COVID-19 Data Project

Ryan Scott

2025-06-23

R Markdown

Statement of Interest

The following project is an Exploratory Data Analysis of COVID-19.

Background of Source Data

The data set for this project was obtained from the John Hopkins website.

We will be using the tidyverse library for this project.

```
#install.packages("tidyverse")
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'purrr' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

I assign each csv file to a variable.

```
global_cases <- read_csv("~/Desktop/Documents/Data Analysis Tools/Projects/RStudio Projects/COVID/time_s
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
global_deaths <- read_csv("~/Desktop/Documents/Data Analysis Tools/Projects/RStudio Projects/COVID/time_
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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_cases <- read_csv("~/Desktop/Documents/Data Analysis Tools/Projects/RStudio Projects/COVID/time_serie
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): 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("~/Desktop/Documents/Data Analysis Tools/Projects/RStudio Projects/COVID/time_seri
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

I first want to look the columns for global cases and global deaths. I use the head function to see the column names with a set number of rows.

```
head(global_cases, n = 5)
```

```
## # A tibble: 5 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
## # 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>,
## #   '2/17/20' <dbl>, '2/18/20' <dbl>, '2/19/20' <dbl>, '2/20/20' <dbl>, ...
```

```
head(global_deaths, n=5)
```

```
## # A tibble: 5 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
## # 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>,
## #   '2/17/20' <dbl>, '2/18/20' <dbl>, '2/19/20' <dbl>, '2/20/20' <dbl>, ...
```

This code reshapes the global cases data set by converting daily case columns into a longer, tidy format where each row represents a specific location and date. It removes latitude and longitude columns, streamlining the data for easier analysis or visualization.

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

This code reshapes the global deaths data set by converting daily case columns into a longer, tidy format where each row represents a specific location and date. It removes latitude and longitude columns, streamlining the data for easier analysis or visualization.

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

The following code performs a full join to keep all records from both data tables, renames the country and province columns for consistency, and converts the date column from character format to a proper date format using mdy() (month-day-year).

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

```
## Joining with `by = join_by(`Province/State`, `Country/Region`, date)`
```

We use summary() to give us a quick statistical overview of the global data table

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:330327      Length:330327      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
##      deaths
##  Min.   :      0
##  1st Qu.:      3
##  Median :    150
##  Mean   :  13380
##  3rd Qu.:   3032
##  Max.   :1123836
```

We filter the global table to give us countries where cases are greater than 0.

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

We use the head function to view the columns in the US cases data set.

```
head(US_cases, n =5)
```

```
## # A tibble: 5 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
## # 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>,
## #   `2/12/20` <dbl>, `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, ...
```

This code reshapes the US cases data set by converting daily case columns into a longer, tidy format where each row represents a specific location and date. It removes latitude and longitude columns, streamlining the data for easier analysis or visualization.

```
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_))
```

```
head(US_deaths, n = 5)
```

```
## # A tibble: 5 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
## # 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>,
## #   `2/11/20` <dbl>, `2/12/20` <dbl>, `2/13/20` <dbl>, `2/14/20` <dbl>, ...
```

This code reshapes the US Deaths data set by converting daily case columns into a longer, tidy format where each row represents a specific location and date. It removes latitude and longitude columns, streamlining the data for easier analysis or visualization.

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

The following code performs a full join to keep all records from both data tables

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

