COVID19 Report

Data

The data for this report comes from COVID-19 Data Repository by CSSEt Johns Hopkins University. We are going to be using the provided time series data about global cases, deaths and recoveries.

```
data_root_url = str_c("https://raw.githubusercontent.com/CSSEGISandData/",
                      "COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/")
filenames = c(
  "time_series_covid19_confirmed_global.csv",
  "time_series_covid19_deaths_global.csv",
  "time_series_covid19_recovered_global.csv"
)
file_urls = str_c(data_root_url, filenames)
```

```
global_cases = read_csv(file_urls[1])
global_deaths = read_csv(file_urls[2])
global_recovered = read_csv(file_urls[3])
```

```
head(global_cases)
```

```
## # A tibble: 6 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
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # ... with 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>, 2/21/20 <dbl>,
## #   2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, 2/25/20 <dbl>, ...
```

The data has each date as a column. We are going to transform it, so that date is a column and cases are a separate column. We are also going to remove coordinate columns. Since each of the three datasets are structred similarly, we'll repeat this procedure for each of them.

After that we'll combine them into one table.

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

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

global_recovered = global_recovered %>%
  pivot_longer(
    cols=-c("Province/State", "Country/Region", "Lat", "Long"),
    names_to="date",
    values_to="recovered"
  ) %>%
  select(-c(Lat, Long))

head(global_cases)
```

```
## # A tibble: 6 x 4
##   `Province/State` `Country/Region` date    cases
##   <chr>            <chr>            <chr>   <dbl>
## 1 <NA>             Afghanistan      1/22/20     0
## 2 <NA>             Afghanistan      1/23/20     0
## 3 <NA>             Afghanistan      1/24/20     0
## 4 <NA>             Afghanistan      1/25/20     0
## 5 <NA>             Afghanistan      1/26/20     0
## 6 <NA>             Afghanistan      1/27/20     0
```

```
global = global_cases %>%
  full_join(global_deaths) %>%
  full_join(global_recovered) %>%
  rename(
    country_region="Country/Region",
    province_state="Province/State"
  ) %>%
  mutate(date=mdy(date))
head(global)
```

```
## # A tibble: 6 x 6
##   province_state country_region date       cases deaths recovered
##   <chr>          <chr>          <date>     <dbl>  <dbl>     <dbl>
## 1 <NA>           Afghanistan    2020-01-22     0      0         0
## 2 <NA>           Afghanistan    2020-01-23     0      0         0
## 3 <NA>           Afghanistan    2020-01-24     0      0         0
## 4 <NA>           Afghanistan    2020-01-25     0      0         0
## 5 <NA>           Afghanistan    2020-01-26     0      0         0
## 6 <NA>           Afghanistan    2020-01-27     0      0         0
```

```
summary(global)
```

```
##  province_state     country_region          date                cases
##  Length:331470      Length:331470      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
##                                                          NA's   :1143
##      deaths          recovered
##  Min.   :      0   Min.   :      -1
##  1st Qu.:      3   1st Qu.:       0
##  Median :    150   Median :       0
##  Mean   :  13380   Mean   :   75009
##  3rd Qu.:   3032   3rd Qu.:     934
##  Max.   :1123836   Max.   :30974748
##  NA's   :1143      NA's   :18288
```

Looking at the summary of the combined table, we have a few NAs, rows with 0 cases and rows with -1 recovered. First, let's look which countries have NAs for cases.

```
cases_na = global %>% filter(is.na(cases))
head(cases_na)
```

```
## # A tibble: 6 x 6
##   province_state country_region date       cases deaths recovered
##   <chr>          <chr>          <date>     <dbl>  <dbl>     <dbl>
## 1 <NA>           Canada         2020-01-22    NA     NA         0
## 2 <NA>           Canada         2020-01-23    NA     NA         0
## 3 <NA>           Canada         2020-01-24    NA     NA         0
## 4 <NA>           Canada         2020-01-25    NA     NA         0
## 5 <NA>           Canada         2020-01-26    NA     NA         0
## 6 <NA>           Canada         2020-01-27    NA     NA         0
```

```
unique(cases_na$country_region)
```

```
## [1] "Canada"
```

The only country with NAs for cases is Canada. To make sure that it's a problem with only a specific province, let's see if Canada has rows with not NA cases.

```
tail(global %>% filter(!is.na(cases), country_region=="Canada"))
```

```
## # A tibble: 6 x 6
##   province_state country_region date       cases deaths recovered
##   <chr>          <chr>          <date>     <dbl>  <dbl>     <dbl>
## 1 Yukon          Canada         2023-03-04  4989     32        NA
## 2 Yukon          Canada         2023-03-05  4989     32        NA
## 3 Yukon          Canada         2023-03-06  4989     32        NA
## 4 Yukon          Canada         2023-03-07  4989     32        NA
## 5 Yukon          Canada         2023-03-08  4989     32        NA
## 6 Yukon          Canada         2023-03-09  4989     32        NA
```

It seems that Canada has an invalid NA province which causes problems. So, we are going to remove them and all the rows with 0 cases.

```
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          recovered
##  Min.   :      0   Min.   :     -1
##  1st Qu.:      7   1st Qu.:      0
##  Median :    214   Median :      0
##  Mean   :  14405   Mean   :  79865
##  3rd Qu.:   3665   3rd Qu.:   1235
##  Max.   :1123836   Max.   :30974748
##                    NA's   :16010
```

Removing those rows, also removed NAs for deaths. Now, we need to look at -1 for recovered.

```
global %>% filter(recovered == -1)
```

```
## # A tibble: 8 x 6
##   province_state   country_region date       cases deaths recovered
##   <chr>            <chr>          <date>     <dbl>  <dbl>     <dbl>
## 1 Pitcairn Islands United Kingdom 2022-09-13     4      0        -1
## 2 Pitcairn Islands United Kingdom 2022-09-14     4      0        -1
## 3 Pitcairn Islands United Kingdom 2022-09-15     4      0        -1
## 4 Pitcairn Islands United Kingdom 2022-09-16     4      0        -1
## 5 Pitcairn Islands United Kingdom 2022-09-17     4      0        -1
## 6 Pitcairn Islands United Kingdom 2022-09-18     4      0        -1
## 7 Pitcairn Islands United Kingdom 2022-09-19     4      0        -1
## 8 Pitcairn Islands United Kingdom 2022-09-20     4      0        -1
```

The problem only affects Pitcarnd Islands. We are going to take a look at it in more detail.

```
summary(global %>% filter(province_state == "Pitcairn Islands"))
```

```
##  province_state     country_region          date                cases
##  Length:233         Length:233         Min.   :2022-07-20   Min.   :4
##  Class :character   Class :character   1st Qu.:2022-09-16   1st Qu.:4
##  Mode  :character   Mode  :character   Median :2022-11-13   Median :4
##                                        Mean   :2022-11-13   Mean   :4
##                                        3rd Qu.:2023-01-10   3rd Qu.:4
##                                        Max.   :2023-03-09   Max.   :4
##      deaths    recovered
##  Min.   :0   Min.   :-1.00000
##  1st Qu.:0   1st Qu.: 0.00000
##  Median :0   Median : 0.00000
##  Mean   :0   Mean   :-0.03433
##  3rd Qu.:0   3rd Qu.: 0.00000
##  Max.   :0   Max.   : 0.00000
```

The only problem with it, seems to be those 8 rows. It's probably indicating missing data, and we are going to remove those rows. We also need to look at NAs.

```
na_recovered = global %>% filter(is.na(recovered))
head(na_recovered)
```

```
## # A tibble: 6 x 6
##   province_state country_region date       cases deaths recovered
##   <chr>          <chr>          <date>     <dbl>  <dbl>     <dbl>
## 1 Alberta        Canada         2020-03-06     1      0        NA
```

```
## 2 Alberta        Canada         2020-03-07     2      0        NA
## 3 Alberta        Canada         2020-03-08     4      0        NA
## 4 Alberta        Canada         2020-03-09     7      0        NA
## 5 Alberta        Canada         2020-03-10     7      0        NA
## 6 Alberta        Canada         2020-03-11    19      0        NA
```

```
unique(na_recovered$country_region)
```

```
## [1] "Canada"
```

Once again the problem is with Canada. In this case we'll replace those values with 0 to indicate that there are no known/tracked recoveries.

