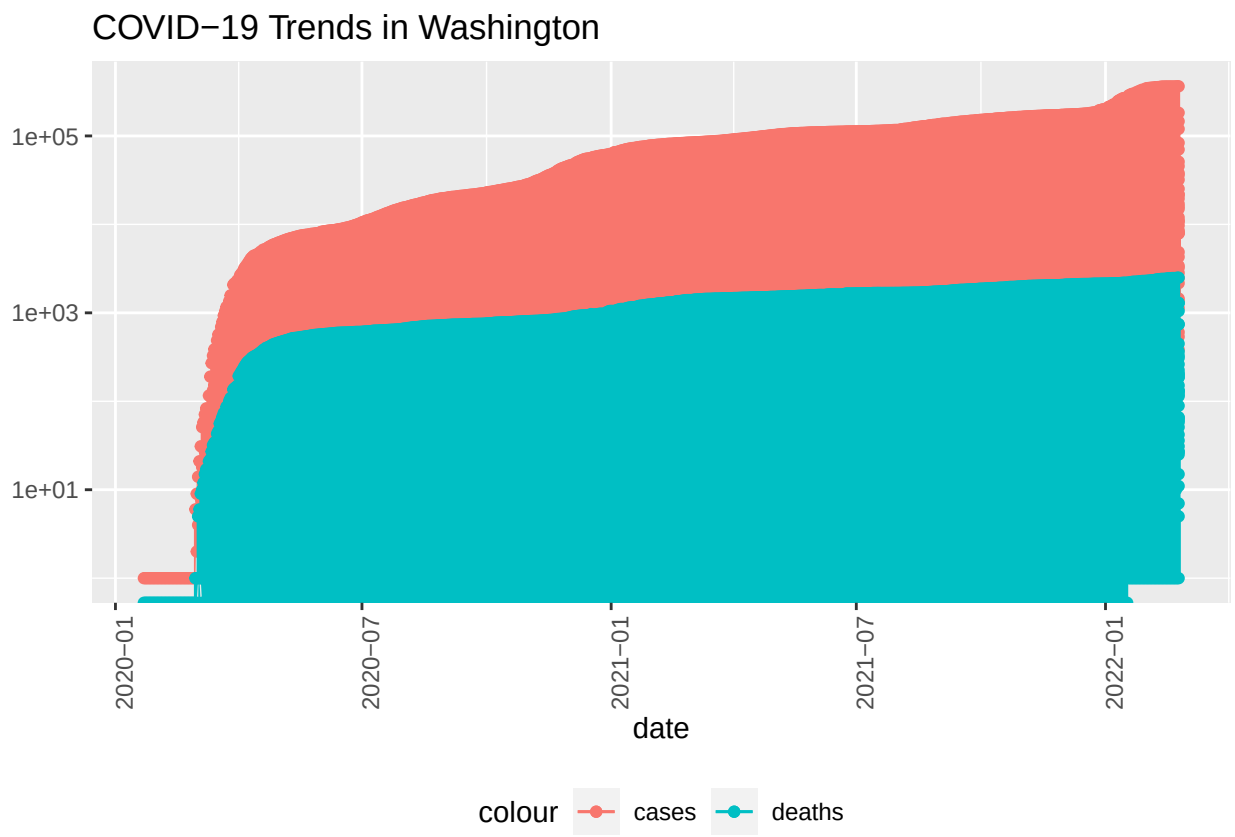
- We can also single out a state of interest and do the same, I picked Washington because that is where I currently live. You can insert any state that may interest you.

```
## Pick a state to analyze

state_of_interest <- "Washington"
state <- state_of_interest
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("COVID-19 Trends in ", state), y = NULL)
```