```
## Joining with `by = join_by(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/U
```

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

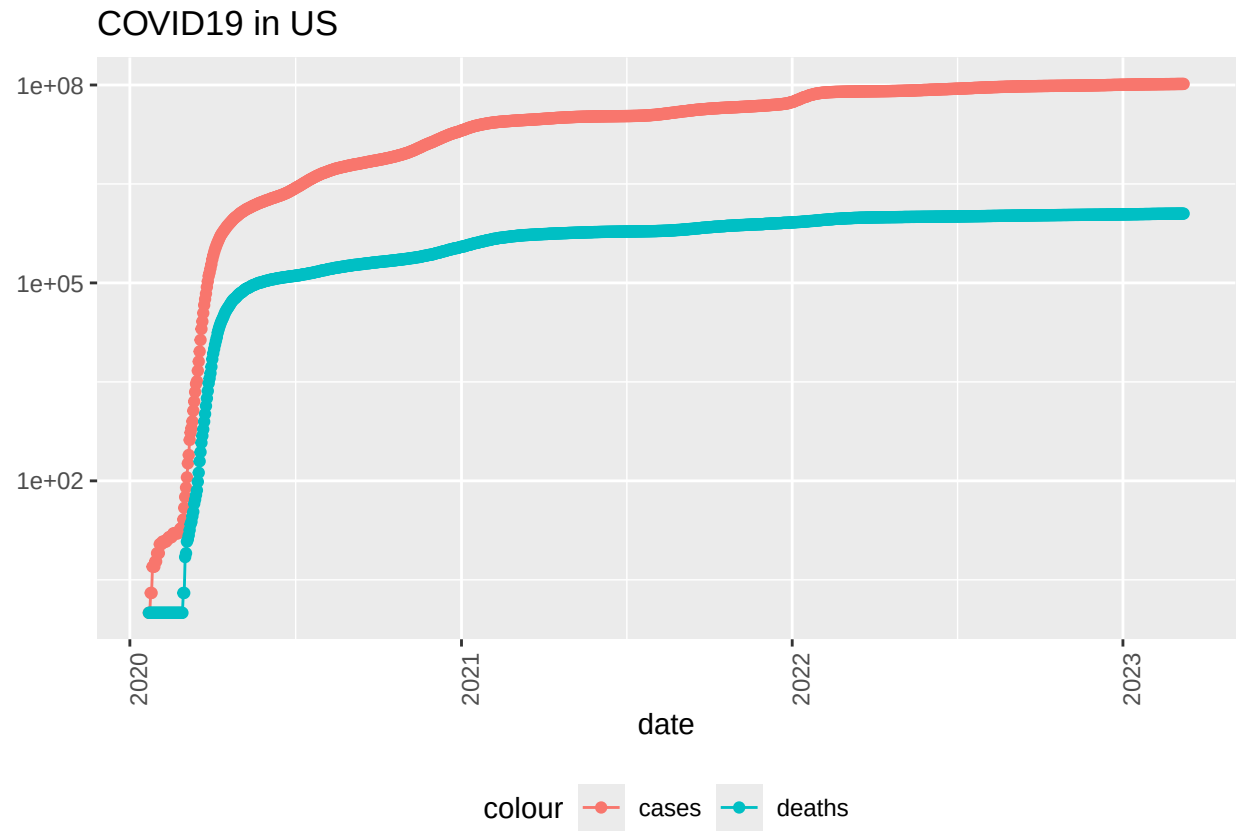
```
US_by_state <- US %>%
    group_by(Province_State, Country_Region, date) %>%
    summarize(cases = sum(cases), deaths = sum(deaths),
              Population = sum(Population))  %>%
    mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
    select(Province_State, Country_Region, date, cases,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