```
global = global %>%
  replace_na(list(recovered=0)) %>%
  filter(recovered >= 0)
summary(global)
```

```
##  province_state     country_region          date                cases
##  Length:306819      Length:306819      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 :    20366
##                                        Mean   :2021-09-11   Mean   :  1032890
##                                        3rd Qu.:2022-06-15   3rd Qu.:   271286
##                                        Max.   :2023-03-09   Max.   :103802702
##      deaths           recovered
##  Min.   :      0   Min.   :       0
##  1st Qu.:      7   1st Qu.:       0
##  Median :    214   Median :       0
##  Mean   :  14405   Mean   :   75700
##  3rd Qu.:   3666   3rd Qu.:     974
##  Max.   :1123836   Max.   :30974748
```

With NAs dealt with it we are going to make sure that the maximum values are correct.

```
global %>% filter(cases>103000000)
```

```
## # A tibble: 23 x 6
##    province_state country_region date           cases  deaths recovered
##    <chr>          <chr>          <date>         <dbl>   <dbl>     <dbl>
##  1 <NA>           US             2023-02-15 103023231 1115741         0
##  2 <NA>           US             2023-02-16 103083910 1116851         0
##  3 <NA>           US             2023-02-17 103131898 1117572         0
##  4 <NA>           US             2023-02-18 103134605 1117589         0
##  5 <NA>           US             2023-02-19 103136077 1117590         0
##  6 <NA>           US             2023-02-20 103138119 1117663         0
##  7 <NA>           US             2023-02-21 103198669 1118025         0
##  8 <NA>           US             2023-02-22 103308832 1118886         0
##  9 <NA>           US             2023-02-23 103365511 1119521         0
## 10 <NA>           US             2023-02-24 103378408 1119573         0
## # ... with 13 more rows
```

```
global %>% filter(deaths>1100000)
```

```
## # A tibble: 55 x 6
##    province_state country_region date           cases  deaths recovered
##    <chr>          <chr>          <date>         <dbl>   <dbl>     <dbl>
##  1 <NA>           US             2023-01-14 101642336 1100006         0
```

```
##  2 <NA>           US             2023-01-15 101645654 1100023         0
##  3 <NA>           US             2023-01-16 101653171 1100068         0
##  4 <NA>           US             2023-01-17 101734426 1100812         0
##  5 <NA>           US             2023-01-18 101863056 1102393         0
##  6 <NA>           US             2023-01-19 101954244 1103712         0
##  7 <NA>           US             2023-01-20 101991763 1104154         0
##  8 <NA>           US             2023-01-21 101996891 1104178         0
##  9 <NA>           US             2023-01-22 102000179 1104191         0
## 10 <NA>           US             2023-01-23 102031232 1104505         0
## # ... with 45 more rows
```

```
global %>% filter(recovered>30000000)
```

```
## # A tibble: 25 x 6
##    province_state country_region date          cases deaths recovered
##    <chr>          <chr>          <date>        <dbl>  <dbl>     <dbl>
##  1 <NA>           India          2021-07-11 30874376 408764  30014713
##  2 <NA>           India          2021-07-12 30907282 410784  30063720
##  3 <NA>           India          2021-07-13 30946147 411406  30104659
##  4 <NA>           India          2021-07-14 30987880 411989  30143850
##  5 <NA>           India          2021-07-15 31026829 412531  30183876
##  6 <NA>           India          2021-07-16 31064908 413091  30227792
##  7 <NA>           India          2021-07-17 31106065 413609  30269796
##  8 <NA>           India          2021-07-18 31144229 414108  30308456
##  9 <NA>           India          2021-07-19 31174322 414482  30353710
## 10 <NA>           India          2021-07-20 31216337 418480  30390687
## # ... with 15 more rows
```

All the values seem to be valid, but there might be a problem with recoveries. For US there are 0 and for India they seem to stop in 2021. Let's see what the latest date with nonzero number of recoveries.

```
max((global %>% filter(recovered>0))$date)
```

```
## [1] "2021-08-04"
```

It seems that recoveries have not been tracked since August 2021. Since it was almost 2 years ago, it means the information is way out of date and we won't be able to use it for the analysis. Thus we are going to remove this column.

```
global = global %>% select(-c("recovered"))
tail(global)
```

```
## # A tibble: 6 x 5
##   province_state country_region date        cases deaths
##   <chr>          <chr>          <date>      <dbl>  <dbl>
## 1 <NA>           Zimbabwe       2023-03-04 264127   5668
## 2 <NA>           Zimbabwe       2023-03-05 264127   5668
## 3 <NA>           Zimbabwe       2023-03-06 264127   5668
## 4 <NA>           Zimbabwe       2023-03-07 264127   5668
## 5 <NA>           Zimbabwe       2023-03-08 264276   5671
## 6 <NA>           Zimbabwe       2023-03-09 264276   5671
```

Now, let's check if there is a similar problem with cases or deaths.

```
unique(global %>% filter(date>"2023-01-01", deaths==0) %>% select(province_state, country_region))
```

```
## # A tibble: 17 x 2
##    province_state                    country_region
```

```
##    <chr>                                          <chr>
##  1 <NA>                                           Antarctica
##  2 Grand Princess                                 Canada
##  3 Repatriated Travellers                         Canada
##  4 Jiangsu                                        China
##  5 Ningxia                                        China
##  6 Qinghai                                        China
##  7 Shanxi                                         China
##  8 Tibet                                          China
##  9 Unknown                                        China
## 10 <NA>                                           Holy See
## 11 Niue                                           New Zealand
## 12 <NA>                                           Summer Olympics 2020
## 13 <NA>                                           Tuvalu
## 14 Falkland Islands (Malvinas)                    United Kingdom
## 15 Pitcairn Islands                               United Kingdom
## 16 Saint Helena, Ascension and Tristan da Cunha United Kingdom
## 17 <NA>                                           Winter Olympics 2022
```

```
unique(global %>% filter(date>"2023-01-01", cases==0) %>% select(province_state, country_region))
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: province_state <chr>, country_region <chr>
```

There are some place with no deaths, but since they had cases and considering their locations, it seems that those are correct.

Since we have finished cleaning the COVID data, we are going to add population statistics in order to be able to calculate additional statistics. The population data comes from the same repository.

```
pop_url = str_c("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/",
                "master/csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv")
population = read_csv(pop_url) %>%
  group_by(Country_Region, Province_State) %>%
  summarize(Population=sum(Population)) %>%
  rename(
    country_region=Country_Region,
    province_state=Province_State,
    population=Population
  ) %>%
  ungroup()
summary(population)
```

```
##  country_region     province_state       population
##  Length:978         Length:978         Min.   :6.700e+01
##  Class :character   Class :character   1st Qu.:5.404e+05
##  Mode  :character   Mode  :character   Median :1.704e+06
##                                        Mean   :1.369e+07
##                                        3rd Qu.:5.853e+06
##                                        Max.   :1.412e+09
##                                        NA's   :90
```

They are a few NAs for the population column we need to check out.

```
population %>% filter(is.na(population))
```

```
## # A tibble: 90 x 3
##    country_region   province_state   population
```

```
##    <chr>            <chr>                <dbl>
##  1 Antarctica       <NA>                    NA
##  2 Belgium          Unknown                 NA
##  3 Brazil           Unknown                 NA
##  4 Canada           Diamond Princess        NA
##  5 Canada           Grand Princess          NA
##  6 Canada           Recovered               NA
##  7 Chile            Unknown                 NA
##  8 China            Unknown                 NA
##  9 Colombia         Unknown                 NA
## 10 Diamond Princess <NA>                    NA
## # ... with 80 more rows
```

Those seem to be mostly erroneous, missing or repeated additions, which we can ignore, since even if some of them have unaccounted population, the population data is not that accurate in many place anyway and it contains populations numbers from 2020.

```
population = population %>% filter(!is.na(population))
summary(population)
```

```
##  country_region     province_state       population
##  Length:888         Length:888         Min.   :6.700e+01
##  Class :character   Class :character   1st Qu.:5.404e+05
##  Mode  :character   Mode  :character   Median :1.704e+06
##                                        Mean   :1.369e+07
##                                        3rd Qu.:5.853e+06
##                                        Max.   :1.412e+09
```

With that done, we can add population numbers to the combined table.

