COVID-19 Analysis

Alfonso Gutiérrez

2024-04-28

Introduction

The objective of this analysis is to get some insights about the global pandemic we lived during 2019-2022. We are going to pass trough all the steps of the data analysis process and come up with some conclusions.

The information used for this analysis were took from the John Hopkins University dataset in GitHub.

Libraries

```
library(tidyverse)
```

```
## -- 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   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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
```

```
library(dplyr)
library(lubridate)
library(ggplot2)
```

First Step: Get the data

Let's get the data from our URL

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

Now let's read the data and see what we have.

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

After importing the data and saved it in different datasets, let's dig in and check the general structure of the data in order to tidying it. We'll delete non-necessary variables for our analysis, rename another ones and be sure that the four datasets uses the same lingo.

```
#Tidy up global cases
global_cases <- global_cases %>%
  #mutate(across(-c(`Province/State`, `Country/Region`, Lat, Long), as.numeric)) %>%
  pivot_longer(
    cols = -c(`Province/State`, `Country/Region`, Lat, Long),
    names_to = "date",
    values_to = "cases"
 ) %>%
 select(-c(Lat, Long))

#Tidy up global deaths
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 cases and deaths into dataset "global"
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)`
```

Now that we created the global dataset, let's check some summary data and outliers

```
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
```

```
#After take a look at the summary, it looks like the minimum cases in the dataset is 0 which is weird s
global <- global %>% filter(cases > 0)
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:306827      Length:306827      Min.   :2020-01-22   Min.   :        1
##  Class :character   Class :character   1st Qu.:2020-12-12   1st Qu.:     1316
##  Mode  :character   Mode  :character   Median :2021-09-16   Median :    20365
##                                        Mean   :2021-09-11   Mean   :  1032863
##                                        3rd Qu.:2022-06-15   3rd Qu.:   271281
##                                        Max.   :2023-03-09   Max.   :103802702
##      deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    214
##  Mean   :  14405
##  3rd Qu.:   3665
##  Max.   :1123836
```

```
#I also checked the max cases to avoid any possible error with the data. The max value is the United St
```

Now that we transform and tide the Global dataset, let's do the same for the United States files. Since it have a lot of columns we don't need, let's keep only the ones we want. Same process as the previous files.

```
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_))

#While doing the tide of the deaths archive, I found roughly 3400 dates that failed to pass so I create
US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date), parse_failed = is.na(date)) %>%
  select(-c(Lat, Long_))

#Let's check the parse_failed cases
problem_dates <- filter(US_deaths, parse_failed)
head(problem_dates)
```

```
## # A tibble: 0 x 8
```

```
## # i 8 variables: Admin2 <chr>, Province_State <chr>, Country_Region <chr>,
## #   Combined_Key <chr>, Population <dbl>, date <date>, deaths <dbl>,
## #   parse_failed <lgl>
```

```
#There are missing data in those lines, there's no date so I'll remove this values
US_deaths <- US_deaths %>%
  filter(!is.na(date)) %>%
  select(-c(parse_failed))

#Now let's merge both datasets
US <- US_cases %>%
  full_join(US_deaths)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

Let's create a variable combined_key into the global dataset so we can add population

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

#Retrieve population info
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.
```

```
#Merge with global dataset
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)
global
```

```
## # A tibble: 306,827 x 7
##    Province_State Country_Region date       cases deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341 Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341 Afghanistan
```

```
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341 Afghanistan
## # i 306,817 more rows
```

Visualize the data

Now that we tide up our info and we are sure that there's not outliers or missing datam, let's start with a couple of visualizations and analysis of the info. Of course, we could do tons of different analysis with such a huge info but let's focus on some basics.

