COVID-19 Data Report

H. Chopra

2022-11-12

1) Importing Data

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

```
library(tidyverse)
library(lubridate)
url_in =
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti

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)

global_cases = read_csv(urls[2])
global_deaths = read_csv(urls[4])
US_cases = read_csv(urls[1])
US_deaths = read_csv(urls[3])
```

2) Tidying and Transforming Data

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

```
global_cases = global_cases %>%
  pivot_longer(cols = -c('Province/State','Country/Region', Lat, Long),names_to = "date",values_to = "ca
  select(-c(Lat,Long))
global_deaths = global_deaths %>%
  pivot_longer(cols = -c('Province/State','Country/Region', Lat, Long),names_to = "date",values_to = "de
  select(-c(Lat,Long))
global = global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region',
         Province_State  = 'Province/State') %>%
  mutate(date = mdy(date))
global
```

```
## # A tibble: 301,138 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-01-22     0      0
##  2 <NA>           Afghanistan    2020-01-23     0      0
##  3 <NA>           Afghanistan    2020-01-24     0      0
##  4 <NA>           Afghanistan    2020-01-25     0      0
##  5 <NA>           Afghanistan    2020-01-26     0      0
##  6 <NA>           Afghanistan    2020-01-27     0      0
##  7 <NA>           Afghanistan    2020-01-28     0      0
##  8 <NA>           Afghanistan    2020-01-29     0      0
##  9 <NA>           Afghanistan    2020-01-30     0      0
## 10 <NA>           Afghanistan    2020-01-31     0      0
## # ... with 301,128 more rows
```

Now we want to look at a summary of the data to see if there are any problems.

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:301138      Length:301138      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-10-08   1st Qu.:     502
##  Mode  :character   Mode  :character   Median :2021-06-25   Median :   11511
##                                        Mean   :2021-06-25   Mean   :  829339
##                                        3rd Qu.:2022-03-13   3rd Qu.:  191997
##                                        Max.   :2022-11-28   Max.   :98628566
##      deaths
##  Min.   :      0
##  1st Qu.:      3
##  Median :    125
##  Mean   :  12407
##  3rd Qu.:   2645
##  Max.   :1079477
```

I will filter out and keep only cases that are positive. After that we will again look at the summary.

```
global = global %>%
  filter(cases>0)
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:277840      Length:277840      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-11-16   1st Qu.:    1021
##  Mode  :character   Mode  :character   Median :2021-07-26   Median :   16626
##                                        Mean   :2021-07-22   Mean   :  898883
##                                        3rd Qu.:2022-03-31   3rd Qu.:  235485
##                                        Max.   :2022-11-28   Max.   :98628566
##      deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    185
##  Mean   :  13448
##  3rd Qu.:   3200
##  Max.   :1079477
```

I want to see cases that are bigger than 90000000.

```
global %>%
  filter(cases>90000000)
```

```
## # A tibble: 132 x 5
##    Province_State Country_Region date          cases  deaths
##    <chr>          <chr>          <date>        <dbl>   <dbl>
##  1 <NA>           US             2022-07-20 90078062 1026176
##  2 <NA>           US             2022-07-21 90253512 1026673
##  3 <NA>           US             2022-07-22 90384904 1027254
##  4 <NA>           US             2022-07-23 90411234 1027325
##  5 <NA>           US             2022-07-24 90434482 1027356
##  6 <NA>           US             2022-07-25 90624438 1027782
##  7 <NA>           US             2022-07-26 90756647 1028332
##  8 <NA>           US             2022-07-27 90996776 1029287
##  9 <NA>           US             2022-07-28 91173824 1029689
## 10 <NA>           US             2022-07-29 91333067 1030314
## # ... with 122 more rows
```

This shows data starting from July 2022.

Moving over to US data sets, I will tidy and transform in the same manner.

```
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 = US_cases %>%
  full_join(US_deaths) %>%
  filter(cases>0)
```

We're going to do comparative analysis between the data. First we will combine the data in a combined key.

```
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))
```

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

```
## # A tibble: 277,840 x 6
##    Province_State Country_Region date       cases deaths Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0 Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0 Afghanistan
##  3 <NA>           Afghanistan    2020-02-26     5      0 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0 Afghanistan
## # ... with 277,830 more rows
```

3) Visualizing Data

We're going to focus on analyzing US as a whole, and for a given state, to see what sorts of things we might want to do.

I'm going to start with US by state.