```
global = global %>%
  left_join(
    population,
    by=c("province_state", "country_region")
  )
```

Additionally, we'll prepare tables with aggregated statistics globally and for each country separately

```
global_by_country = global %>%
  group_by(country_region, date) %>%
  summarize(cases=sum(cases), deaths=sum(deaths), population=sum(population)) %>%
  ungroup()
tail(global_by_country)
```

```
## # A tibble: 6 x 5
##   country_region date        cases deaths population
##   <chr>          <date>      <dbl>  <dbl>      <dbl>
## 1 Zimbabwe       2023-03-04 264127   5668   14862927
## 2 Zimbabwe       2023-03-05 264127   5668   14862927
## 3 Zimbabwe       2023-03-06 264127   5668   14862927
## 4 Zimbabwe       2023-03-07 264127   5668   14862927
## 5 Zimbabwe       2023-03-08 264276   5671   14862927
## 6 Zimbabwe       2023-03-09 264276   5671   14862927
```

```
global_totals = global_by_country %>%
  group_by(date) %>%
  summarize(cases=sum(cases), deaths=sum(deaths)) %>%
  ungroup()
```

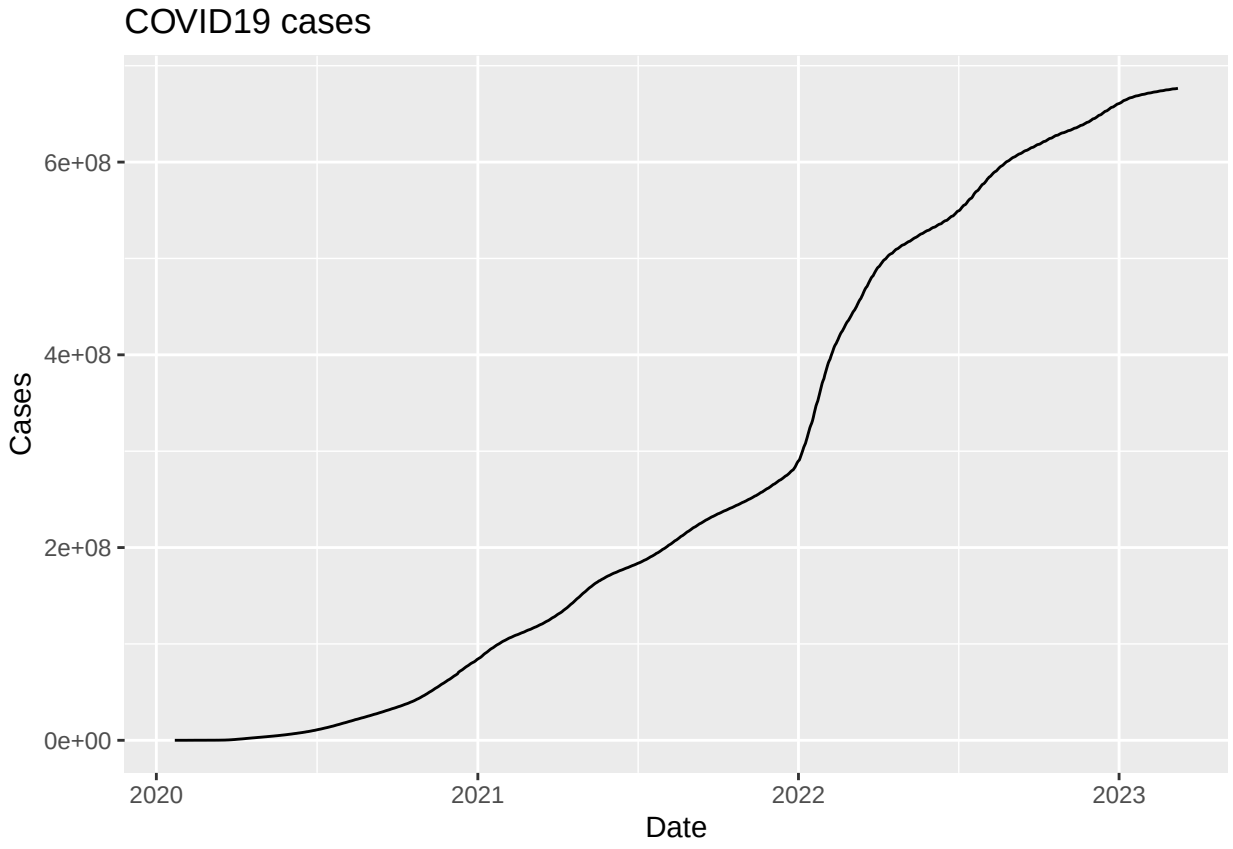
```
tail(global_totals)
```

```
## # A tibble: 6 x 3
##   date           cases  deaths
##   <date>         <dbl>   <dbl>
## 1 2023-03-04 675968775 6877600
## 2 2023-03-05 676024901 6877748
## 3 2023-03-06 676082941 6878114
## 4 2023-03-07 676213378 6879037
## 5 2023-03-08 676392824 6880482
## 6 2023-03-09 676570149 6881801
```

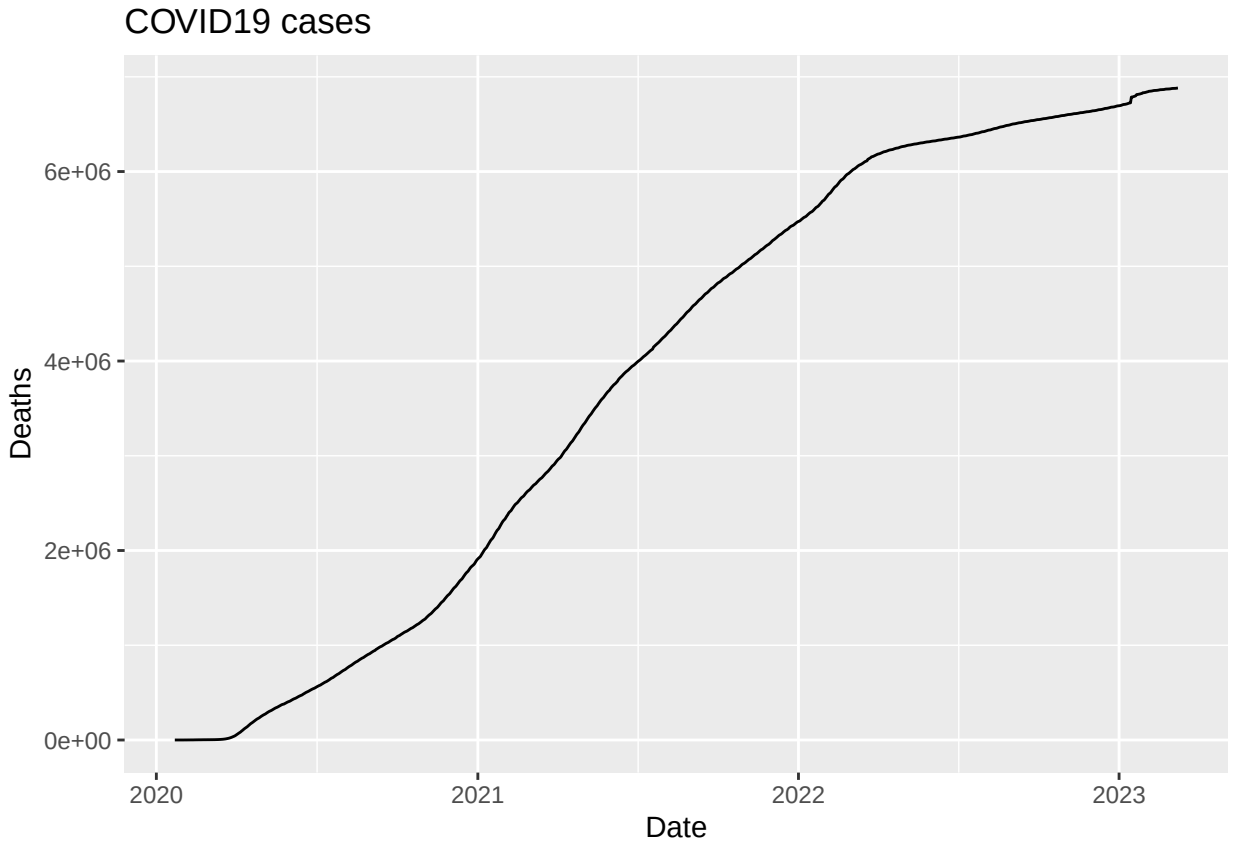
Visualizations

With the data cleaned, we can move on to analyzing it. First, we'll start with visualizations.