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

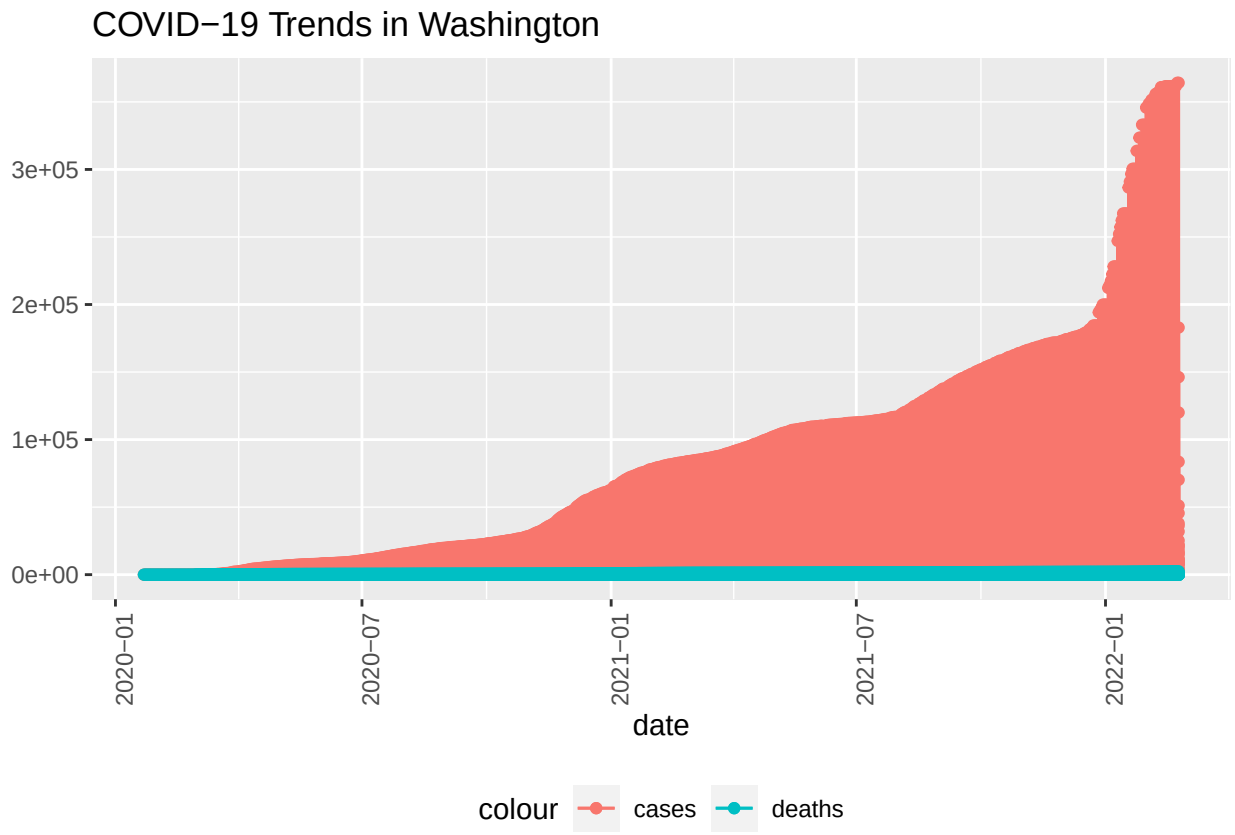
```
## Warning: Transformation introduced infinite values in continuous y-axis
```



- At the time of this analysis there is a spike in cases and we can see that infections continue to gain in numbers.
- It is worth noting that although the cases are rising the death's seem to be rising as well but at a slower rate than case's.
- One could make the argument that the virus is becoming less fatal.
- This could be for many reasons, vaccination status of the public, the virus becoming less fatal on it's own etc.
- If we really wanted to accentuate this fact we could recreate the same plot with death's as a baseline:

```
state <- state_of_interest
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_continuous() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID-19 Trends in ", state), y = NULL)
```



- If we use death's as a baseline we can see the divergence much more pronounced between case's and death's.
- I am not sure how I feel about this. If I was trying to prove that the virus was becoming less deadly for some reason I would choose this plot.
- I am not sure if this is a more bias plot than the logarithm plot. I will leave it to the reader to decide, but I thought it was important to include both to underline the power we have as data scientist's to tell a narrative.
- At the time of this analysis there is a global and local spike in cases (2-7-2022). I provided some code below to check what date is the most recent in the dataset.

```
## If you want to see how current your data is:
current_date <- max(US_by_state$date)
current_date <- paste("Data is thru:", current_date)

## Total number of cases up to date
cases_to_date <- (max(US_totals$cases)/1000000)
```

```
cases_to_date <- round(cases_to_date, digits = 1)
cases_to_date <- paste("Current Total Case Count:", cases_to_date, "million cases")

## Total number of deaths up to date
deaths_to_date <- max(US_totals$deaths)

deaths_to_date <- (max(US_totals$deaths)/1000000)
deaths_to_date <- round(deaths_to_date, digits = 1)
deaths_to_date <- paste("Current Total Death Count:", deaths_to_date, "million cases")

current_date; cases_to_date; deaths_to_date;
```

```
## [1] "Data is thru: 2022-02-24"
```

```
## [1] "Current Total Case Count: 77.7 million cases"
```

```
## [1] "Current Total Death Count: 0.9 million cases"
```

Creating variables to measure new cases and new deaths.

```
## We are going to add some more variables to answer more questions about the data

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))
tail(US_totals %>% select(new_cases, new_deaths, everything()))
```

```
## # A tibble: 6 x 9
##   new_cases new_deaths Country_Region date          cases deaths Population
##       <dbl>      <int> <chr>          <date>        <dbl>  <int>      <dbl>
## 1     28023        594 US             2022-02-19 77342204 915448  329940452
## 2     21263        378 US             2022-02-20 77363467 915826  329940452
## 3     76410        758 US             2022-02-21 77439877 916584  329940452
## 4     82942       1825 US             2022-02-22 77522819 918409  329940452
## 5    110843       4769 US             2022-02-23 77633662 923178  329940452
## 6     62895       2141 US             2022-02-24 77696557 925319  329940452
## # ... with 2 more variables: deaths_per_mill <dbl>, death_rate_percent <dbl>
```

```
## Now that we have created variables for our new deaths and cases, lets visualize them
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)
```

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

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

```
## Warning in self$trans$transform(x): NaNs produced
```

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

```
## Warning in self$trans$transform(x): NaNs produced
```