```
US_by_state = US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths)) %>%
  mutate(death_rate = deaths/cases) %>%
  select(Province_State, Country_Region, date, cases, deaths, death_rate) %>%
  ungroup()
US_by_state
```

```
## # A tibble: 57,358 x 6
##    Province_State Country_Region date       cases deaths death_rate
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 Alabama        US             2020-03-11     3      0          0
##  2 Alabama        US             2020-03-12     4      0          0
##  3 Alabama        US             2020-03-13     8      0          0
##  4 Alabama        US             2020-03-14    15      0          0
##  5 Alabama        US             2020-03-15    28      0          0
##  6 Alabama        US             2020-03-16    36      0          0
##  7 Alabama        US             2020-03-17    51      0          0
##  8 Alabama        US             2020-03-18    61      0          0
##  9 Alabama        US             2020-03-19    88      0          0
## 10 Alabama        US             2020-03-20   115      0          0
## # ... with 57,348 more rows
```

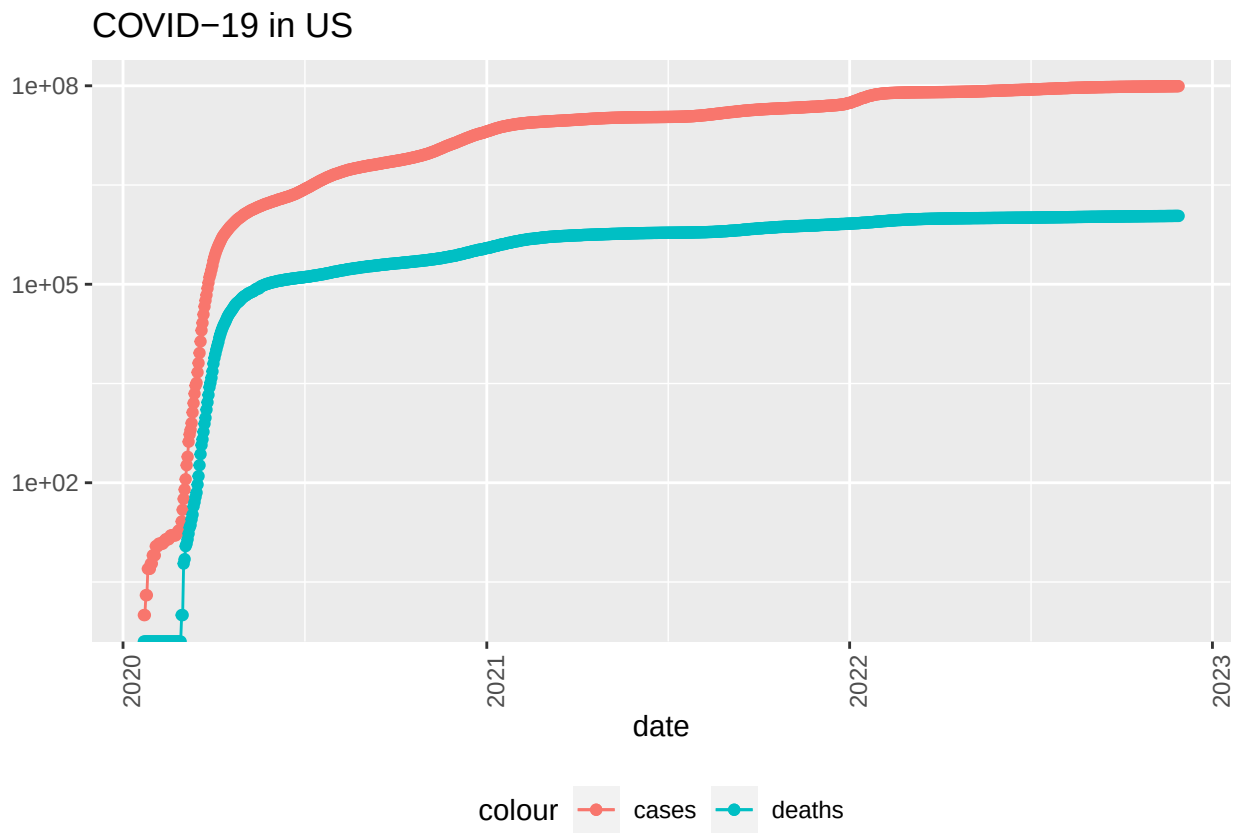
Now I'm going to look at the totals of the US.