```
global_totals %>%
  ggplot(aes(x=date, y=cases)) +
  geom_line() +
  labs(title="COVID19 cases", x="Date", y="Cases")
```

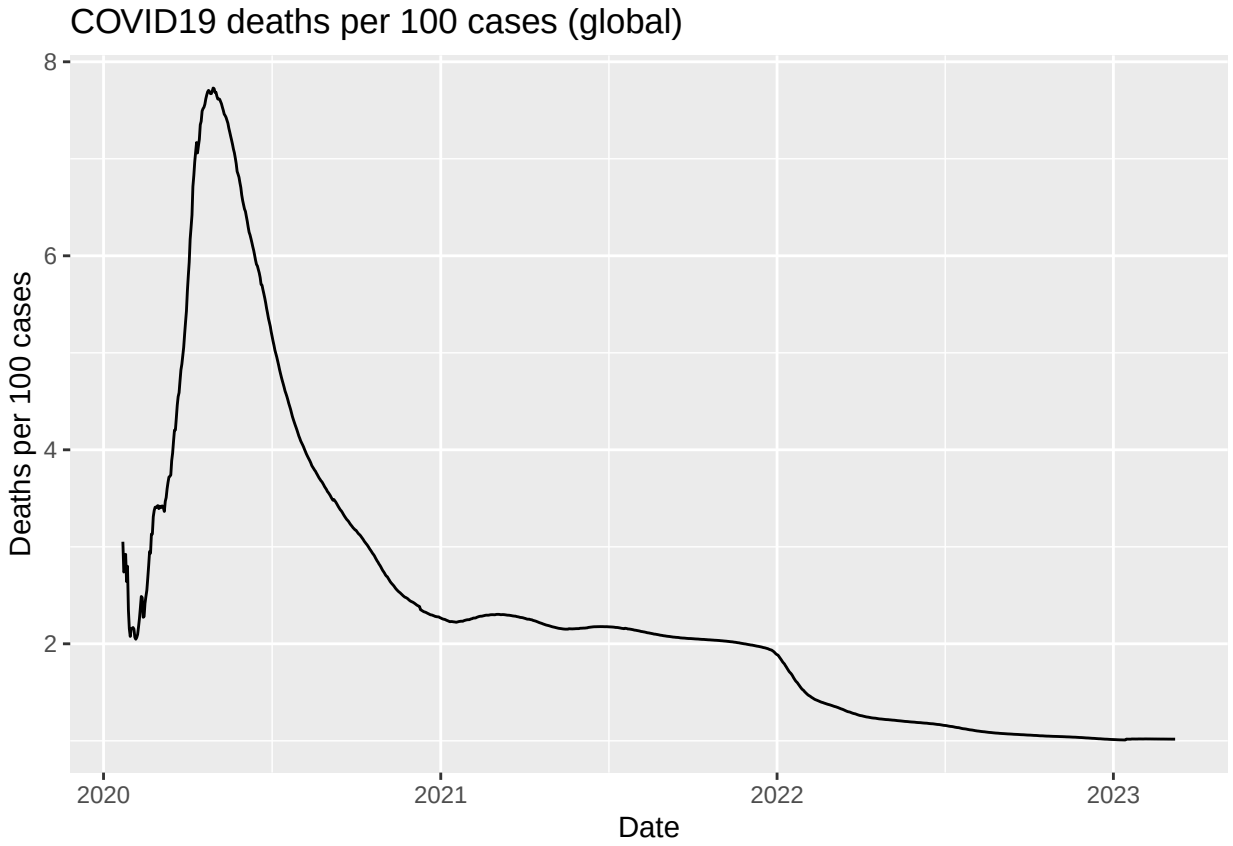



```
global_totals %>%
  ggplot(aes(x=date, y=deaths)) +
  geom_line() +
  labs(title="COVID19 cases", x="Date", y="Deaths")
```



From these we can see that, as expected, deaths and cases are correlated and we can see how COVID progressed through the beginning, vaccine roll out, mitigation removal and Omicron variant until the end of extensive monitoring. From these graph, we can see that lethality trended down. To check that we can look at the number of deaths per 100 cases.

```
global_totals %>%
  mutate(deaths_per_100_cases=deaths/cases*100) %>%
  ggplot(aes(x=date, y=deaths_per_100_cases)) +
  geom_line() +
  labs(title="COVID19 deaths per 100 cases (global)", x="Date", y="Deaths per 100 cases")
```



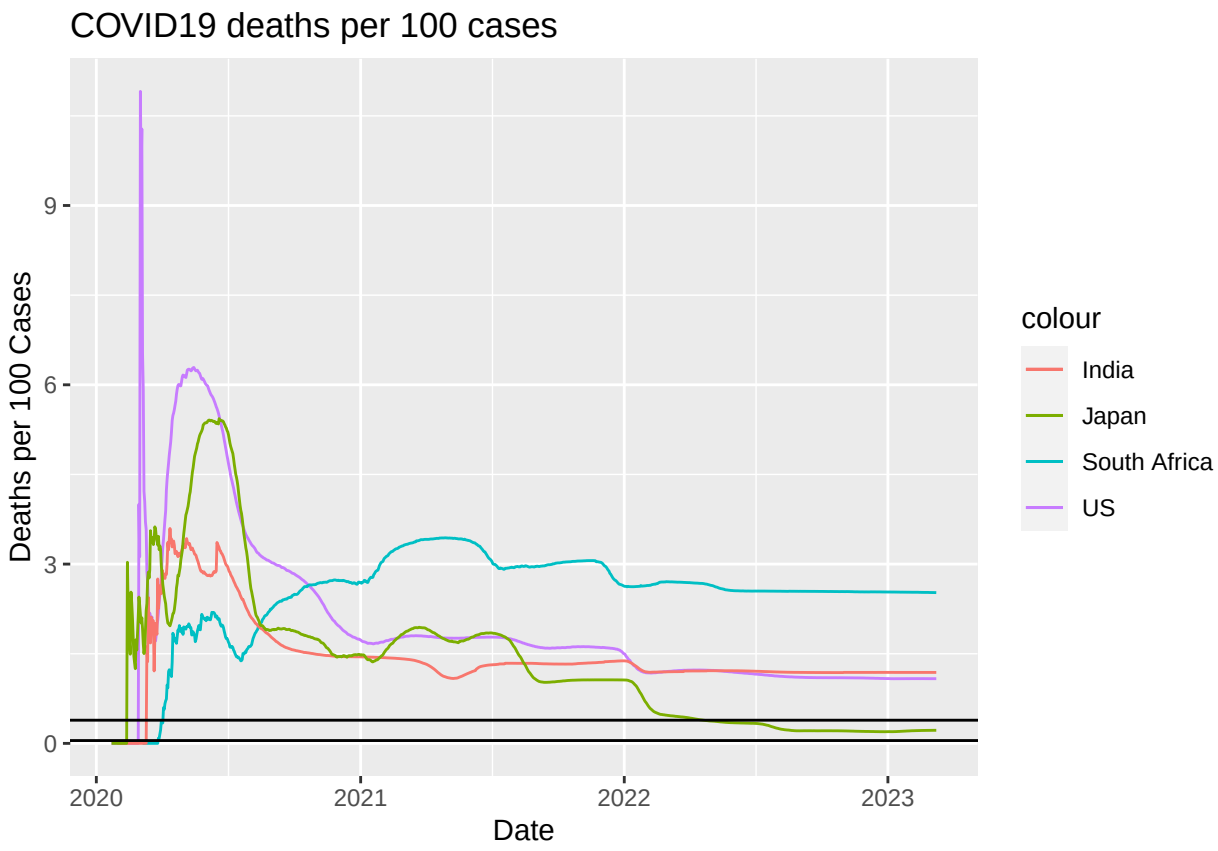
The visualization cofirms the hypothesis. Lethality did drop down, ending at around 1%. One thing that aggregated statistics can hide is the differences between groups. For example, we can look at the same plot, but aggregated by country to see how it can differ from the global trend.

```
global_by_country = global_by_country %>%
  mutate(
    deaths_per_100_cases=deaths/cases*100,
    cases_per_100_pop=cases/population*100,
  )
tail(global_by_country %>% select(deaths_per_100_cases, cases_per_100_pop, everything()))
```

```
## # A tibble: 6 x 7
##   deaths_per_100_cases cases_per_100_pop country_region date        cases deaths
##                  <dbl>             <dbl> <chr>          <date>      <dbl>  <dbl>
## 1                 2.15              1.78 Zimbabwe       2023-03-04 264127   5668
## 2                 2.15              1.78 Zimbabwe       2023-03-05 264127   5668
## 3                 2.15              1.78 Zimbabwe       2023-03-06 264127   5668
## 4                 2.15              1.78 Zimbabwe       2023-03-07 264127   5668
## 5                 2.15              1.78 Zimbabwe       2023-03-08 264276   5671
## 6                 2.15              1.78 Zimbabwe       2023-03-09 264276   5671
## # ... with 1 more variable: population <dbl>
```

```
ggplot(data=global_by_country, aes(x=date, y=deaths_per_100_cases)) +
  geom_line(data=subset(global_by_country, country_region=="US"), aes(color="US")) +
  geom_line(data=subset(global_by_country, country_region=="South Africa"), aes(color="South Africa")) +
  geom_line(data=subset(global_by_country, country_region=="India"), aes(color="India")) +
  geom_line(data=subset(global_by_country, country_region=="Japan"), aes(color="Japan")) +
  geom_abline(intercept=97000/25000000 * 100, slope=0, aes(color="Influenza (high)")) +
```