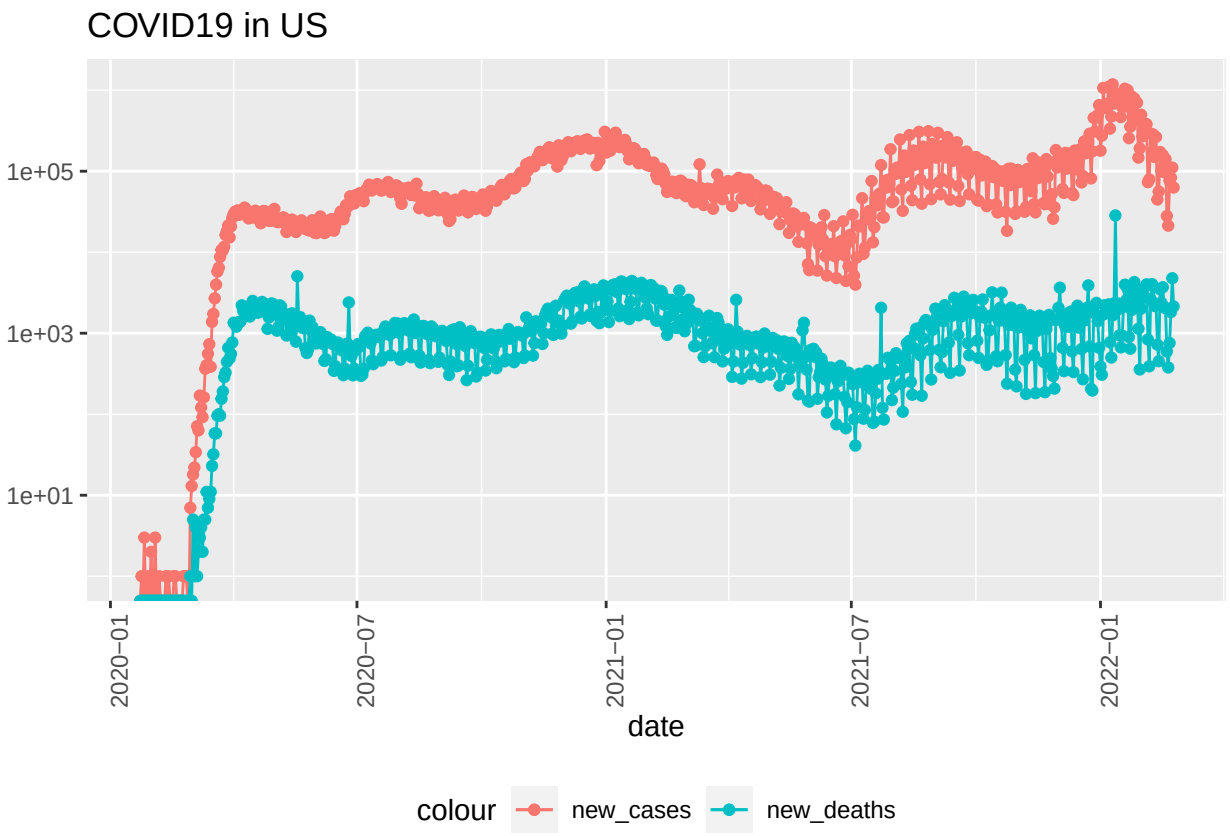
```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## 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 2 rows containing missing values (geom_point).
```