```
US_totals = US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths)) %>%
  mutate(death_rate = deaths/cases) %>%
  select(Country_Region, date, cases, deaths, death_rate) %>%
  ungroup()
US_totals
```

```
## # A tibble: 1,042 x 5
##    Country_Region date       cases deaths death_rate
##    <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 US             2020-01-22     1      0          0
##  2 US             2020-01-23     1      0          0
##  3 US             2020-01-24     2      0          0
##  4 US             2020-01-25     2      0          0
##  5 US             2020-01-26     5      0          0
##  6 US             2020-01-27     5      0          0
##  7 US             2020-01-28     5      0          0
##  8 US             2020-01-29     6      0          0
##  9 US             2020-01-30     6      0          0
## 10 US             2020-01-31     8      0          0
## # ... with 1,032 more rows
```

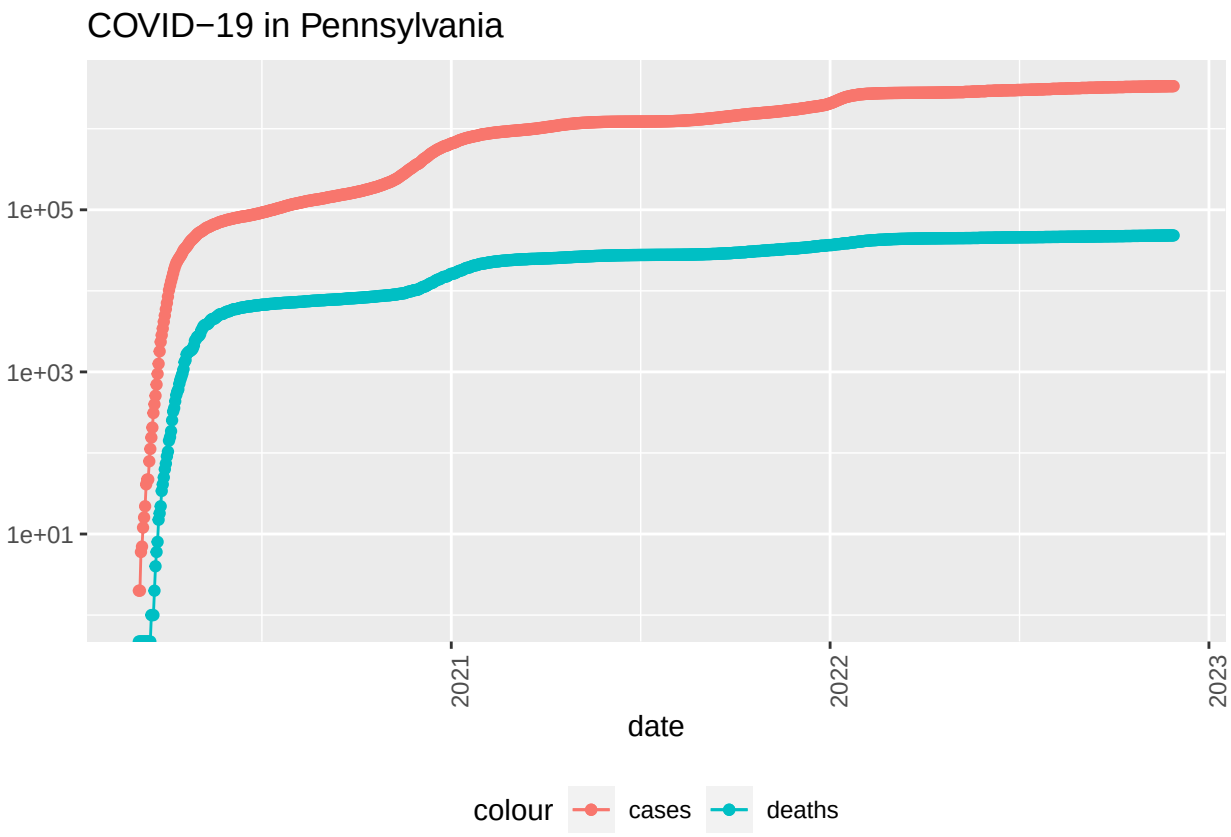
Let's visualize some of this data.