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

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

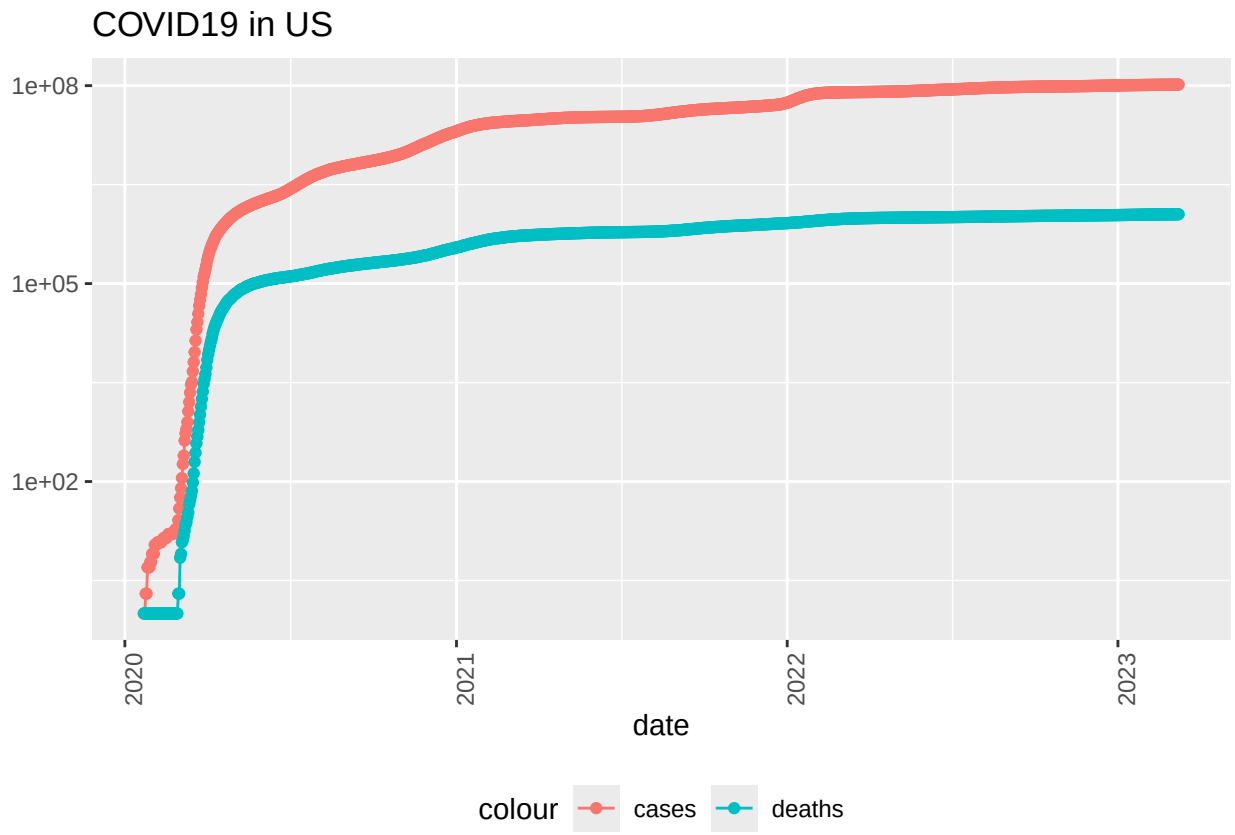
```
#US Totals
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mil = deaths * 1000000 / Population) %>%
  select(Country_Region, date,
