Final Project 2: Reproducible Report on COVID19 Data

Madhumita Mondal

2023-06-23

COVID-19 data analysis

The COVID-19 pandemic has caused significant disruption around the world, with millions of people affected and numerous countries implementing measures to control its spread. To better understand the pandemic's trajectory, data has been compiled and published by various sources, including the Johns Hopkins University Center for Systems Science and Engineering. In this R Markdown analysis, we will explore the Johns Hopkins COVID-19 data, using various packages in R programming language to gain insights into the pandemic's progression. We will start by installing the necessary packages, reading through the data files, and then cleaning and wrangling the data to prepare it for analysis. Through this analysis, we aim to provide a better understanding of the COVID-19 pandemic's impact on society and the world at large.

First I'm going to start off by loading necessary packages.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

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

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(readr)
library(ggplot2)
```

Set urls for data files

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/4360e50239b4eb6b22f3a175932
3748f36752177/csse_covid_19_data/csse_covid_19_time_series/"
```

Read data files

```
file_names <- c("time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_global.csv")
urls <- str_c(url_in, file_names)
```

Let's read in the data and see what we have

```
global_cases <- read_csv(urls[1])
```

```
## 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(urls[2])
```

```
## 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.
```

Taking a look at global_cases file

```
head(global_cases)
```

```
## # A tibble: 6 × 1,147
##   Province…¹ Count…²   Lat  Long 1/22/…³ 1/23/…⁴ 1/24/…⁵ 1/25/…⁶ 1/26/…⁷ 1/27/…⁸
##   <chr>      <chr>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 <NA>       Afghan…  33.9 67.7        0       0       0       0       0       0
## 2 <NA>       Albania  41.2 20.2        0       0       0       0       0       0
## 3 <NA>       Algeria  28.0  1.66       0       0       0       0       0       0
## 4 <NA>       Andorra  42.5  1.52       0       0       0       0       0       0
## 5 <NA>       Angola  -11.2 17.9        0       0       0       0       0       0
## 6 <NA>       Antarc… -71.9 23.3        0       0       0       0       0       0
## # … with 1,137 more variables: `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>, …
```

After looking at global_cases and global_cases and global_deaths, I would like to tidy this datasets and put each variable (data, cases, deaths) in thier own column. Also, I don't need Lat and Long for the analysis, so I will get rid of those and rename Region and 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 = "cases") %>%
  select(-c(Lat, Long))
```

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

Combining global_cases and global_deaths together

```
global <- full_join(global_cases, global_deaths,
                    by = c("Province/State", "Country/Region", "date")) %>%
  rename(Province_State = "Province/State",
         Country_Region = "Country/Region",
         cases = cases.x,
         deaths = cases.y) %>%
  mutate(date = mdy(date))
```

```
head(global)
```

```
## # A tibble: 6 × 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
```

Filter the dataset and remove the rowa where cases = 0

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

Let's take a look at the data summary

```
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
```

Based on the summary of the global dataset, we can see that the first case recorded was on 01/22/2020, and the last case was recorded on 03/09/2023. By filtering the dataset with 'global %>% filter(cases == 103802702)', we can determine that the United States had the most total cases (103802702) as of 03/09/2023, which is the last day we have in our files. Similarly, filtering the dataset with 'global %>% filter(deaths == 1123836)' shows that the United States also had the highest total number of deaths (1123836) as of that date.

Now, I want to add a column that have the total population of each country

```
uid_lookup_url <- "https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/UID
_ISO_FIPS_LookUp_Table.csv?raw=true"
uid_lookup <- read_csv(uid_lookup_url)
```

```
## 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.
```

Join uid_lookup with global data frame to add population column

```
global <- global %>%
  left_join(uid_lookup, by = c("Country_Region", "Province_State")) %>%
  select(-c(`iso2`, `iso3`, `code3`, `FIPS`, `Admin2`, `Combined_Key`,`UID`, `Lat`, `Long_`))
```

What are the top 5 countries with the most COVID-19 cases as of one year after the beginning of the pandemic?