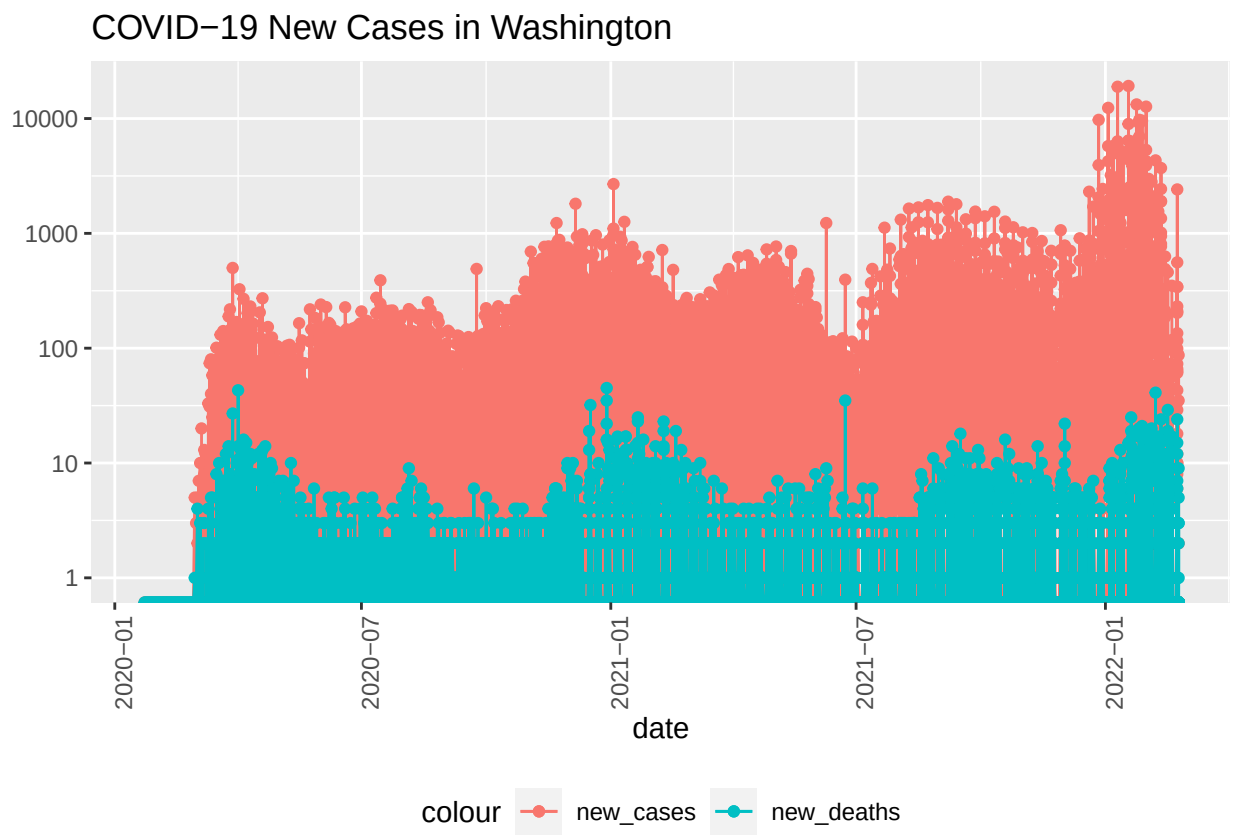
- We can see that when the pandemic began, we were at exponential growth when looking at total cases, or total deaths.
- This is not an effective way to communicate data which is changing extremely rapidly. *Because of our ability to test for and even diagnose this virus you can see that early on the numbers were growing at exponential rates.
- This was in part due simply because of our methods for counting and measuring these things were adapting and spreading until they were adopted in enough countries throughout the world to give more accurate measurements.
- This is why it is important to measure "new cases" and "new deaths" and proportion measurements as opposed to totals.

We can do the same for state level data with new cases and new deaths:

```
## Pick a state to analyze
state <- "Washington"
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("COVID-19 New Cases in ", state), y = NULL)
```

```
## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning: Removed 3 row(s) containing missing values (geom_path).

## Warning: Removed 295 rows containing missing values (geom_point).

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

```
## Warning: Removed 205 rows containing missing values (geom_point).
```



- Try different states yourself and compare. *But what if we wanted to find the worst and best states?
- How would we even decide this?
- Intuition would tell us that cases and deaths might be somewhat a function of population.
- Without proving this yet we can just filter states with the highest and lowest rates of cases and deaths per thousand.

```
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    # Preview

## Best ten states with lowest rates
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>           <int>  <dbl>      <dbl>
##  1           0.953           156. Vermont           156  25474     163774
##  2           0.987           167. Hawaii            962 163111     974563
##  3           1.11            162. Washington       2497 364166    2252782
##  4           1.14            157. Maine             335  46254     295003
##  5           1.18            150. Virginia         1349 172124    1147532
##  6           1.26            248. Nebraska          718 141560     571327
##  7           1.29            136. Oregon           1045 110224     812855
##  8           1.33            292. Utah             1539 338537    1160437
##  9           1.39            256. North Carolina   1541 284450    1111761
## 10           1.74            347. Alaska            501 100063     288000
```

```
## Worst ten states with highest rates
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>           <int>   <dbl>      <dbl>
##  1            4.95           268. New York        12673  684898    2559903
##  2            4.35           224. Michigan         7601  391035    1749343
##  3            3.82           338. Rhode Island     2442  215862     638931
##  4            3.81           234. New Jersey       3556  217704     932202
##  5            3.80           432. Florida         10320 1173497    2716940
##  6            3.74           260. West Virginia     666   46376     178124
##  7            3.52           327. South Carolina   1841  171094     523542
##  8            3.50           277. Arizona         15710 1243994    4485414
##  9            3.33           276. Alabama          2194  181577     658573
## 10            3.29           249. Tennessee        3081  233156     937166
```

Modelling

- Without getting too far into how accurate this data is and how it is reported (which are very important), I would like to start with some very simple modelling of this data as it sits without adding any extra variables at this time.
- If we wanted to try to predict say a states deaths per thousand based off from another variable, what would that look like?