         cases, deaths, deaths_per_mil,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

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

Now let's visualize the data

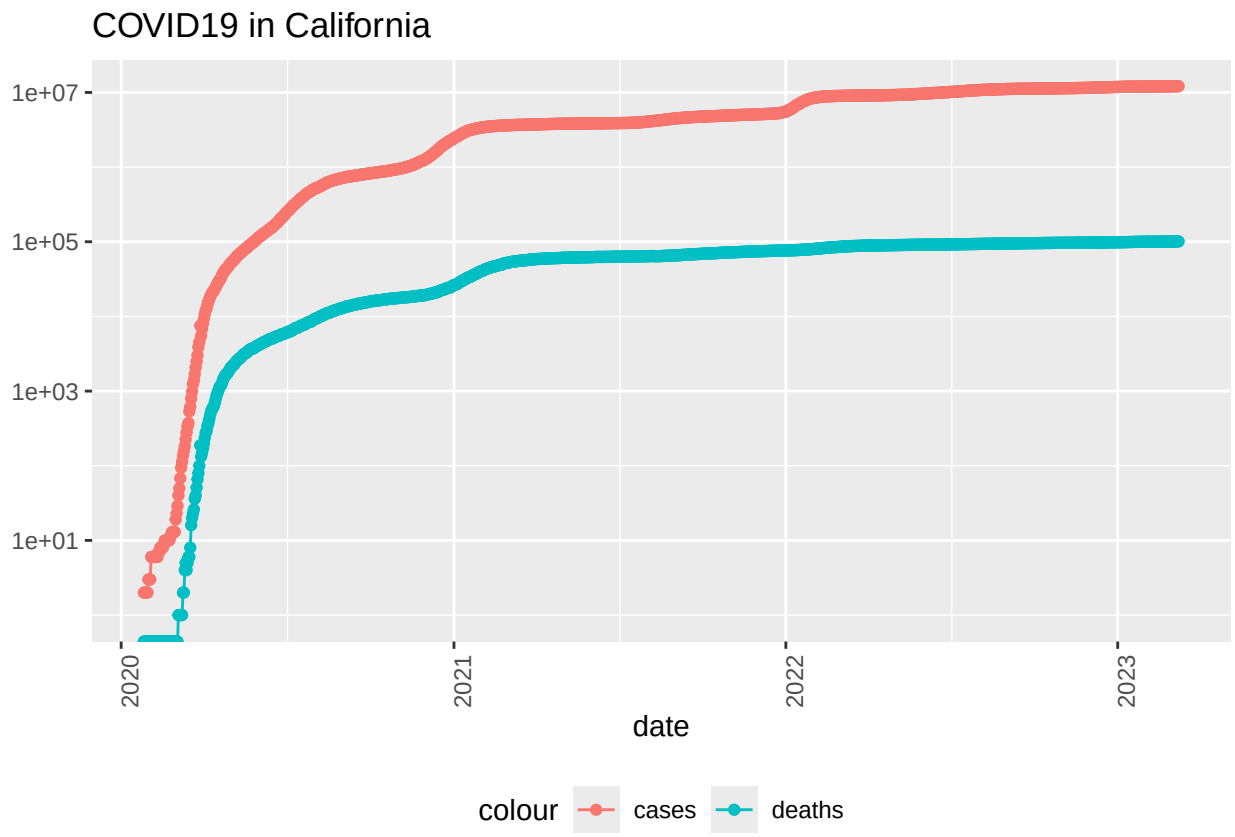
```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
```