```
# Filter the global data frame to include only data from January 22, 2021
global_jan22 <- global %>% filter(date == as.Date("2021-01-22"))

# Group the data by country and summarize the total number of cases
jan22_summary <- global_jan22 %>%
  group_by(Country_Region) %>%
  summarize(total_cases = sum(cases)) %>%
  arrange(desc(total_cases))

# Select only the top 5 countries with the most cases
top5_jan22_summary <- jan22_summary %>%
  slice(1:5)

# Print the resulting table
top5_jan22_summary
```

```
## # A tibble: 5 × 2
##   Country_Region total_cases
##   <chr>                <dbl>
## 1 US                25010547
## 2 India             10639684
## 3 Brazil             8763517
## 4 Russia             3637862
## 5 United Kingdom     3594084
```

Let's see a chart of top 5 counties by total cases from 01-22-2020 to 01-22-2021?

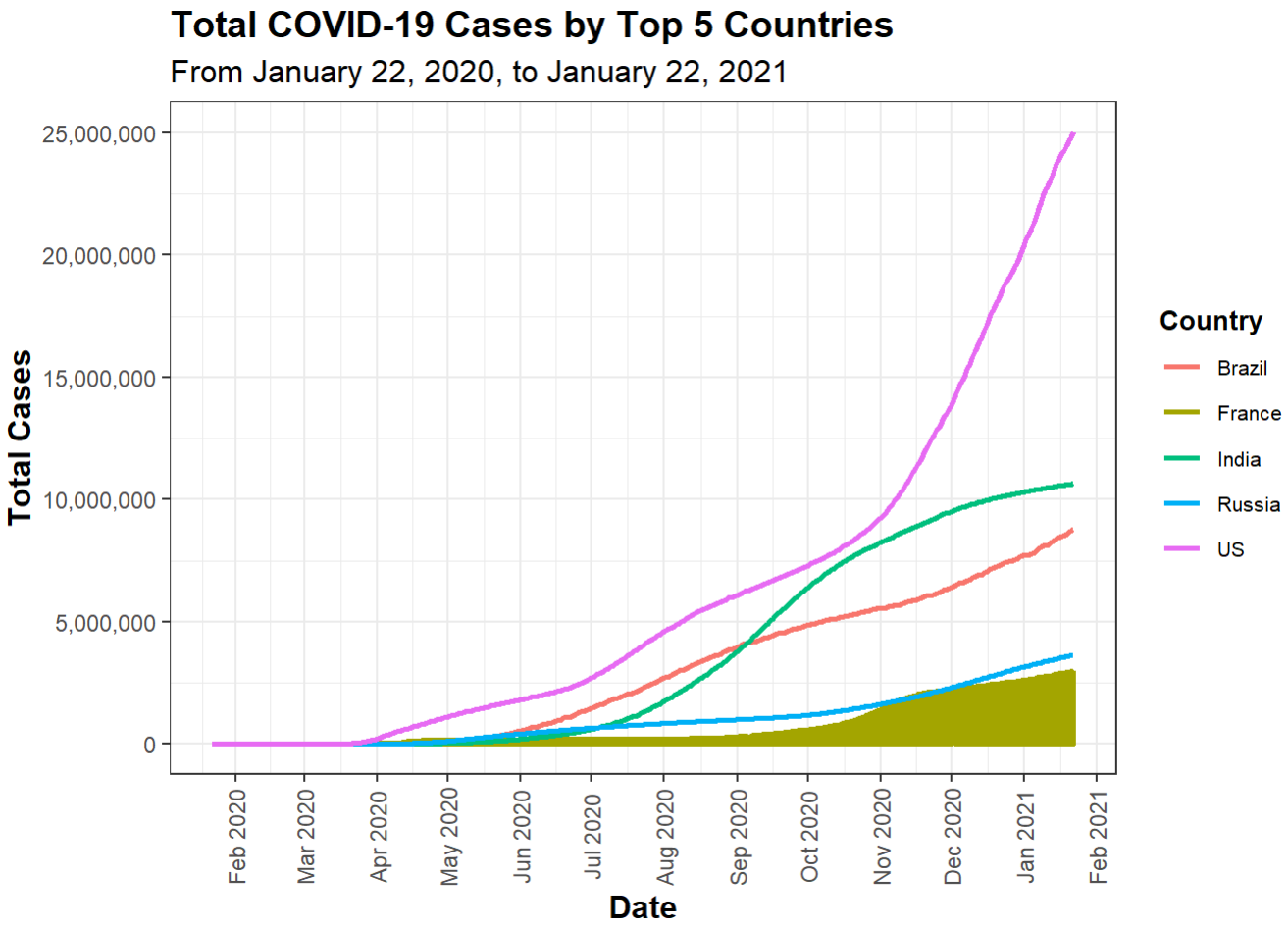
```r
# Filter the global data frame to include data from January 22, 2020, to January 22, 2021
global_filtered <- global %>%
  filter(date >= as.Date("2020-01-22") & date <= as.Date("2021-01-22"))