```
  geom_abline(intercept=19000/40000000 * 100, slope=0, aes(color="Influenza (low)")) +
  labs(title="COVID19 deaths per 100 cases", x="Date", y="Deaths per 100 Cases")
```



This plot reveals that there are substantial differences in mortality between countries. For example, South Africa has mortality as high as 2.5%, while Japan is around the value of Influenza mortality in US - between 0.4% and 0.05% (calculated as upper and lower limits from estimates from CDC. Other groupings that will differ are based on ages, but the breakdown isn't included in this data and we'll not look into it.

Now, let's move onto looking at the number of new cases and deaths. Since the data is accumulated over 3 years, we are going to using a weekly granularity.

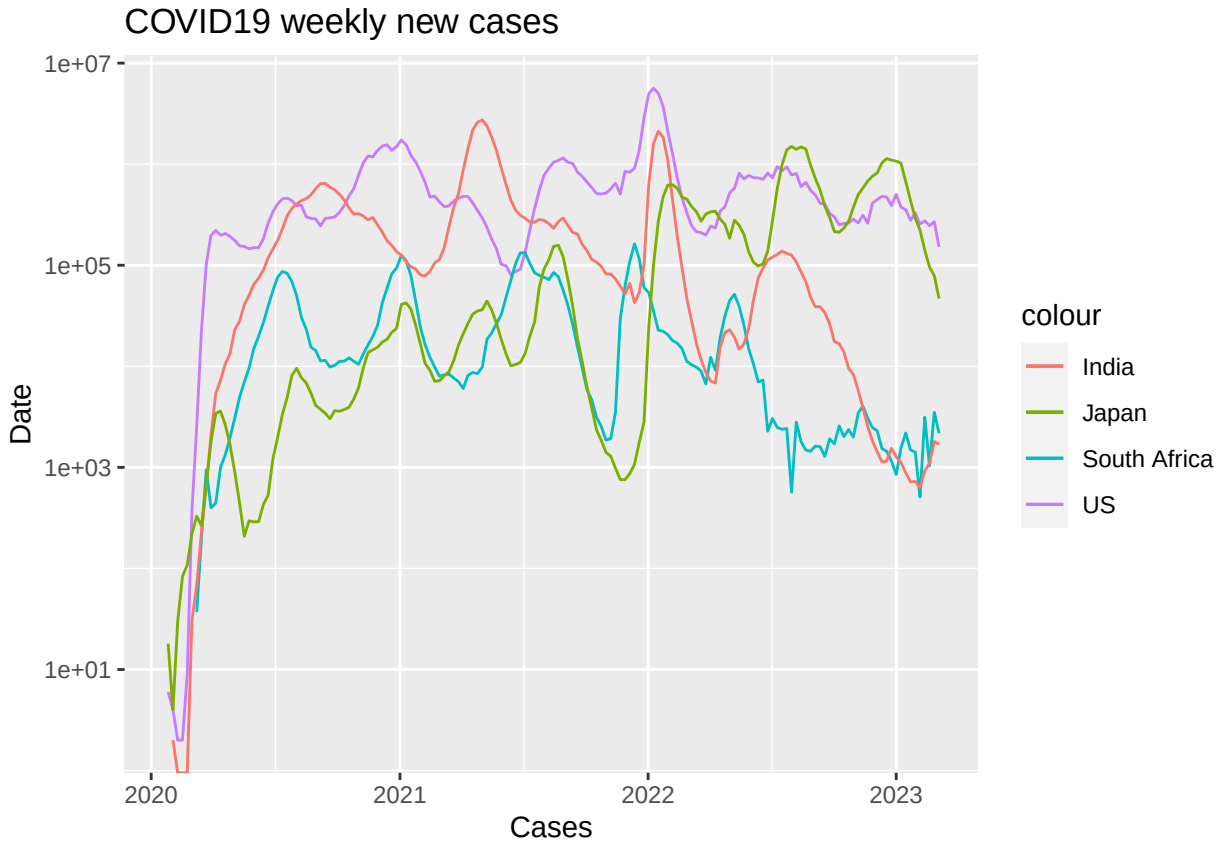
```
global_by_country = global_by_country %>%
  mutate(
    new_cases=cases-lag(cases),
    new_deaths=deaths-lag(deaths)
  )

global_by_country_weekly =
  global_by_country %>%
  mutate(week=floor_date(date, "week")) %>%
  group_by(country_region, week) %>%
  summarise(
    cases=max(cases), deaths=max(deaths),
    new_cases=sum(new_cases), new_deaths=sum(new_deaths),
    population=max(population)
  ) %>%
  mutate(
    deaths_per_100_cases=deaths/cases*100,
```

```
    deaths_per_100_pop=deaths/population*100,
    cases_per_100_pop=cases/population*100
  ) %>%
  ungroup()
head(global_by_country_weekly %>% filter(new_deaths>0) %>% select(new_cases, cases, new_deaths, deaths,
```

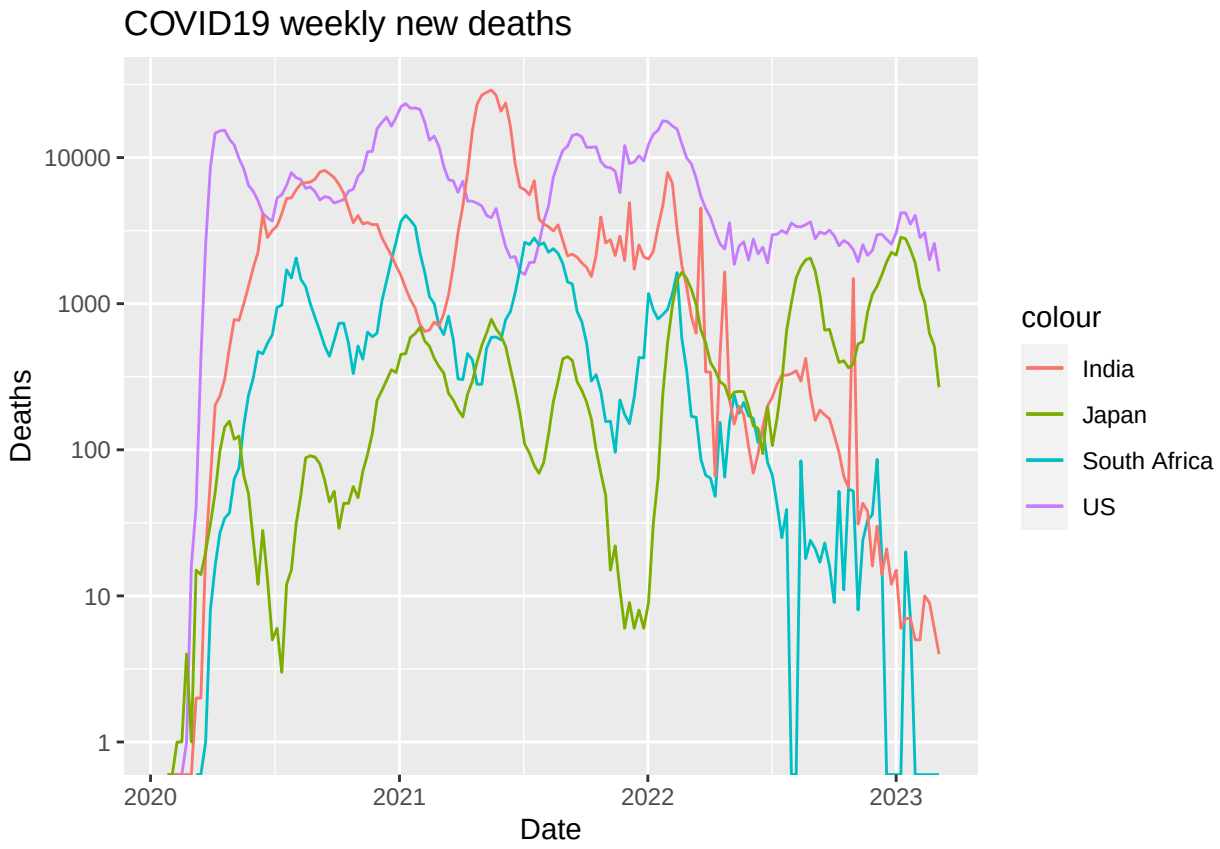
```
## # A tibble: 6 x 10
##   new_cases cases new_deaths deaths country_region week       population
##       <dbl> <dbl>      <dbl>  <dbl> <chr>          <date>          <dbl>
## 1        82   106          2      2 Afghanistan    2020-03-22   38928341
## 2       164   270          3      5 Afghanistan    2020-03-29   38928341
## 3       251   521         10     15 Afghanistan    2020-04-05   38928341
## 4       387   908         15     30 Afghanistan    2020-04-12   38928341
## 5       422  1330         13     43 Afghanistan    2020-04-19   38928341
## 6      1139  2469         29     72 Afghanistan    2020-04-26   38928341
## # ... with 3 more variables: deaths_per_100_cases <dbl>,
## #   deaths_per_100_pop <dbl>, cases_per_100_pop <dbl>
```