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

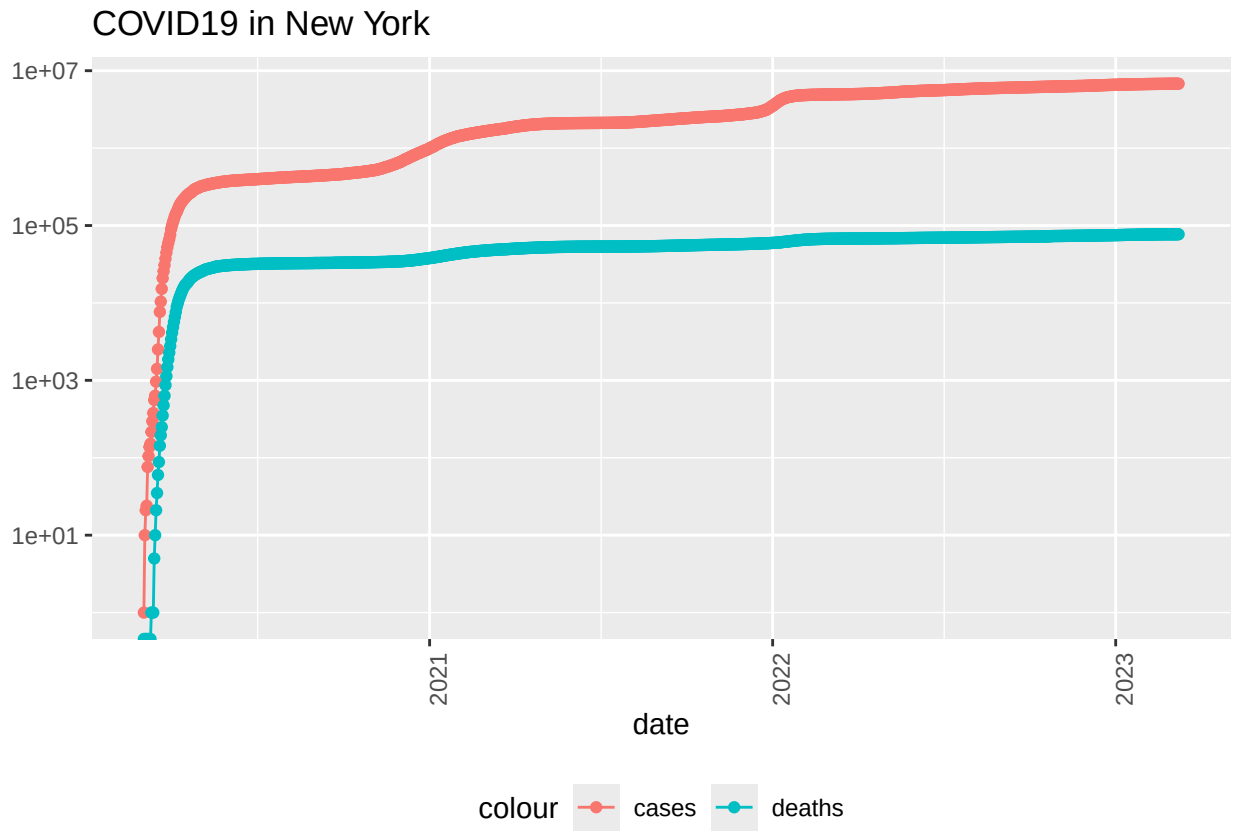
```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

```
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)
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```




```
max(US_totals$date)
```

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

```
max(US_totals$deaths)
```

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

Building new columns to determine the number of new cases each 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))
```

```
tail(US_totals)
```

```
## # A tibble: 6 x 8
##   Country_Region date          cases deaths deaths_per_mill Population new_cases
##   <chr>          <date>        <dbl>  <dbl>           <dbl>      <dbl>     <dbl>
## 1 US             2023-03-04   1.04e8 1.12e6           3371.  332875137      2147
## 2 US             2023-03-05   1.04e8 1.12e6           3371.  332875137     -3862
## 3 US             2023-03-06   1.04e8 1.12e6           3371.  332875137      8564
## 4 US             2023-03-07   1.04e8 1.12e6           3372.  332875137     35371
## 5 US             2023-03-08   1.04e8 1.12e6           3374.  332875137     64861
## 6 US             2023-03-09   1.04e8 1.12e6           3376.  332875137     46931
## # i 1 more variable: new_deaths <dbl>
```

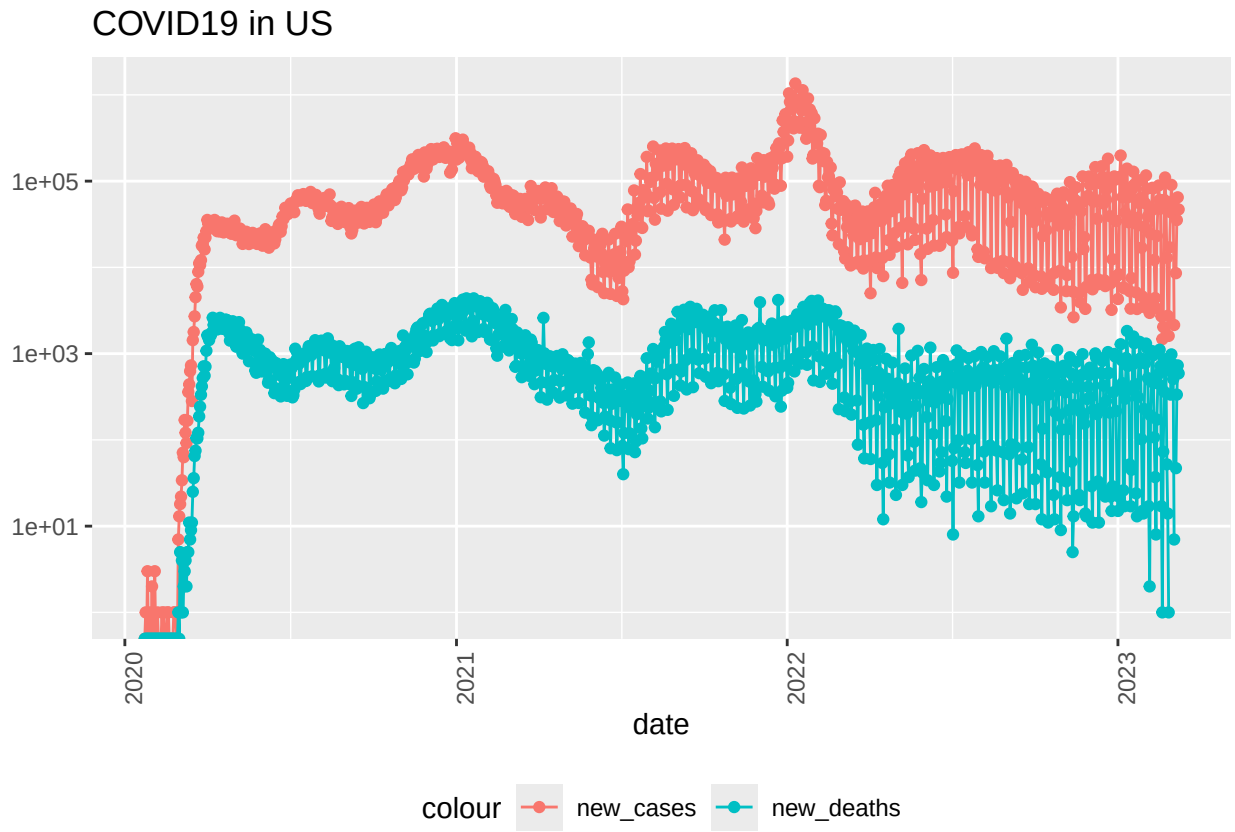
```
tail(US_totals %>% select(new_cases, new_deaths, everything()))
```