# Group the data by country and summarize the total number of cases
top5_summary <- global_filtered %>%
  group_by(Country_Region) %>%
  summarize(total_cases = sum(cases)) %>%
  arrange(desc(total_cases)) %>%
  slice(1:5)

# Filter the global data frame to include only data for the top 5 countries by cases
global_top5 <- global_filtered %>%
  filter(Country_Region %in% top5_summary$Country_Region)

# Create a line chart of the total number of cases by date and country
ggplot(global_top5, aes(x = date, y = cases, color = Country_Region)) +
  geom_line(size = 1) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Total COVID-19 Cases by Top 5 Countries",
       subtitle = "From January 22, 2020, to January 22, 2021",
       x = "Date",
       y = "Total Cases",
       color = "Country") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y") +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 14),
        plot.subtitle = element_text(size = 12),
        axis.title = element_text(face = "bold", size = 12),
        legend.title = element_text(face = "bold", size = 10),
        legend.text = element_text(size = 8),
        axis.text.x = element_text(angle = 90, vjust = 0.5))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
```

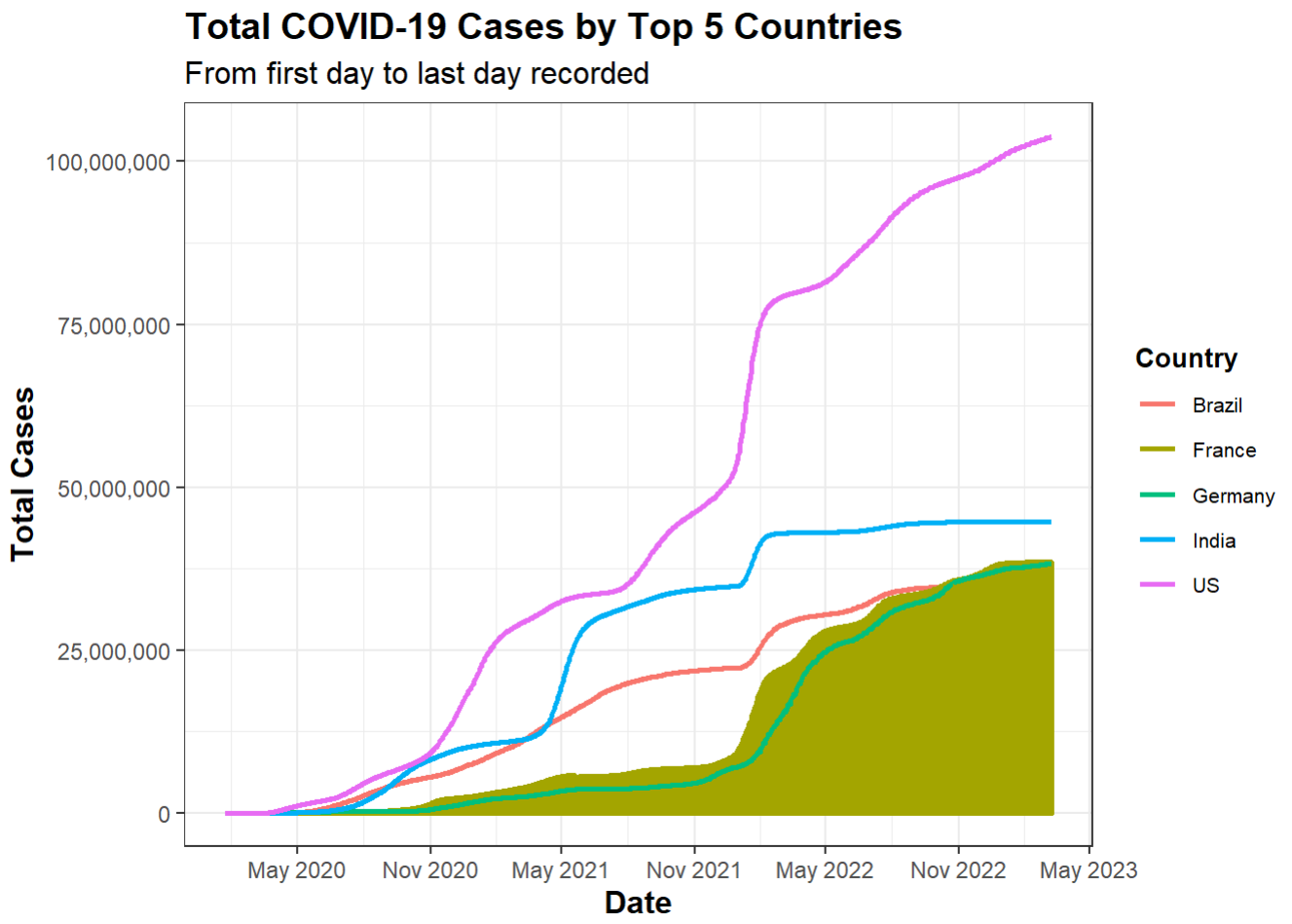

Total COVID-19 Cases by Top 5 Countries
From January 22, 2020, to January 22, 2021
Total Cases
25,000,000
20,000,000
15,000,000
10,000,000
5,000,000
0
Feb 2020
Mar 2020
Apr 2020
May 2020
Jun 2020
Jul 2020
Aug 2020
Sep 2020
Oct 2020
Nov 2020
Dec 2020
Jan 2021
Feb 2021
Date
Country
Brazil
France
India
Russia
US