```
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 = "COVID-19 in US", y = NULL)
```



We can do the same thing and analyze cases vs. deaths in the state of Pennsylvania.

```
state = "Pennsylvania"
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 in ", state), y = NULL)
```



We can see that the deaths curve is significantly turning down but follows the same pattern as the cases.

Let's look at what date had the maximum deaths in Pennsylvania and all of the US.

```
max(US_by_state$date)
```

```
## [1] "2022-11-28"
```

```
max(US_by_state$deaths)
```

```
## [1] 97510
```

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

```
## [1] "2022-11-28"
```

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

```
## [1] 1078714
```

I can see the maximum number of death's as of today's days. This raises the question of whether or not the cases have truly leveled off.

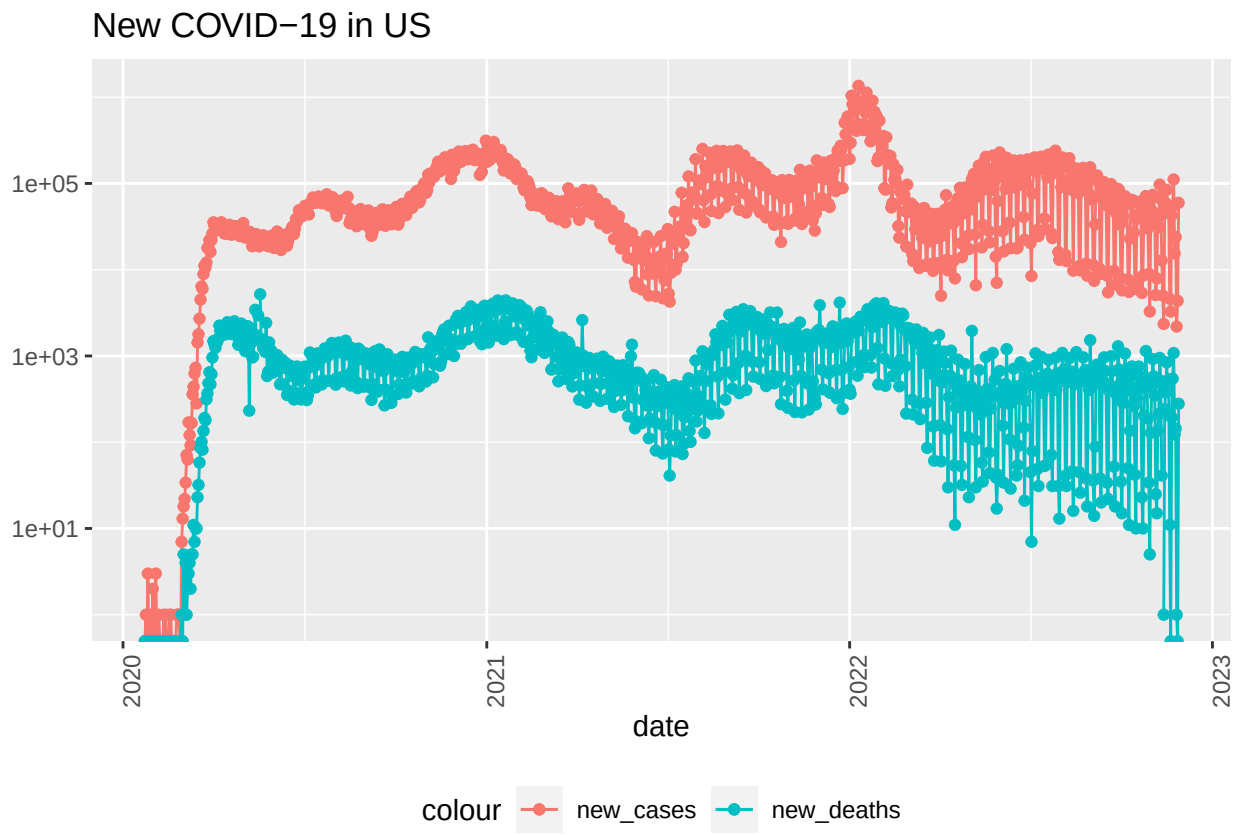
4) Analyzing Data

In order to analyze this, we will add new columns to the existing data sets so that we can see the new cases and new deaths everyday.