```
ggplot(data=global_by_country_weekly, aes(x=week, y=new_cases)) +
  geom_line(data=subset(global_by_country_weekly, country_region=="US"), aes(color="US")) +
  geom_line(data=subset(global_by_country_weekly, country_region=="South Africa"), aes(color="South Afri
  geom_line(data=subset(global_by_country_weekly, country_region=="India"), aes(color="India")) +
  geom_line(data=subset(global_by_country_weekly, country_region=="Japan"), aes(color="Japan")) +
  scale_y_log10() +
  labs(title="COVID19 weekly new cases", x="Cases", y="Date")
```




```
ggplot(data=global_by_country_weekly, aes(x=week, y=new_deaths)) +
  geom_line(data=subset(global_by_country_weekly, country_region=="US"), aes(color="US")) +
```

```
  geom_line(data=subset(global_by_country_weekly, country_region=="South Africa"), aes(color="South Afri
  geom_line(data=subset(global_by_country_weekly, country_region=="India"), aes(color="India")) +
  geom_line(data=subset(global_by_country_weekly, country_region=="Japan"), aes(color="Japan")) +
  scale_y_log10() +
  labs(title="COVID19 weekly new deaths", x="Date", y="Deaths")
```



Here, we can see that new cases and deaths are following a similar periodic trend, however, towards the end there are differences. For South Africa and India, new deaths have been dropping for a year while new cases were more stable and started rising in 2023. For the US, both new deaths and new cases have stabilized around 20000 cases and 2500 deaths both dipping slightly in March 2023. In Japan, the periodic trend continued.

This again highlights how big are the differences between countries. The causes for that are not clear and can vary from under reporting due to not having enough testing capacity to maintaining a high level of COVID mitigations such as the use of respirators and air filtering.

Since the dynamics also differ depending on the period selected, we are going to drop the first months of the pandemic, approximately from the start of gradual return to schools around the world, and do the modeling for that range and for the tail end of the time frame.

```
global_by_country_weekly_wo_start = global_by_country_weekly %>%
  filter(week > "2020-09-15")
```

Modeling

Since we mostly focused on analyzing 4 countries, we are going to stick with them. This will both allow to inspect the trends for them and also avoid misinterpreting any possible things we might have not looked at for other countries. However, this means that the models will not be suitable to use in the global context.

```
train_data = global_by_country_weekly_wo_start %>%
  filter(country_region %in% c("India", "Japan", "South Africa", "US")) %>%
  mutate(country_region=as.factor(country_region))
tail(train_data)
```

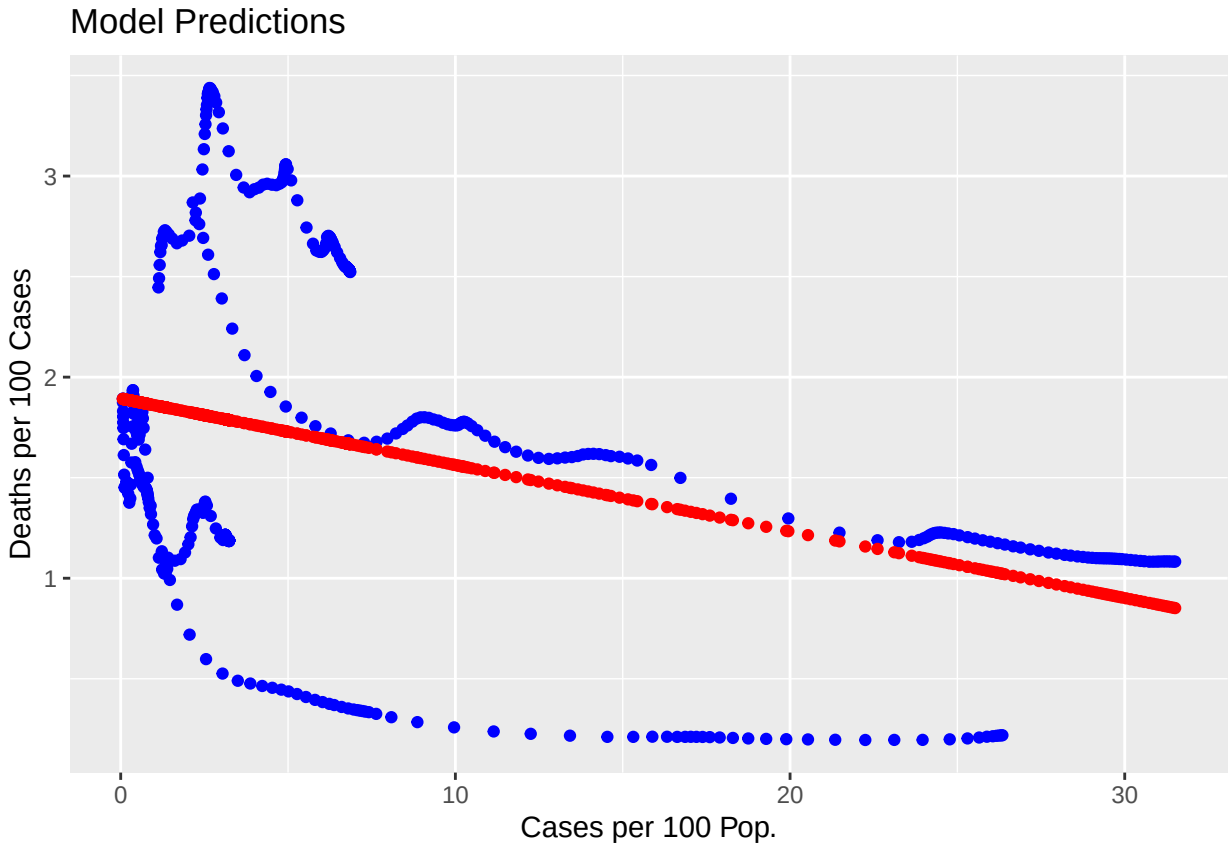
```
## # A tibble: 6 x 10
##   country_region week           cases  deaths new_cases new_deaths population
##   <fct>          <date>         <dbl>   <dbl>     <dbl>      <dbl>      <dbl>
## 1 US             2023-01-29 102603942 1111696    329036       4020  329466283
## 2 US             2023-02-05 102859510 1114529    255568       2833  329466283
## 3 US             2023-02-12 103134605 1117589    275095       3060  329466283
## 4 US             2023-02-19 103381157 1119587    246552       1998  329466283
## 5 US             2023-02-26 103650837 1122172    269680       2585  329466283
## 6 US             2023-03-05 103802702 1123836    151865       1664  329466283
## # ... with 3 more variables: deaths_per_100_cases <dbl>,
## #   deaths_per_100_pop <dbl>, cases_per_100_pop <dbl>
```

First, we'll try to fit the mortality rate based on the number of cases per 100 pop.

```
model_dc = lm(deaths_per_100_cases ~ cases_per_100_pop, data=train_data)
summary(model_dc)
```

```
##
## Call:
## lm(formula = deaths_per_100_cases ~ cases_per_100_pop, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31727 -0.59191  0.03513  0.79937  1.63215
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.893528   0.045189  41.903   <2e-16 ***
## cases_per_100_pop -0.033073   0.003801  -8.701   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7628 on 514 degrees of freedom
## Multiple R-squared:  0.1284, Adjusted R-squared:  0.1267
## F-statistic: 75.71 on 1 and 514 DF,  p-value: < 2.2e-16
```