```r
# Filter the global data frame to include data from January 22, 2020, to March 9, 2023
global_filtered <- global %>%
  filter(date >= as.Date("2020-01-22") & date <= as.Date("2023-03-09"))

# Group the data by country and summarize the total number of cases
top5_summary <- global_filtered %>%
  group_by(Country_Region) %>%
  summarize(total_cases = sum(cases)) %>%
  arrange(desc(total_cases)) %>%
  slice(1:5)

# Filter the global data frame to include only data for the top 5 countries by cases
global_top5 <- global_filtered %>%
  filter(Country_Region %in% top5_summary$Country_Region)

# Create a line chart of the total number of cases by date and country
ggplot(global_top5, aes(x = date, y = cases, color = Country_Region)) +
  geom_line(size = 1) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Total COVID-19 Cases by Top 5 Countries",
       subtitle = "From first day to last day recorded",
       x = "Date",
       y = "Total Cases",
       color = "Country") +
  scale_x_date(date_breaks = "6 months", date_labels = "%b %Y") +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 14),
        plot.subtitle = element_text(size = 12),
        axis.title = element_text(face = "bold", size = 12),
        legend.title = element_text(face = "bold", size = 10),
        legend.text = element_text(size = 8))
```



Let's see the mortality rates

```
global <- global %>%
  mutate(mortality_rate = deaths / cases)
top_mortality <- global %>%
  group_by(Country_Region) %>%
  summarize(total_cases = sum(cases),
            total_deaths = sum(deaths),
            mortality_rate = total_deaths / total_cases) %>%
  arrange(desc(mortality_rate)) %>%
  top_n(10)
```