```
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))
```

Let's visualize the data once more to see what it does.

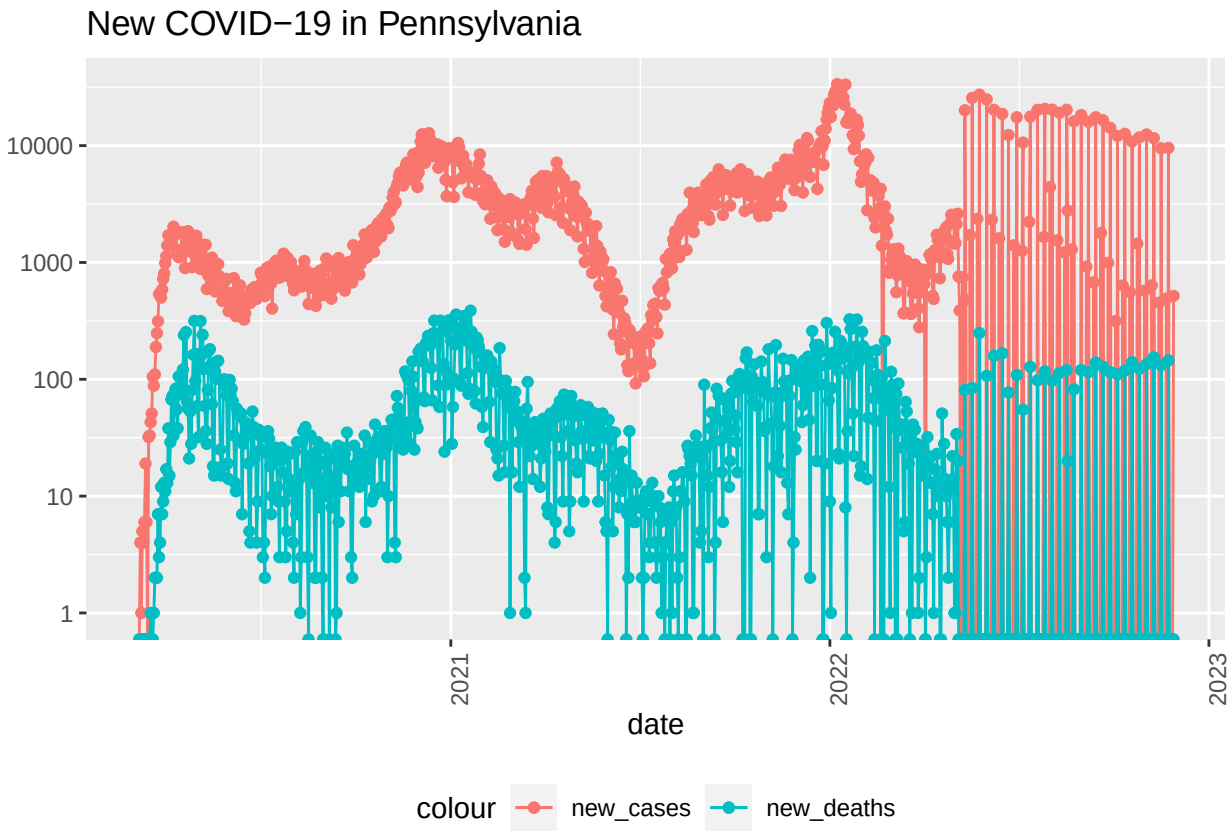
```
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases"))+
  geom_point(aes(color= "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths"))+
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90))+
  labs(title = "New COVID-19 in US", y = NULL)
```



It seems like I still have the same number of new cases per day and the number of new deaths per day. It eventually flattens out but is still up a little bit over what it was before.

Let's see what's happening in Pennsylvania right now.

```
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("New COVID-19 in ", state), y = NULL)
```



It seems like we need to look at another state's data after looking at one state's data. The question raised here is which state is the worst and which is the best?

We'll transform the data once again before we do a little bit of an analysis.