```
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```




```
state <- "California"
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.
```



Further analysis

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

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

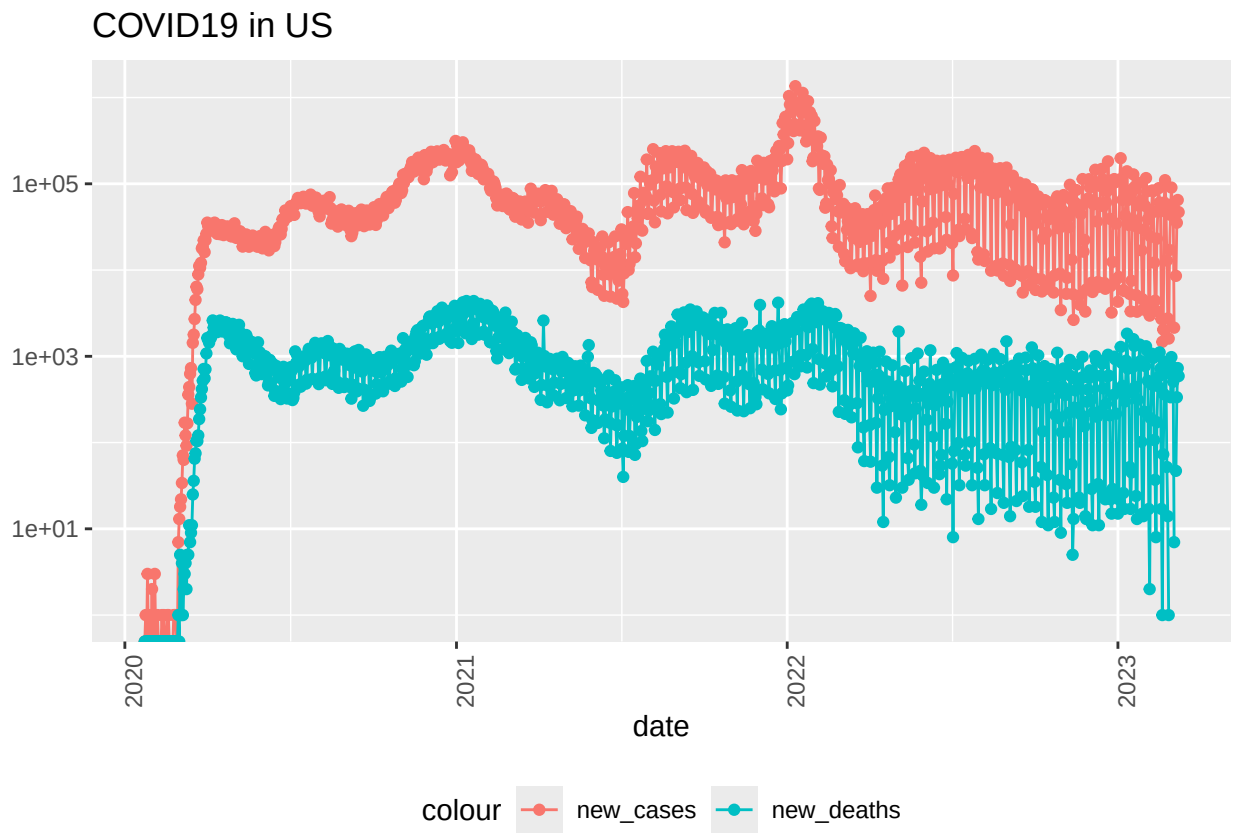
```
max(US_totals$deaths)
```

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

```
#Creating new variables to see only new cases/deaths against the previous 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))

#Visualize new cases and deaths in the US
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)
```