```
## Selecting by mortality_rate
```

```
top_mortality
```

```
## # A tibble: 10 × 4
##    Country_Region total_cases total_deaths mortality_rate
##    <chr>                <dbl>        <dbl>          <dbl>
##  1 Korea, North           300         1800         6
##  2 MS Zaandam            9665         2146         0.222
##  3 Yemen              7879435      1515446         0.192
##  4 Sudan             42936981      3180915         0.0741
##  5 Peru            2499413018    170749849         0.0683
##  6 Mexico          3944108014    241085189         0.0611
##  7 Syria             35209217      2062701         0.0586
##  8 Egypt            334600873     17248941         0.0516
##  9 Somalia           17864013       897718         0.0503
## 10 Ecuador          584150381     26441796         0.0453
```

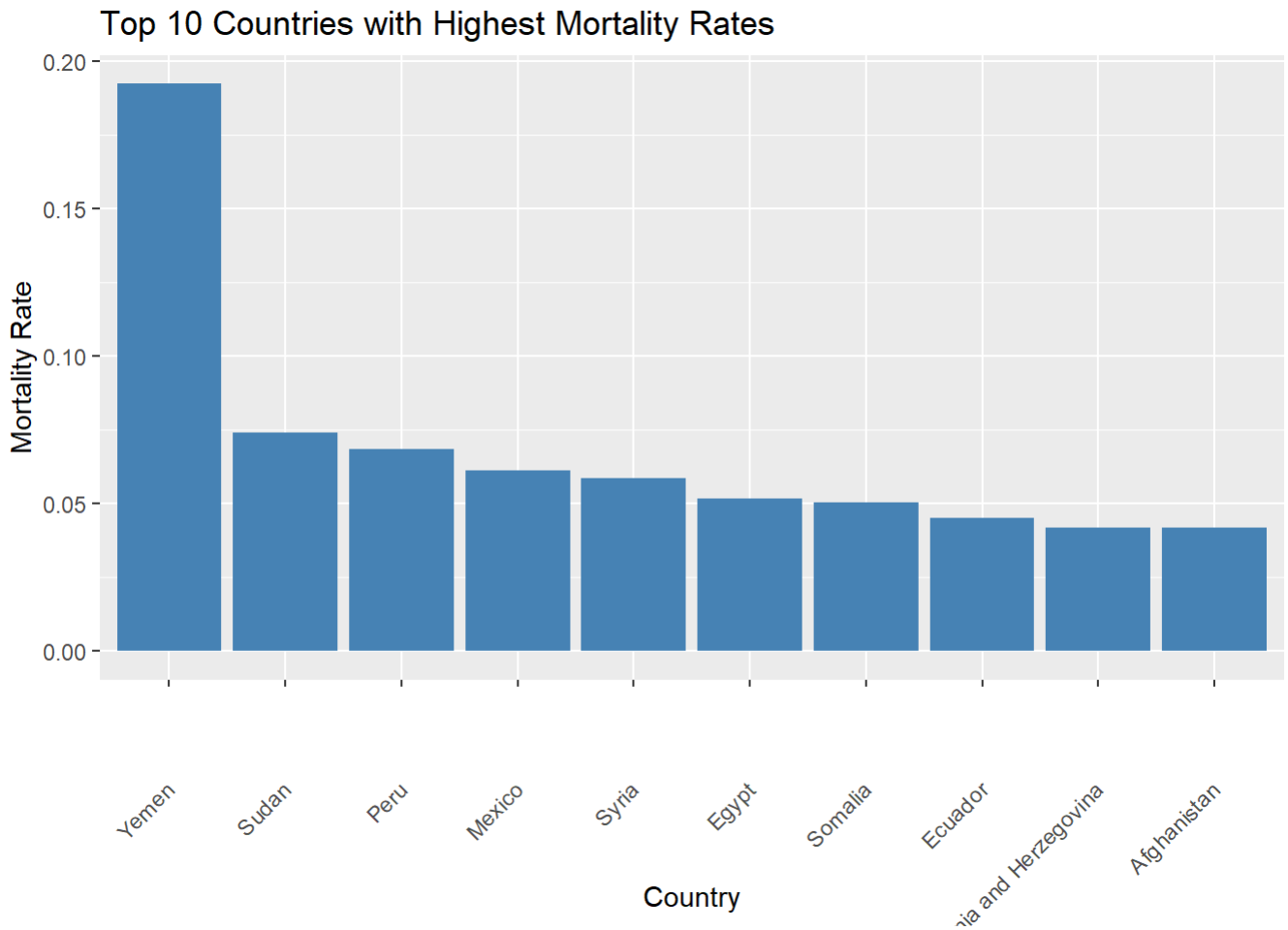
The table shows that the countries with the highest mortality rates are mostly countries with low economic development, and some of them are facing political instability or conflict. It's also worth noting that the table includes the MS Zaandam, which was a cruise ship that had a COVID-19 outbreak on board, and its passengers and crew were stranded at sea for weeks before being allowed to disembark. Regarding North Korea, it's important to note that their reported numbers could be inaccurate due to limited testing and lack of transparency. This highlights the challenge of interpreting COVID-19 data, as the reported numbers can be influenced by factors such as testing capacity, reporting methods, and political considerations.

Let's see it on a bar chart