```
# Turn off scientific notation globally for outputs to be full decimal form
options(scipen = 999)

## Lets just look at deaths per thousand as a function of cases per thousand
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
## -1.58254 -0.52203  0.03972  0.57545  2.21134
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.863780   0.527912   1.636  0.10820
## cases_per_thou 0.007010   0.002139   3.278  0.00193 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8432 on 49 degrees of freedom
## Multiple R-squared:  0.1798, Adjusted R-squared:  0.1631
## F-statistic: 10.74 on 1 and 49 DF,  p-value: 0.001929
```

- We can see that the mean for deaths per thousand is our Beta_0 (Intercept). And our Beta_1 (cases_per_thousand) tells us for every case we have that many deaths.

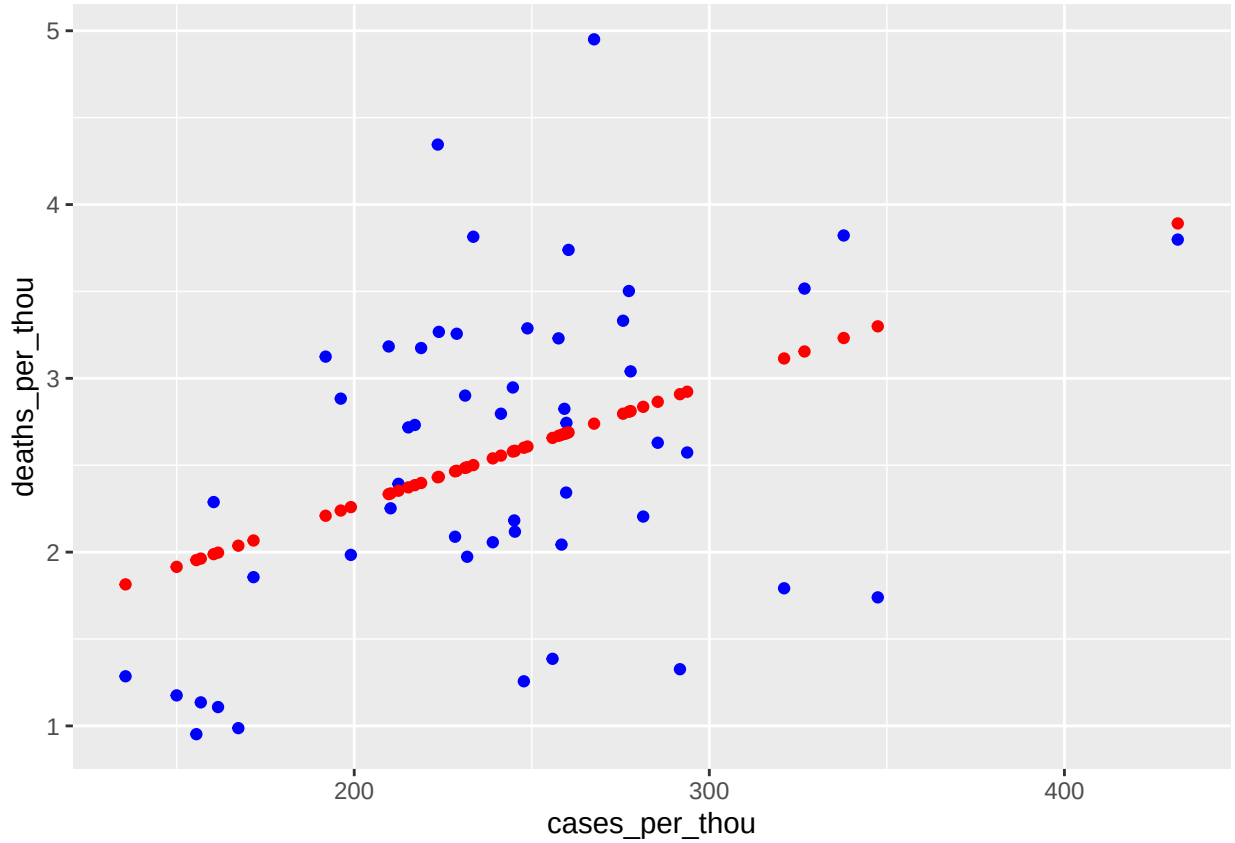
```
## Lets continue our modelling.
## I want to define our lower and upper bounds:
lower <- US_state_totals %>%
  slice_min(cases_per_thou)
lower  ## 122 at the time of this analysis
```

```
## # A tibble: 1 x 6
##   Province_State deaths  cases population cases_per_thou deaths_per_thou
##   <chr>           <int>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 Oregon           1045 110224     812855           136.            1.29
```

```
upper <- US_state_totals %>%
  slice_max(cases_per_thou)
upper   ## 416 at the time of this analysis
```

```
## # A tibble: 1 x 6
##   Province_State deaths   cases population cases_per_thou deaths_per_thou
##   <chr>           <int>   <dbl>      <dbl>          <dbl>           <dbl>
## 1 Florida         10320 1173497    2716940           432.            3.80
```

```
## Add a column of predictions from the model in our US state totals dataframe: (at this time our model
# predicting deaths per thousand using just one independent variable =cases_per_thou)
US_state_totals<- US_state_totals %>%
  mutate(pred = predict(mod))