```
## # A tibble: 6 x 8
##   new_cases new_deaths Country_Region date          cases deaths deaths_per_mill
##       <dbl>      <dbl> <chr>          <date>        <dbl>  <dbl>           <dbl>
## 1      2147          7 US             2023-03-04   1.04e8 1.12e6           3371.
## 2     -3862        -38 US             2023-03-05   1.04e8 1.12e6           3371.
## 3      8564         47 US             2023-03-06   1.04e8 1.12e6           3371.
## 4     35371        335 US             2023-03-07   1.04e8 1.12e6           3372.
## 5     64861        730 US             2023-03-08   1.04e8 1.12e6           3374.
## 6     46931        590 US             2023-03-09   1.04e8 1.12e6           3376.
## # i 1 more variable: Population <dbl>
```

```
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)
```

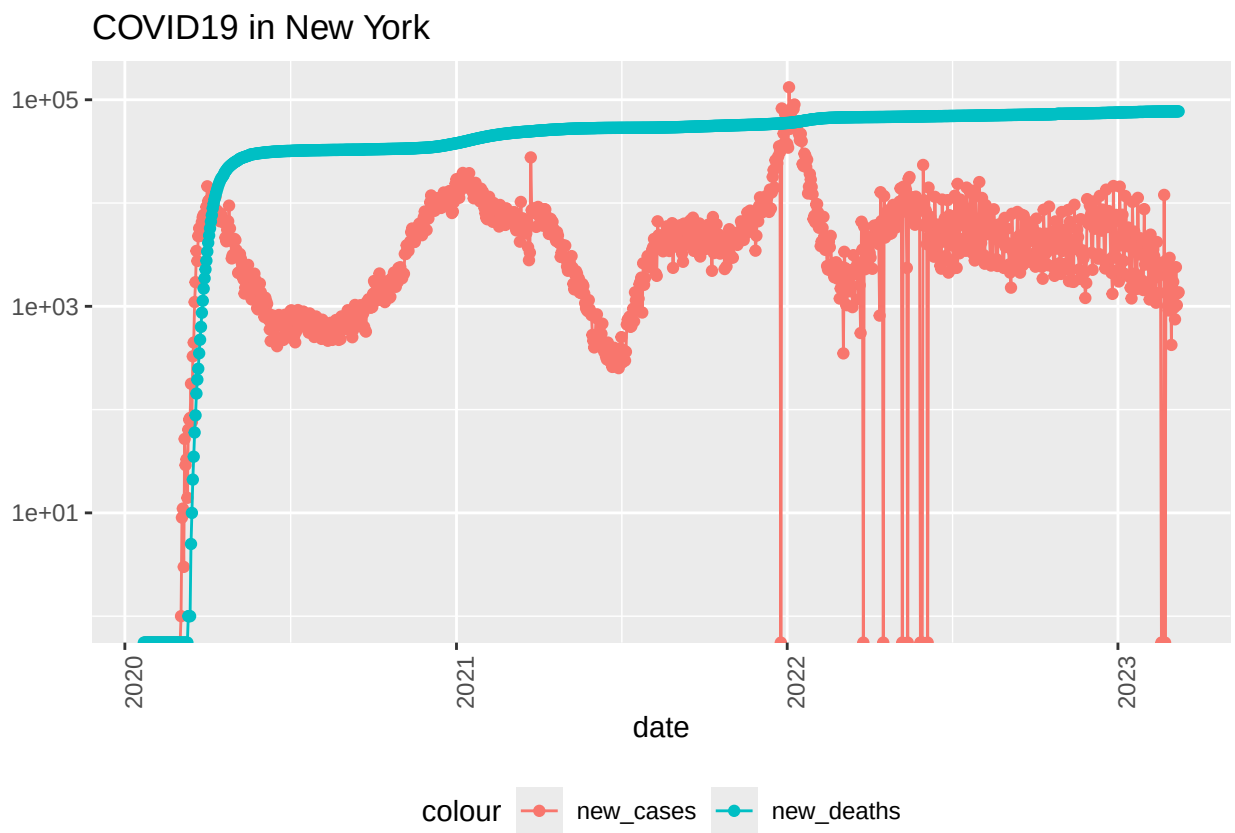
```
## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
state <- "New York"
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 = 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 = str_c("COVID19 in ", state), y = NULL)
```

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

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
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)
```

Tells me the the states with the smallest states of deaths per thousand

```
US_state_totals %>%
    slice_min(deaths_per_thou, n = 10)
```

```
## # A tibble: 10 x 6
##    Province_State        deaths  cases population cases_per_thou deaths_per_thou
##    <chr>                  <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
##  1 American Samoa            34 8.32e3      55641           150.           0.611
##  2 Northern Mariana Isl~     41 1.37e4      55144           248.           0.744
##  3 Virgin Islands           130 2.48e4     107268           231.           1.21
##  4 Hawaii                  1841 3.81e5    1415872           269.           1.30
##  5 Vermont                  929 1.53e5     623989           245.           1.49
##  6 Puerto Rico             5823 1.10e6    3754939           293.           1.55
##  7 Utah                    5298 1.09e6    3205958           340.           1.65
##  8 Alaska                  1486 3.08e5     740995           415.           2.01
##  9 District of Columbia    1432 1.78e5     705749           252.           2.03
## 10 Washington             15683 1.93e6    7614893           253.           2.06
```

```
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           0.611           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
```

Worst states

```
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            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
```

```
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
## -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
```

```
US_state_totals %>% slice_min(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 American Samoa     34  8320      55641           150.           0.611
```

```
US_state_totals %>% slice_max(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths  cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 Rhode Island     3870 460697    1059361           435.            3.65
```

```
US_state_totals %>% mutate(pred = predict(mod))
```