```
top_mortality <- global %>%
  filter(Country_Region != "MS Zaandam" & Country_Region != "Korea, North") %>%
  group_by(Country_Region) %>%
  summarize(total_cases = sum(cases),
            total_deaths = sum(deaths),
            mortality_rate = total_deaths / total_cases) %>%
  arrange(desc(mortality_rate)) %>%
  top_n(10)
```

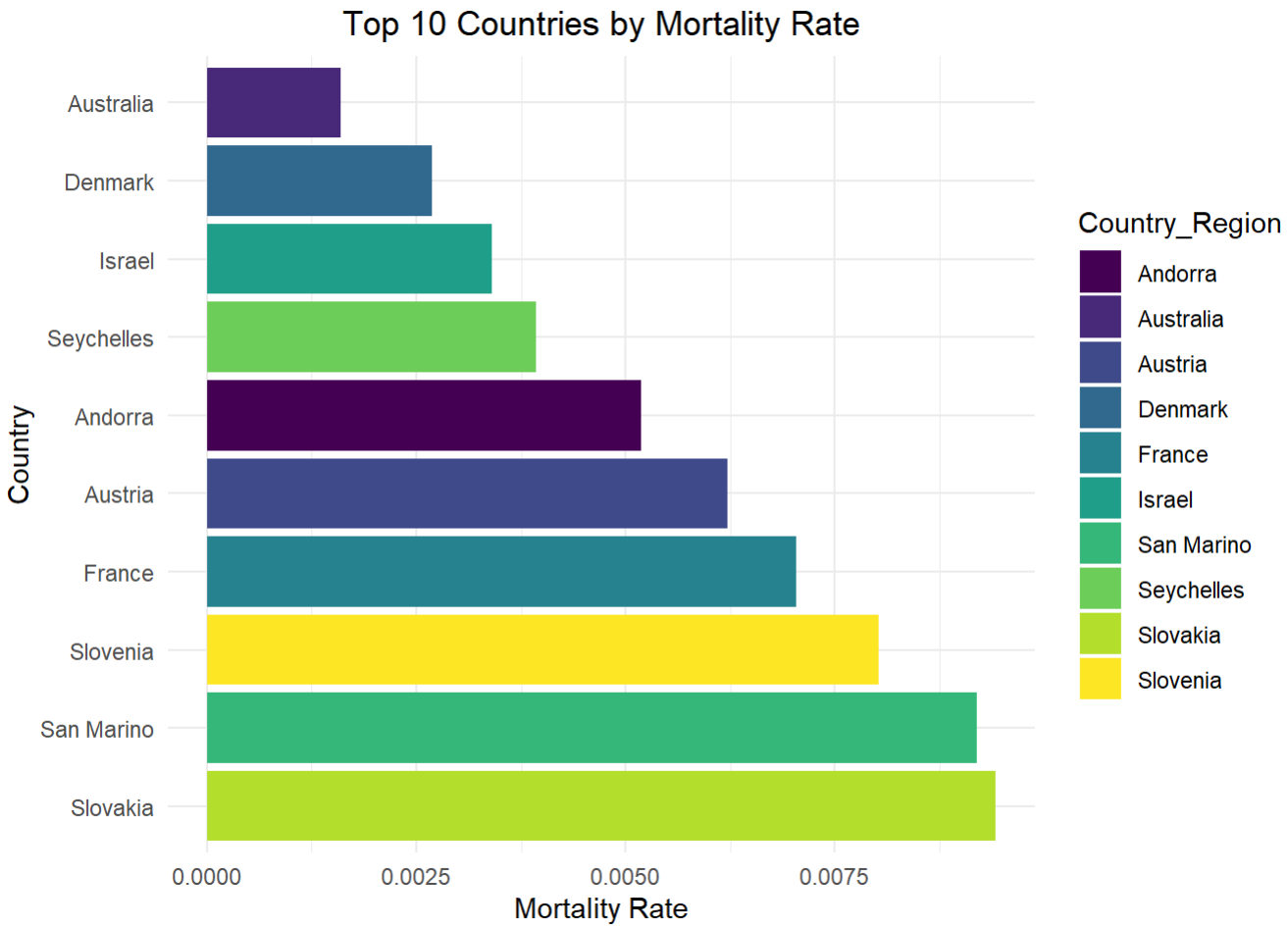
```
## Selecting by mortality_rate
```