## Lets plot out our predictions against actual values and see how it looks
US_state_totals %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") + # Actual values in blue
  geom_point(aes(x = cases_per_thou, y = pred), color = "red") # Predicted values in red
```



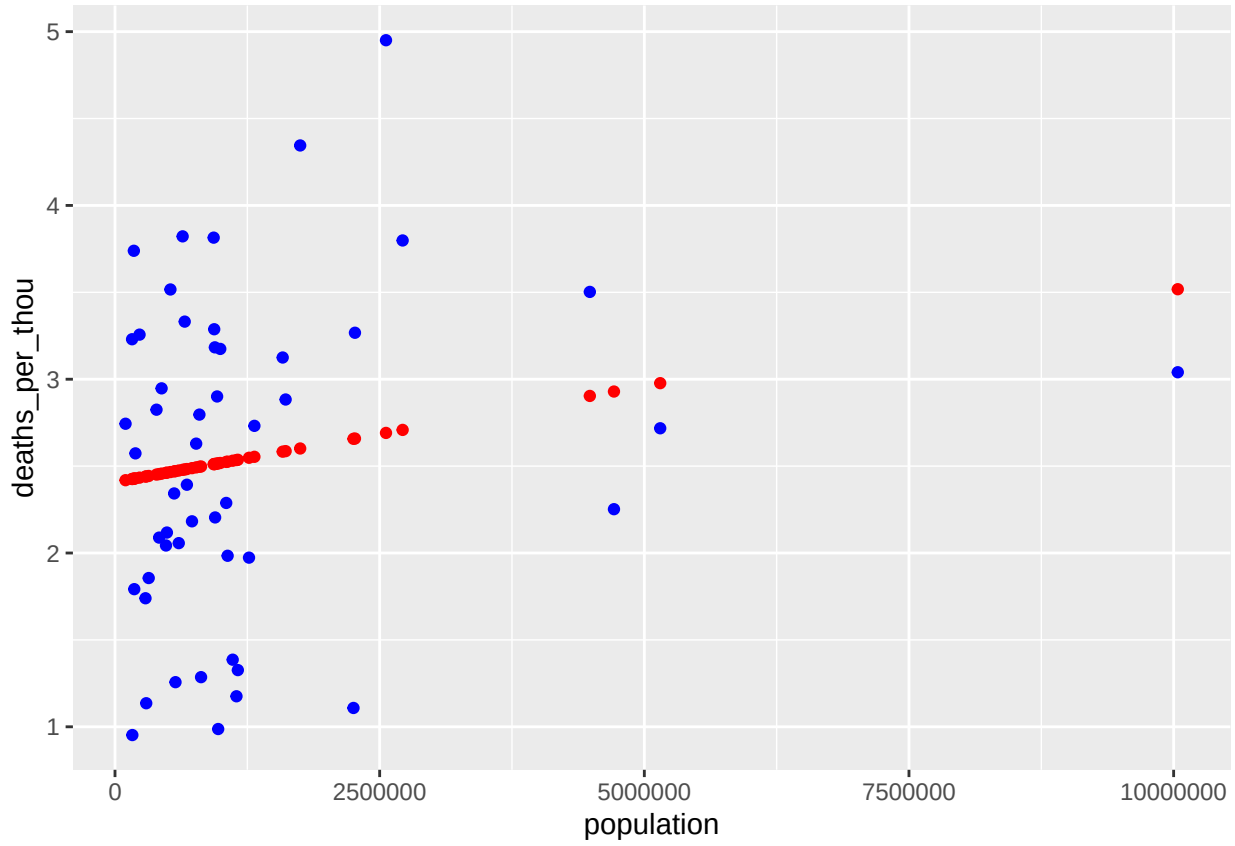
- Our model does not look very flexible as it sits.
- Let's try a population as our predicting variable.

```
mod2 <- lm(deaths_per_thou ~ population, data = US_state_totals)
summary(mod2)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ population, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.54863 -0.58090 -0.09025  0.63260  2.25958
##
## Coefficients:
##                  Estimate    Std. Error t value            Pr(>|t|)
## (Intercept) 2.40796930769 0.16171374467  14.890 <0.0000000000000002 ***
## population  0.00000011056 0.00000007711   1.434               0.158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9121 on 49 degrees of freedom
## Multiple R-squared:  0.04027,    Adjusted R-squared:  0.02068
## F-statistic: 2.056 on 1 and 49 DF,  p-value: 0.158
```

- Using population as our predictor variable gives a higher error and a lower R^2 value.
- And if we plot it:

```
## Add a column of predictions from the model in our US state totals dataframe: (at this time our model
# predicting deaths per thousand using just one independent variable =population)
US_state_totals<- US_state_totals %>%
  mutate(pred = predict(mod2))