```
with_dc_prediction = copy(train_data)
with_dc_prediction$prediction = predict(model_dc, new=train_data)
with_dc_prediction %>%
  ggplot(aes(x=cases_per_100_pop)) +
  geom_point(aes(y=deaths_per_100_cases), color="blue") +
  geom_point(aes(y=prediction), color="red") +
  labs(title="Model Predictions", x="Cases per 100 Pop.", y="Deaths per 100 Cases")
```



We can see that the model without a factor for countries doesn't fit well individual countries. However, it still captures the general trend. Looking at the coefficients, cases per 100 pop has a negative coefficient. This does not mean that with a larger proportion of infected population, the lethality drops, but rather that this number might correlate with time. However, there still might be reasons for the decrease in mortality. For example, with more cases treated doctors develop better ways to provide treatment for severe cases.

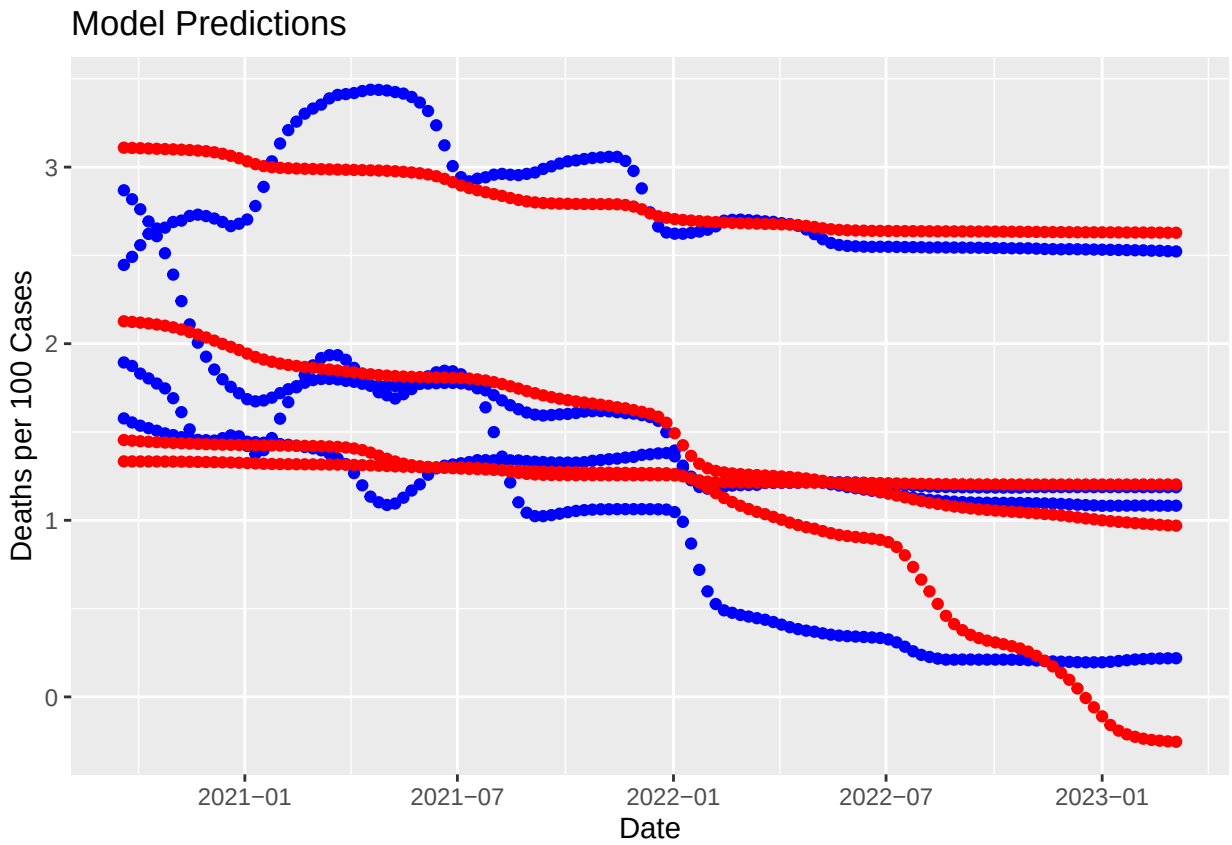
Now, let's try to fit a model that includes country as a predictor.

```
model_dcwc = lm(deaths_per_100_cases ~ cases_per_100_pop*country_region, data=train_data)
summary(model_dcwc)
```

```
##
## Call:
## lm(formula = deaths_per_100_cases ~ cases_per_100_pop * country_region,
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66394 -0.09384 -0.01500  0.07900  0.74183
##
## Coefficients:
##                                            Estimate Std. Error t value
## (Intercept)                                 1.49361    0.05715  26.135
## cases_per_100_pop                          -0.08982    0.02254  -3.985
## country_regionJapan                        -0.15597    0.06373  -2.447
## country_regionSouth Africa                  1.71125    0.08138  21.028
## country_regionUS                            0.71823    0.07417   9.683
## cases_per_100_pop:country_regionJapan       0.02941    0.02270   1.296
```

```
## cases_per_100_pop:country_regionSouth Africa  0.00574    0.02511   0.229
## cases_per_100_pop:country_regionUS            0.05039    0.02266   2.224
##                                                Pr(>|t|)
## (Intercept)                                     < 2e-16 ***
## cases_per_100_pop                              7.74e-05 ***
## country_regionJapan                             0.0147 *
## country_regionSouth Africa                      < 2e-16 ***
## country_regionUS                                < 2e-16 ***
## cases_per_100_pop:country_regionJapan           0.1956
## cases_per_100_pop:country_regionSouth Africa    0.8193
## cases_per_100_pop:country_regionUS              0.0266 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2516 on 508 degrees of freedom
## Multiple R-squared:  0.9063, Adjusted R-squared:  0.905
## F-statistic: 701.7 on 7 and 508 DF,  p-value: < 2.2e-16
```

```
with_dcwc_prediction = copy(train_data)
with_dcwc_prediction$prediction = predict(model_dcwc, new=train_data)
with_dcwc_prediction %>%
  ggplot(aes(x=week)) +
  geom_point(aes(y=deaths_per_100_cases), color="blue") +
  geom_point(aes(y=prediction), color="red") +
  labs(title="Model Predictions", x="Date", y="Deaths per 100 Cases")
```



We can immediately see that this model fits the data much better. The adjusted R^2 improves from 0.1267 to 0.905. The next improvement can be achieved by separating the different stages of the pandemic (for

example, by including the current variant).

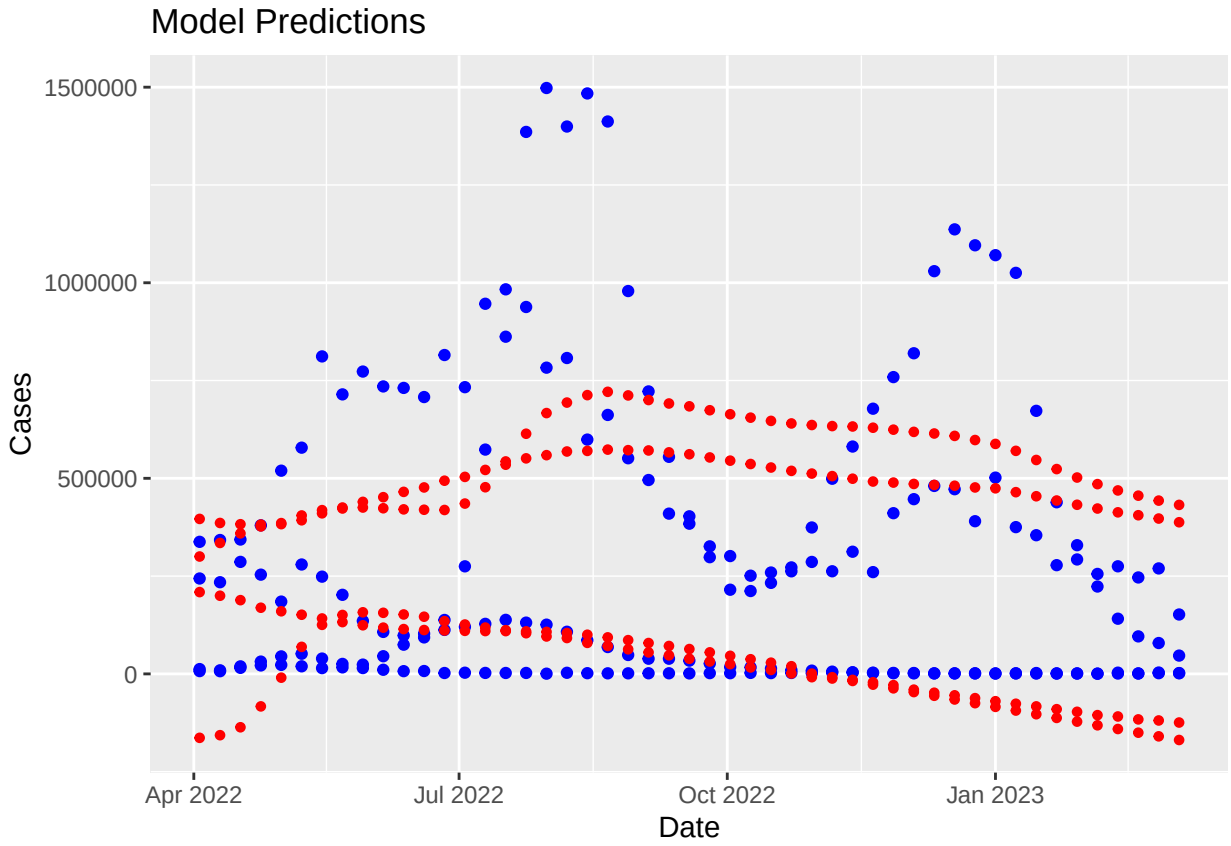
Looking at the coefficients, we can see that all non combined predictors are significant and from combinations only cases per 100 pop and US is significant. I am not sure why is that and it would be interesting to investigate what is the underlying reason for the difference in the slope for the US.

Finally, we are going to model new cases and deaths. Since we know that there are definite differences in the data distribution based on time, we are only going to use the data approximately after the Omicron peak at the start of 2022.