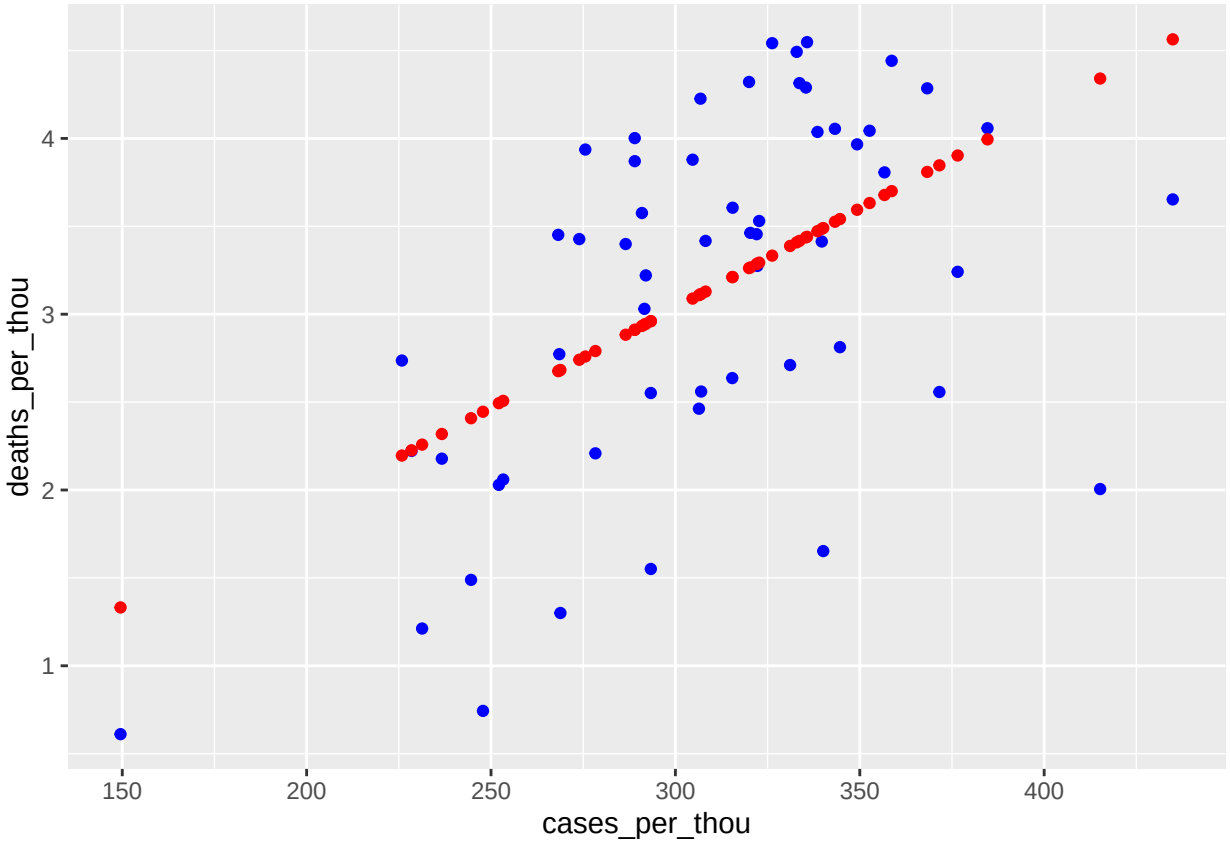
```
## # A tibble: 56 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          21032 1.64e6    4903185           335.            4.29   3.44
##  2 Alaska            1486 3.08e5     740995           415.            2.01   4.34
##  3 American Samoa      34 8.32e3      55641           150.            0.611  1.33
##  4 Arizona          33102 2.44e6    7278717           336.            4.55   3.44
##  5 Arkansas         13020 1.01e6    3017804           334.            4.31   3.42
```

```
##  6 California      101159 1.21e7   39512223           307.           2.56   3.12
##  7 Colorado         14181 1.76e6    5758736           306.           2.46   3.11
##  8 Connecticut      12220 9.77e5    3565287           274.           3.43   2.74
##  9 Delaware          3324 3.31e5     973764           340.           3.41   3.49
## 10 District of Co~   1432 1.78e5     705749           252.           2.03   2.49
## # i 46 more rows
```

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

```
## # A tibble: 56 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          21032 1.64e6    4903185           335.            4.29   3.44
##  2 Alaska            1486 3.08e5     740995           415.            2.01   4.34
##  3 American Samoa      34 8.32e3      55641           150.            0.611  1.33
##  4 Arizona          33102 2.44e6    7278717           336.            4.55   3.44
##  5 Arkansas         13020 1.01e6    3017804           334.            4.31   3.42
##  6 California      101159 1.21e7   39512223           307.            2.56   3.12
##  7 Colorado         14181 1.76e6    5758736           306.            2.46   3.11
##  8 Connecticut      12220 9.77e5    3565287           274.            3.43   2.74
##  9 Delaware          3324 3.31e5     973764           340.            3.41   3.49
## 10 District of Co~   1432 1.78e5     705749           252.            2.03   2.49
## # i 46 more rows
```

```
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")
```


deaths_per_thou
4
3
2
1
150
200
250
300
350
400
cases_per_thou