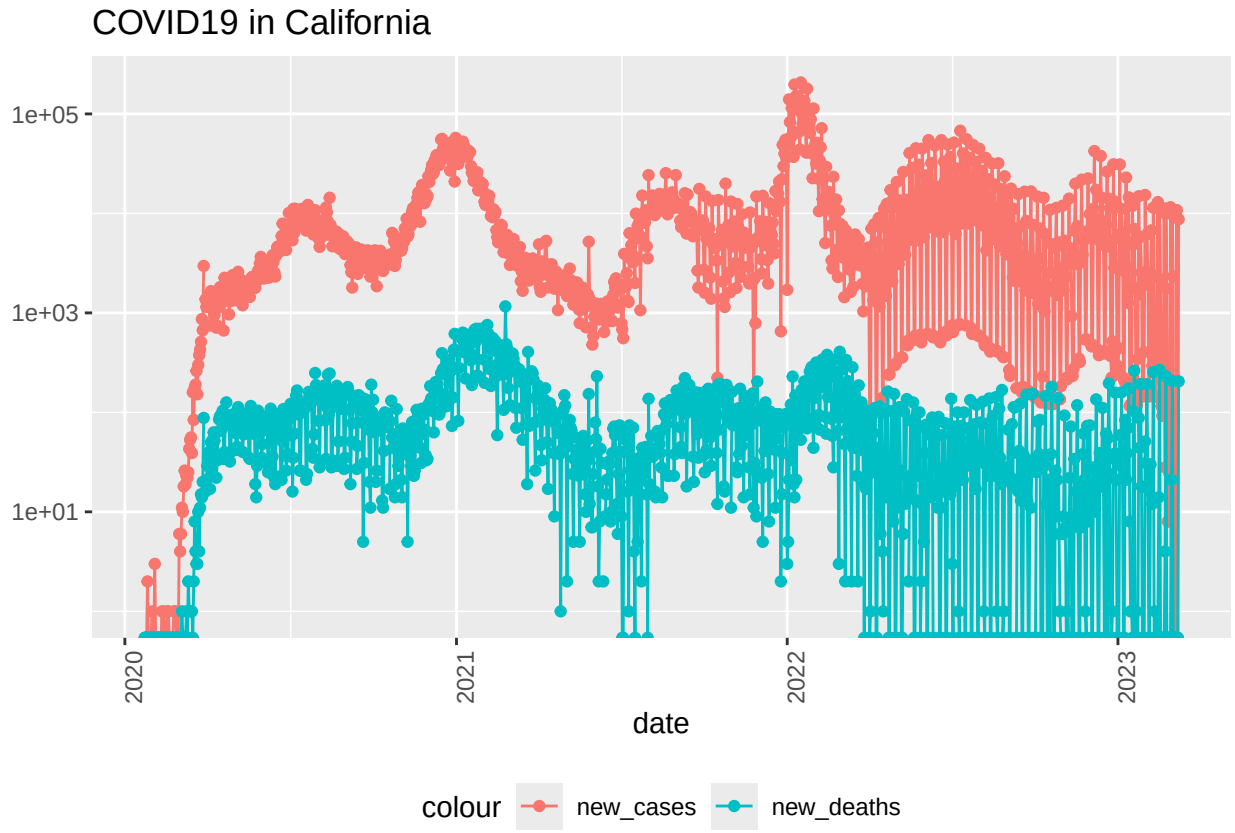
```
## Warning in transformation$transform(x): Se han producido NaNs

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

## Warning in transformation$transform(x): Se han producido NaNs

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

## Warning in transformation$transform(x): Se han producido NaNs

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

## Warning in transformation$transform(x): Se han producido NaNs

## 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()‘).
```




```
#Visualize new cases and deaths in the US
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 = "COVID19 in California", y = NULL)
```

```
## Warning in transformation$transform(x): Se han producido NaNs

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

## Warning in transformation$transform(x): Se han producido NaNs

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

## Warning in transformation$transform(x): Se han producido NaNs

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

```
## Warning in transformation$transform(x): Se han producido NaNs

## 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 3 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 14 rows 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)
```