```
US_state_totals = US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases)) %>%
  mutate(death_ratio =deaths/cases ) %>%
  filter(cases >0)
```

```
US_state_totals %>%
  slice_min(death_ratio, n = 10) %>%
  select(Province_State, death_ratio, everything())
```

```
## # A tibble: 10 x 4
##    Province_State           death_ratio deaths   cases
##    <chr>                          <dbl>  <dbl>   <dbl>
##  1 Diamond Princess             0            0      49
##  2 Northern Mariana Islands     0.00310     41   13212
##  3 American Samoa               0.00411     34    8263
##  4 Hawaii                       0.00473   1732  366340
##  5 Alaska                       0.00478   1436  300544
##  6 Utah                         0.00483   5110 1058874
##  7 Vermont                      0.00526    770  146442
##  8 Virgin Islands               0.00527    124   23509
##  9 Puerto Rico                  0.00531   5352 1008104
## 10 Guam                         0.00689    409   59330
```

The worst states with the highest deaths are:

```
US_state_totals %>%
  slice_min(death_ratio, n = 10) %>%
  select(Province_State, death_ratio, everything())
```

```
## # A tibble: 10 x 4
##    Province_State           death_ratio deaths   cases
##    <chr>                          <dbl>  <dbl>   <dbl>
##  1 Diamond Princess             0            0      49
##  2 Northern Mariana Islands     0.00310     41   13212
##  3 American Samoa               0.00411     34    8263
##  4 Hawaii                       0.00473   1732  366340
##  5 Alaska                       0.00478   1436  300544
##  6 Utah                         0.00483   5110 1058874
##  7 Vermont                      0.00526    770  146442
##  8 Virgin Islands               0.00527    124   23509
##  9 Puerto Rico                  0.00531   5352 1008104
## 10 Guam                         0.00689    409   59330
```

We can see which states fair to the worst of all states so far in terms of deaths per 1000.

5) Modelling Data

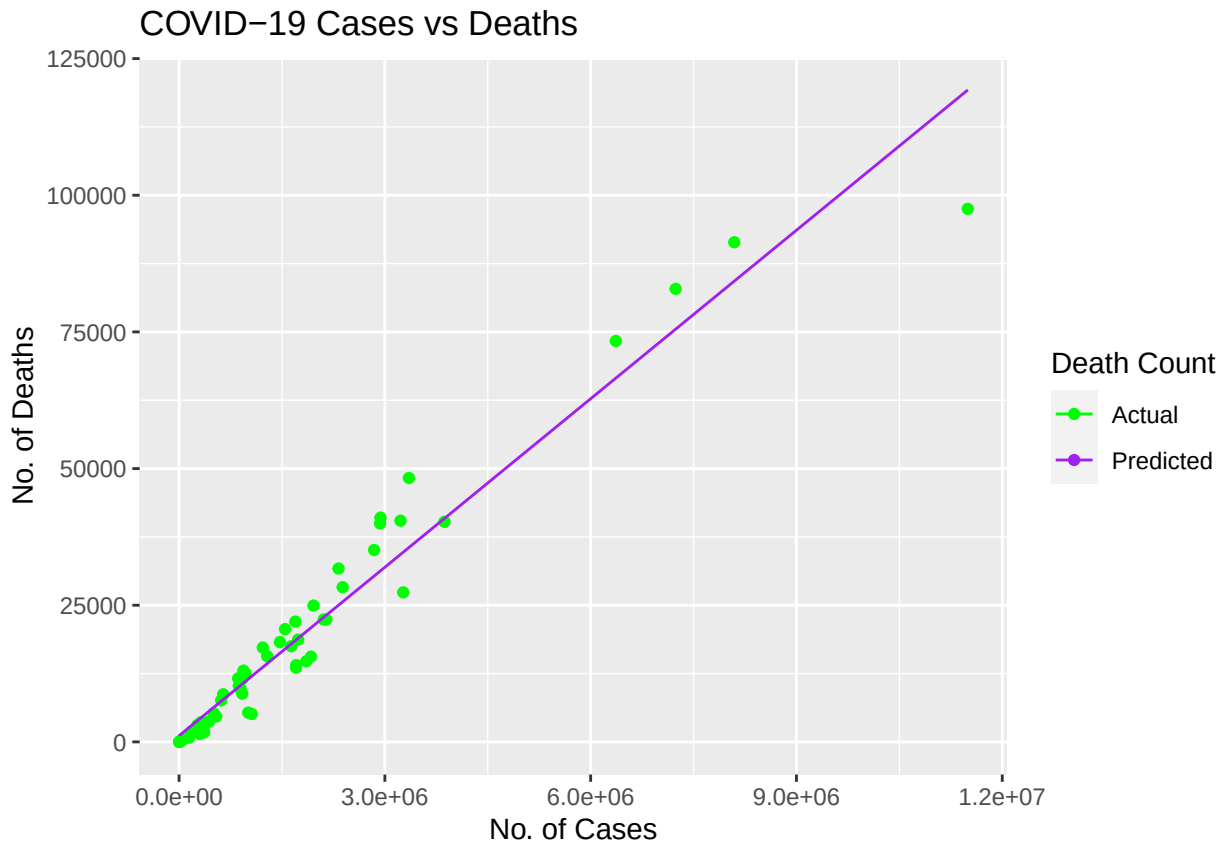
For purposes of our demonstration, we are going to choose to apply a linear model.