```
train_data_latest = train_data %>% filter(week > "2022-04-01")
model_tcfull = lm(
  new_cases ~ country_region + deaths_per_100_cases + as.numeric(week), data=train_data_latest)
summary(model_tcfull)
```

```
##
## Call:
## lm(formula = new_cases ~ country_region + deaths_per_100_cases +
##     as.numeric(week), data = train_data_latest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -448605 -136551  -18219   95161  831048
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 29713080    5266047   5.642 6.02e-08 ***
## country_regionJapan        -2381373     497796  -4.784 3.44e-06 ***
## country_regionSouth Africa  4154971     718358   5.784 2.96e-08 ***
## country_regionUS             233329      58456   3.991 9.37e-05 ***
## deaths_per_100_cases       -3079116     527869  -5.833 2.31e-08 ***
## as.numeric(week)              -1350        248  -5.446 1.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225900 on 190 degrees of freedom
## Multiple R-squared:  0.5879, Adjusted R-squared:  0.577
## F-statistic: 54.21 on 5 and 190 DF,  p-value: < 2.2e-16
```

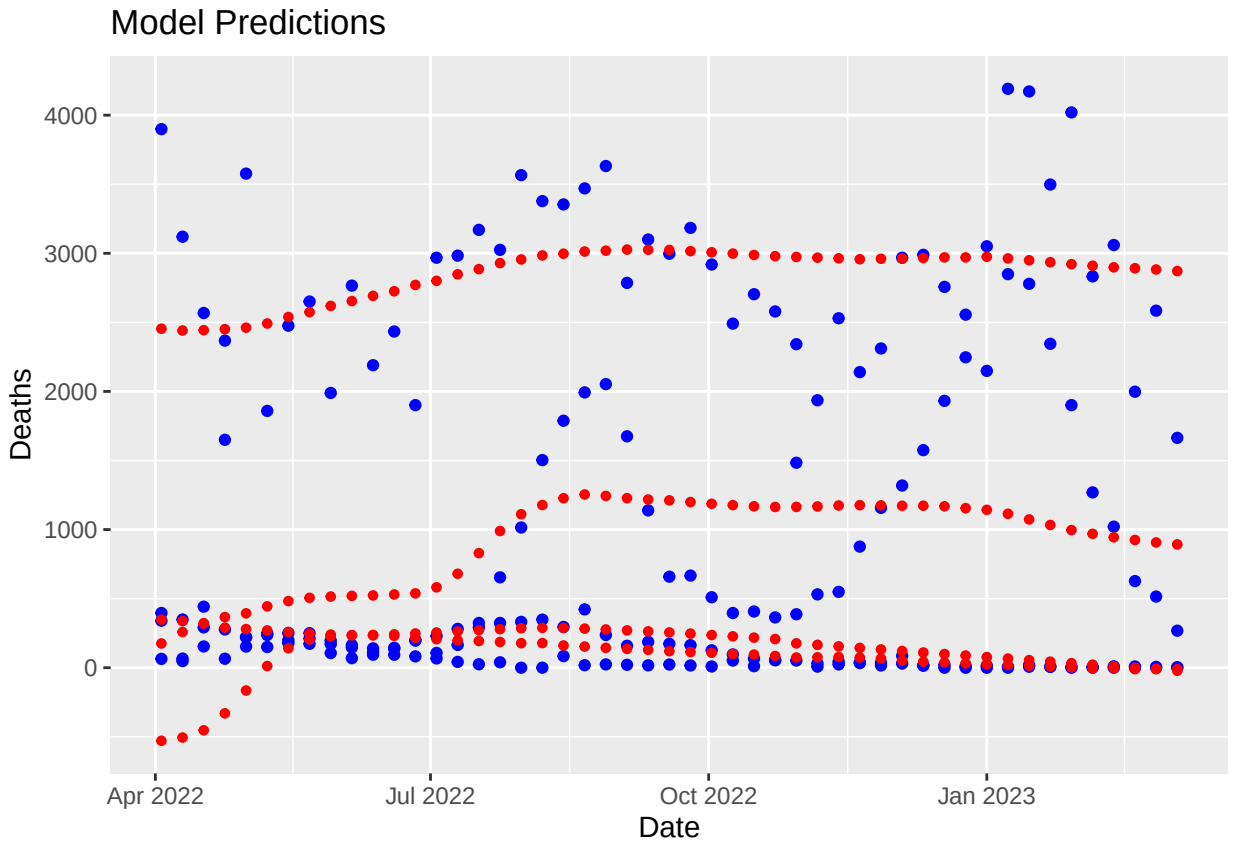
```
with_tcfull_prediction = copy(train_data_latest)
with_tcfull_prediction$prediction = predict(model_tcfull, new=train_data_latest)
with_tcfull_prediction %>%
  ggplot(aes(x=week)) +
  geom_point(aes(y=new_cases), color="blue") +
  geom_point(aes(y=prediction), color="red", size=1.2) +
  labs(title="Model Predictions", x="Date", y="Cases")
```




```
model_tdfull = lm(
  new_deaths ~ country_region + deaths_per_100_cases + as.numeric(week), data=train_data_latest)
summary(model_tdfull)
```

```
##
## Call:
## lm(formula = new_deaths ~ country_region + deaths_per_100_cases +
##     as.numeric(week), data = train_data_latest)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1207.60  -167.36   -59.20    76.34  1734.76
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 38449.1337 11145.9032   3.450 0.000692 ***
## country_regionJapan         -5440.3028  1053.6149  -5.163 6.09e-07 ***
## country_regionSouth Africa   8769.9000  1520.4485   5.768 3.21e-08 ***
## country_regionUS             2204.5296   123.7266  17.818  < 2e-16 ***
## deaths_per_100_cases        -6560.6507  1117.2666  -5.872 1.89e-08 ***
## as.numeric(week)               -1.5797     0.5248  -3.010 0.002968 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 478.1 on 190 degrees of freedom
## Multiple R-squared:  0.8536, Adjusted R-squared:  0.8497
## F-statistic: 221.5 on 5 and 190 DF,  p-value: < 2.2e-16
```

```
with_tdfull_prediction = copy(train_data_latest)
with_tdfull_prediction$prediction = predict(model_tdfull, new=train_data_latest)
with_tdfull_prediction %>%
  ggplot(aes(x=week)) +
  geom_point(aes(y=new_deaths), color="blue") +
  geom_point(aes(y=prediction), color="red", size=1.2) +
  labs(title="Model Predictions", x="Date", y="Deaths")
```



We can see that the models have trouble fitting the data due to how variable it is. In both cases, there is a slight decrease with time, but that decrease is low. However, it's important to remember that there is a period trend, which is not capture by linear models and it might be incorrect to draw any conclusions from this.

Conclusions

While this dataset allows to perform quite a few analyses and the report only barely touches a few directions, it is important to remember that the data might not be as representative as one hopes. First of all, it focuses only on cases and deaths. This completely ignores any possible long term complications that COVID causes. Also, the definitions themselves are difficult to determine and differ between countries. For example, should a death a month after a negative test count if there are no other apparent reasons and how can this be tracked? Secondly, the testing methodology and behaviour are changing. Right now, I would expect fewer people to be doing tests and the tests themselves are worse due to the amount of new variants. This can cause undercounting for both the cases and deaths. For the report itself, the main limitation is that it only focused on 4 selected countries. While the countries are from different regions, they are still not representative of other countries due to how many possible variations they are.

Nevertheless, I think it's pretty clear from the data that at the moment where it stops, there is not enough

information to determine if it's going to continue to decrease, stay at approximately the same level without peaks and valleys or it's going to continue the same periodic trend. Taking into the account new emerging variants, I would say it's important to keep tracking COVID19 extensively and to be aware of the possible risks.