```r
#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
```

```r
#Best states handling the pandemic
US_state_totals %>%
    slice_min(deaths_per_thou, n = 10) %>%
select(deaths_per_thou, cases_per_thou, everything())
```

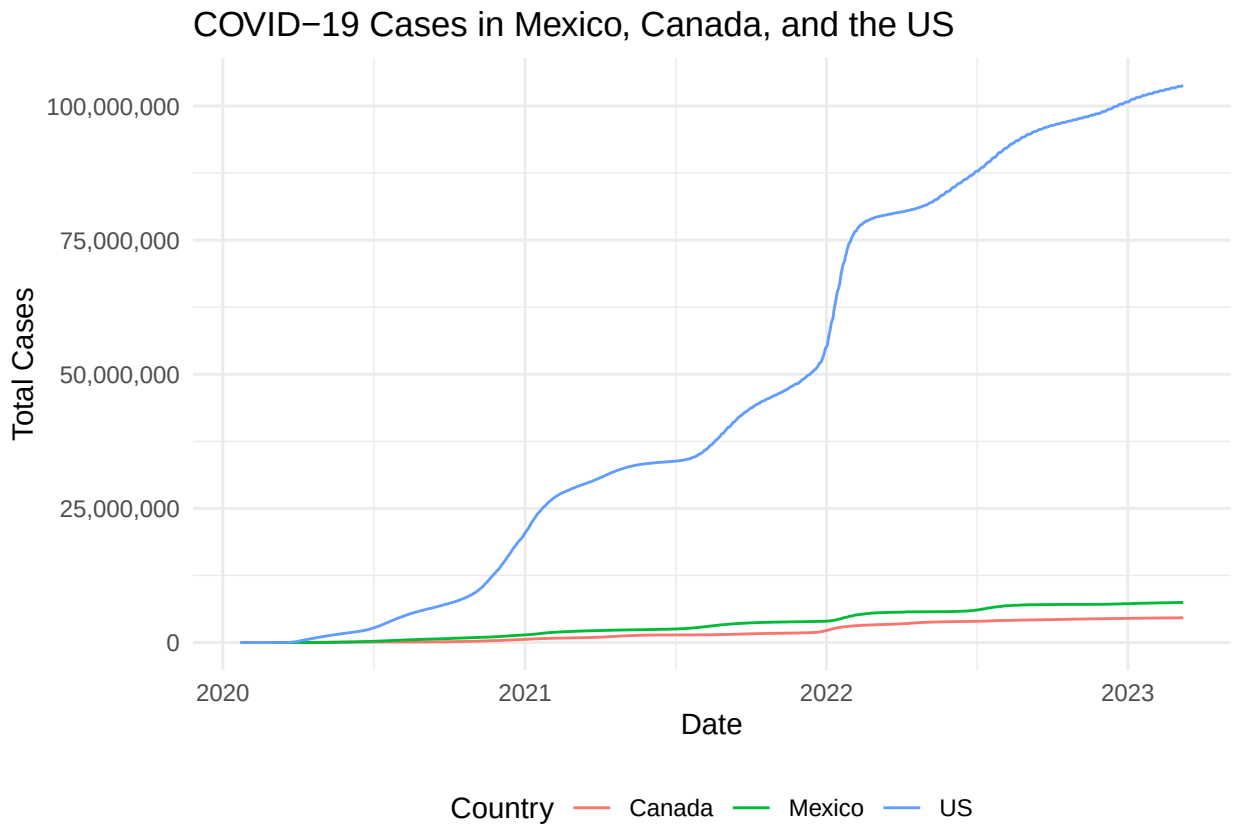
```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State        deaths  cases population
##              <dbl>          <dbl> <chr>                 <dbl>  <dbl>      <dbl>
##  1           0.611           150. American Samoa           34 8.32e3      55641
##  2           0.744           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
```

Let's create another visualization showing the number of cases from North America region (Mexico, US, Canada)

```r
# Filter data for Mexico, Canada, and US
filtered_data <- global %>%
  filter(Country_Region %in% c("Mexico", "Canada", "US"))