## Lets plot out our predictions against actual values and see how it looks
US_state_totals %>% ggplot() +
  geom_point(aes(x = population, y = deaths_per_thou), color = "blue") + # Actual values in blue
  geom_point(aes(x = population, y = pred), color = "red") # Predicted values in red
```



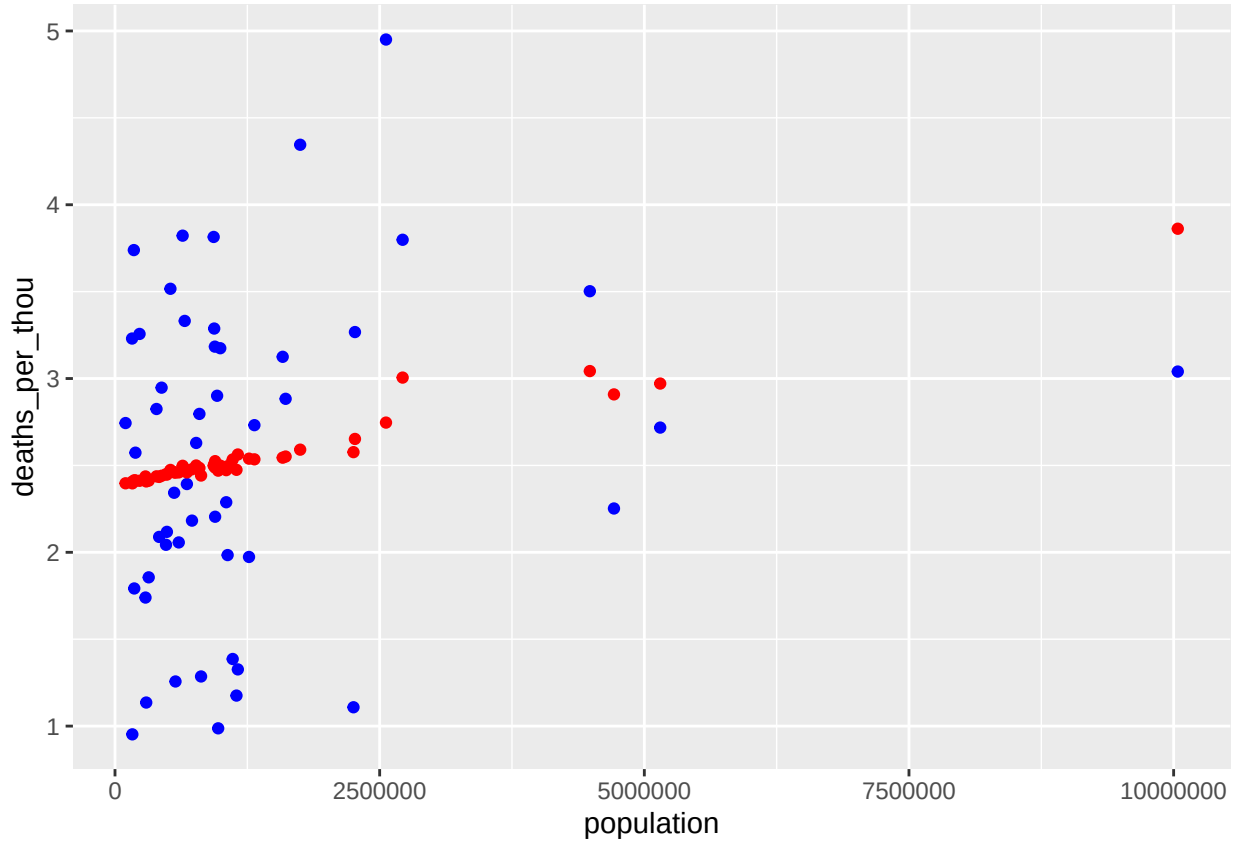
- What if we did multiple predictors to try and predict deaths per thousand?

```
mod3 <- lm(deaths_per_thou ~ cases^2-(population), data = US_state_totals)
summary(mod3)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases^2 - (population), data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.48275 -0.59398 -0.06709  0.64531  2.20414
##
## Coefficients:
##                 Estimate   Std. Error t value            Pr(>|t|)
## (Intercept) 2.3834078183 0.1536835746  15.509 <0.0000000000000002 ***
## cases       0.0000005300 0.0000002803   1.891              0.0645 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8988 on 49 degrees of freedom
## Multiple R-squared:  0.06802,    Adjusted R-squared:  0.049
## F-statistic: 3.576 on 1 and 49 DF,  p-value: 0.06453
```

```
US_state_totals<- US_state_totals %>%
  mutate(pred = predict(mod3))