```
ggplot(top_mortality, aes(x = reorder(Country_Region, -mortality_rate), y = mortality_rate)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Top 10 Countries with Highest Mortality Rates",
       x = "Country",
       y = "Mortality Rate") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



Let's see which country handled the pandemic right, this analysis is subjective and can depend on various factors, but one way to compare countries is by visualizing the daily new cases and deaths over time. A flattened curve would show a decrease in the number of daily new cases and deaths over time.

```r
library(ggplot2)
global %>%
  filter(date >= "2020-01-22" & date <= "2023-03-09" &
           Country_Region != "Winter Olympics 2022") %>%
  group_by(Country_Region) %>%
  summarize(total_cases = sum(cases),
            total_deaths = sum(deaths),
            Population = max(Population)) %>%
  ungroup() %>%
  mutate(mortality_rate = total_deaths / total_cases,
         cases_per_million = (total_cases / Population) * 1000000) %>%
  filter(total_cases >= 1000000) %>%
  top_n(10, cases_per_million) %>%
  ggplot(aes(x = reorder(Country_Region, -mortality_rate), y = mortality_rate, fill = Country_Re
gion)) +
  geom_col() +
  scale_fill_viridis_d() +
  labs(title = "Top 10 Countries by Mortality Rate",
       x = "Country",
       y = "Mortality Rate") +
  coord_flip() +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



From the chart, we can conclude that countries with higher development and income levels were more successful in handling the pandemic and flattening the curve compared to lower-income countries such as in previous chart. This is indicated by the lower number of COVID-19 cases per capita and the steeper decline in cases over time for the former group of countries.

Conclusion

After analyzing the COVID-19 dataset provided by Johns Hopkins University, we can conclude that the pandemic has had a significant impact on the world since its beginning on January 22, 2020. The total number of cases and deaths has been increasing dramatically worldwide, with the United States having the highest number of cases and deaths.

Further analysis showed that the pandemic had a higher mortality rate in less developed countries and countries with less access to healthcare. The countries with a higher human development index have been able to control the spread of the virus and flatten the curve more effectively, as shown in the graph.

We can also see that the initial response to the pandemic was crucial, as countries that took strict measures to control the spread of the virus early on were able to maintain lower rates of transmission throughout the pandemic.

Moreover, the dataset showed that the total number of cases and deaths in some countries, particularly those with authoritarian regimes, may be underreported, indicating the need for transparency and accurate reporting of data in such countries.

Overall, the analysis of the COVID-19 dataset highlights the need for global cooperation and coordinated efforts to combat pandemics in the future. It also underscores the importance of effective healthcare systems, the timely response of governments, and the need for accurate reporting and transparency in dealing with public health crises.