# Group by date and country, summarize total cases
summarized_data <- filtered_data %>%
  group_by(date, Country_Region) %>%
  summarise(total_cases = sum(cases))
```

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

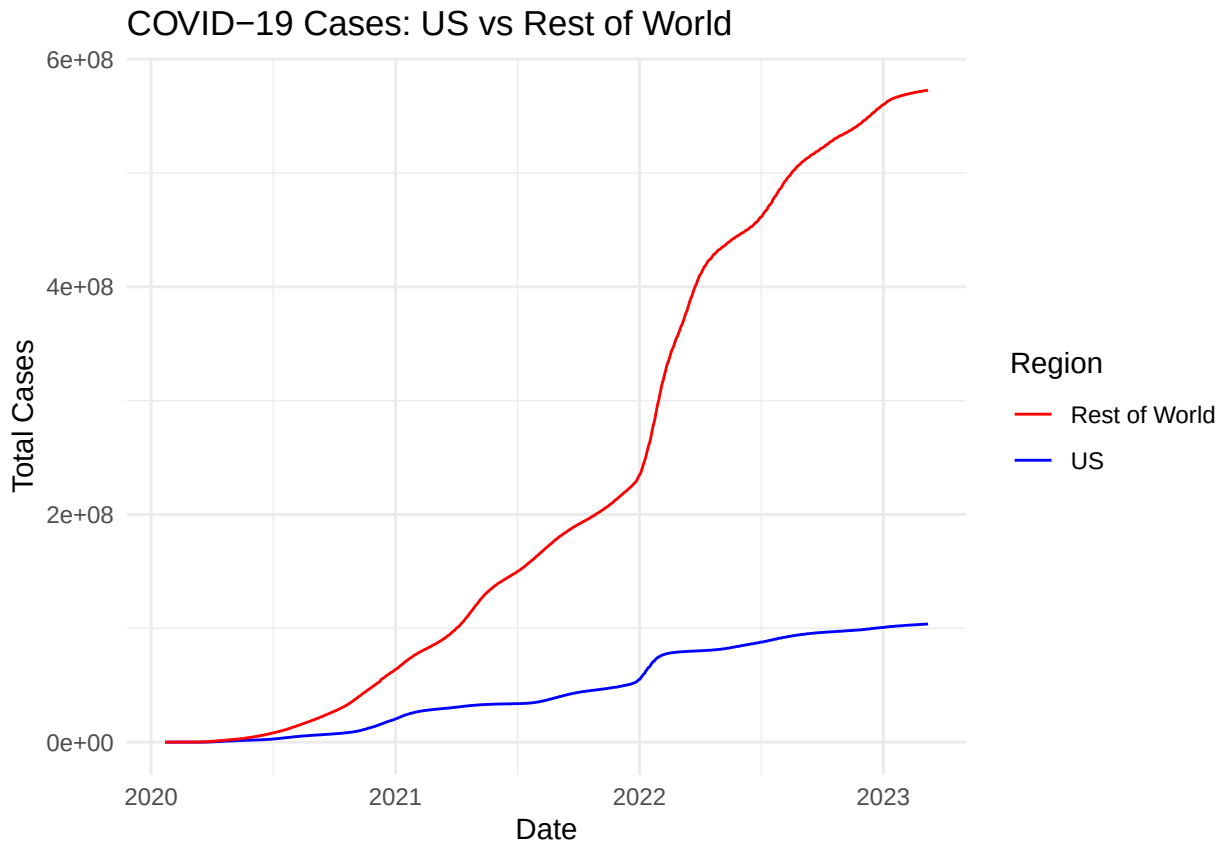
```r
ggplot(summarized_data, aes(x = date, y = total_cases, color = Country_Region)) +
  geom_line() +
  labs(title = "COVID-19 Cases in Mexico, Canada, and the US",
       x = "Date",
       y = "Total Cases",
       color = "Country") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_y_continuous(labels = scales::comma)
```



As we can see on the last chart, the number of cases in the US were much more bigger than it's neighbors. Let's compare the number of cases in the US against the rest of the world

```r
# Filter data for the US and all other countries
us_data <- global %>%
  filter(Country_Region == "US")

rest_of_world_data <- global %>%
  filter(Country_Region != "US")

# Summarize total cases for each date for the rest of the world
rest_of_world_summarized <- rest_of_world_data %>%
  group_by(date) %>%
  summarise(total_cases = sum(cases))

# Plot US cases vs rest of the world
```

```
ggplot() +
  geom_line(data = us_data, aes(x = date, y = cases, color = "US")) +
  geom_line(data = rest_of_world_summarized, aes(x = date, y = total_cases, color = "Rest of World")) +
  labs(title = "COVID-19 Cases: US vs Rest of World",
       x = "Date",
       y = "Total Cases",
       color = "Region") +
  scale_color_manual(values = c("US" = "blue", "Rest of World" = "red")) +
  theme_minimal()
```



Around the 17% of total cases were in the United States which is a lot considering it's population compared with the rest of the world.

Modeling the data

Now let's create a model to predict results for the future