## Lets plot out our predictions against actual values and see how it looks
US_state_totals %>% ggplot() +
  geom_point(aes(x = population, y = deaths_per_thou), color = "blue") + # Actual values in blue
  geom_point(aes(x = population, y = pred), color = "red") # Predicted values in red
```



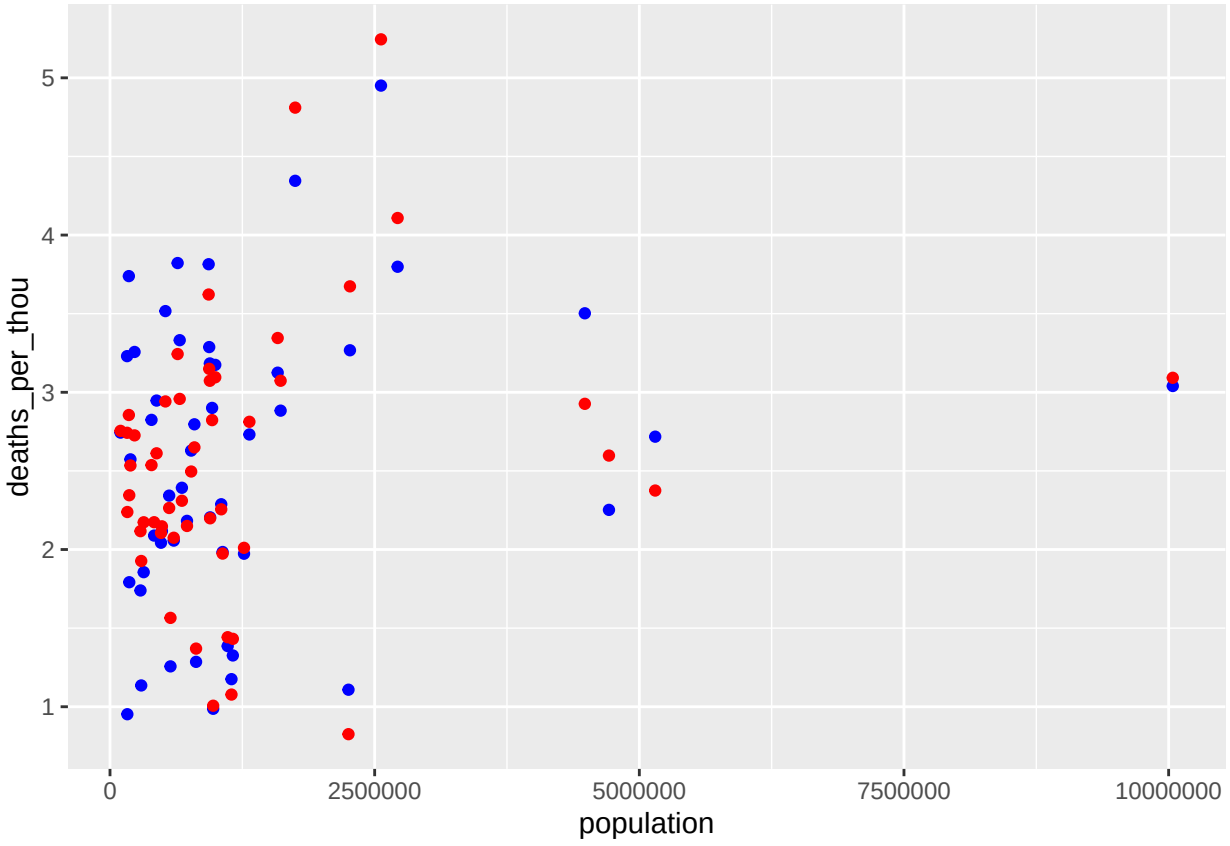
- Now to have some fun with building an accurate but missleading model:

```
mod4 <- lm(deaths_per_thou ~ deaths^2*cases*sqrt(population), data = US_state_totals)
summary(mod4)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ deaths^2 * cases * sqrt(population),
##     data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28616 -0.14799  0.00603  0.14256  0.88330
##
## Coefficients:
##                                         Estimate           Std. Error t value
## (Intercept)                      3.62606815144891836  0.27116155029175015  13.372
## deaths                           0.00169184591438481  0.00022114004366192   7.651
## cases                           -0.00000349225211333  0.00000364131563491  -0.959
## sqrt(population)                -0.00389168009633274  0.00062331434390846  -6.244
## deaths:cases                    -0.00000000085090679  0.00000000024641018  -3.453
## deaths:sqrt(population)         -0.00000061277994918  0.00000016179887731  -3.787
## cases:sqrt(population)           0.00000000526880287  0.00000000202980405   2.596
## deaths:cases:sqrt(population)    0.00000000000020402  0.00000000000005749   3.549
##                                        Pr(>|t|)
## (Intercept)                    < 0.0000000000000002 ***
## deaths                                0.00000000149 ***
## cases                                     0.342889
## sqrt(population)                      0.00000016169 ***
## deaths:cases                              0.001256 **
## deaths:sqrt(population)                   0.000468 ***
## cases:sqrt(population)                    0.012863 *
## deaths:cases:sqrt(population)             0.000950 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3951 on 43 degrees of freedom
## Multiple R-squared:  0.842,  Adjusted R-squared:  0.8162
## F-statistic: 32.73 on 7 and 43 DF,  p-value: 0.00000000000000309
```

- That standard error is much lower than the previous models and our R^2 is pretty high!
- It is a ridiculous coeffiecent I concocted to achieve this and it would not be good to do in the real world.

```
US_state_totals<- US_state_totals %>%
  mutate(pred = predict(mod4))

## Lets plot out our predictions against actual values and see how it looks
US_state_totals %>% ggplot() +
  geom_point(aes(x = population, y = deaths_per_thou), color = "blue") + # Actual values in blue
  geom_point(aes(x = population, y = pred), color = "red") # Predicted values in red
```



Conclusion

- Through analysis I found that the deaths and cases are diverging as time goes on, the virus is becoming less lethal for I am sure many reasons.
- Many regions are having widely varied data right now. This could be because of reporting processes varying by region or merely how they are responding to the pandemic.
- The model above is actually a fluke. It predicts quite accurately but to be honest I tried many combinations of those variables to make it look like that only to make the point that you can misslead yourself or others by manipulating the data with formulas and including variables that are actually causal.
- How to actually improve the model?
- What if we added data like population density etc? I wonder if population density would affect the deaths per thousand.
- I will save this for a later date but I am sure with the right variables we could predict very accurately.

Thank you for reading and I hope you stay safe out there!