```
mod = lm(deaths ~ cases, data = US_state_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths ~ cases, data = US_state_totals)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21758.0  -1755.0   -789.2   2207.6  12718.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.126e+03  8.551e+02   1.317    0.193
## cases       1.027e-02  3.152e-04  32.597   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5074 on 56 degrees of freedom
## Multiple R-squared:  0.9499, Adjusted R-squared:  0.949
## F-statistic:  1063 on 1 and 56 DF,  p-value: < 2.2e-16
```

```
US_tot_w_pred = US_state_totals %>%
  mutate(pred = predict(mod))
US_tot_w_pred %>% ggplot() +
  geom_point(aes(x = cases, y = deaths, color = "Actual")) +
  geom_line(aes(x = cases, y = pred, color = "Predicted"))+
  scale_color_manual(name = "Death Count", values = c("Actual" = "green", "Predicted" = "purple"))+
  xlab("No. of Cases")+
  ylab("No. of Deaths")+
  ggtitle("COVID-19 Cases vs Deaths")
```



We can see that the model does a reasonably good job of predicting deaths at the lower end quite well. Later on, it represents that the number of deaths have decreased however the number of cases were still increasing.

6) Conclusions

a) There has been leveling off of deaths due to to COVID19 in the US.
b) The cases are increasing however the deaths due to COVID-19 are decreasing over the year.

7) Bias

The bias in my analysis could be affected by the fact that the data I am using is old and not updated to exact numbers. The data could also make me interested in looking what is specifically happening with the current state I'm living in. The data provided on a US website might not be accurate in the case of countries that have not reported data in a proper way. I have made my analysis accordance with good research and while implementing inclusion.

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.utf8  LC_CTYPE=English_Canada.utf8
## [3] LC_MONETARY=English_Canada.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.0  timechange_0.1.1 forcats_0.5.2    stringr_1.4.1
##  [5] dplyr_1.0.10     purrr_0.3.4      readr_2.1.3      tidyr_1.2.1
##  [9] tibble_3.1.8     ggplot2_3.3.6    tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.29       utf8_1.2.2
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.17       highr_0.9
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.0.6
## [13] googlesheets4_1.0.1 curl_4.3.2          readxl_1.4.1
## [16] rstudioapi_0.14     rmarkdown_2.17      labeling_0.4.2
## [19] googledrive_2.0.0   bit_4.0.4           munsell_0.5.0
## [22] broom_1.0.1         compiler_4.2.1      modelr_0.1.9
## [25] xfun_0.33           pkgconfig_2.0.3     htmltools_0.5.3
## [28] tidyselect_1.1.2    fansi_1.0.3         crayon_1.5.2
## [31] tzdb_0.3.0          dbplyr_2.2.1        withr_2.5.0
## [34] grid_4.2.1          jsonlite_1.8.2      gtable_0.3.1
## [37] lifecycle_1.0.3     DBI_1.1.3           magrittr_2.0.3
## [40] scales_1.2.1        cli_3.4.1           stringi_1.7.8
## [43] vroom_1.6.0         farver_2.1.1        fs_1.5.2
## [46] xml2_1.3.3          ellipsis_0.3.2      generics_0.1.3
```

```
## [49] vctrs_0.4.2         tools_4.2.1         bit64_4.0.5
## [52] glue_1.6.2          hms_1.1.2           parallel_4.2.1
## [55] fastmap_1.1.0       yaml_2.3.5          colorspace_2.0-3
## [58] gargle_1.2.1        rvest_1.0.3         knitr_1.40
## [61] haven_2.5.1
```