```
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)
##
```

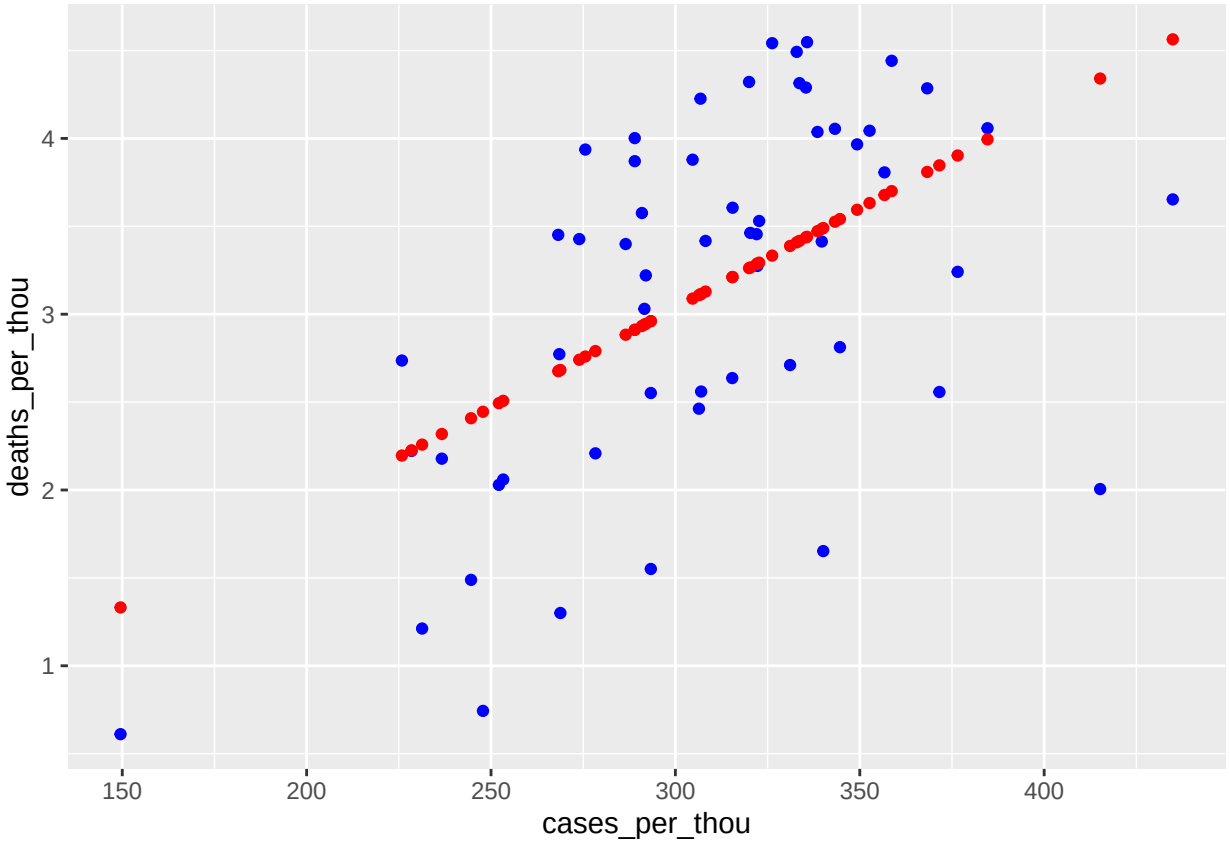
```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.36167    0.72480  -0.499     0.62
## cases_per_thou  0.01133    0.00232   4.881 9.76e-06 ***
## ---
## 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
```

```
#cases_per_thou is a statistically significant predictor of deaths_per_thou, as indicated by the very s
```

```
#Create another dataset with prediction
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
```

```
#Now let's plot real vs prediction
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")
```



Conclusion and bias considerations

Given that COVID-19 was a pandemic that struck the entire world, the collection of data and its standardization became virtually impossible. The policies of each area, restrictions, and the strength of the health system, along with the economic resources of the nation, prevent us from having certainty about the true number of cases and deaths during the years the pandemic lasted.

Even this brief analysis, which is based solely on the United States, is incapable of reflecting 100% of what happened in reality. Such a complex case, with so much bias in its data collection, must be analyzed in far more detail than was seen during the class. Moreover, it would be advisable to focus on even smaller territorial extensions in order to isolate some of the data.

For example, although it seems obvious that the number of deaths is influenced by the number of cases, our model was unable to adjust in a more or less accurate manner to the data presented. This indicates that there are places where perhaps with a lower number of cases, more deaths occurred than in others with many more cases. This discrepancy suggests possible underlying problems such as poor data collection or inadequate prevention campaigns.

There were places where people did not go to the hospital unless they were very ill, which, again, is a factor of social behavior that impacts data collection. Carrying out this exercise in class was very interesting, and it leaves me with a profound understanding of the complexity involved in analyzing databases of this magnitude.