COVID-19 Data Report

2025-06-24

Contents

1 Required Packages & Libraries

```
#install.packages(c("tidyverse", "ggthemes", "lubridate", "countrycode", "choroplethr"))
library("tidyverse")
library("ggthemes")
library("lubridate")
library("countrycode")
library("choroplethr")
```

Note: Uncomment by removing “#” to use.

2 Introduction

This report analyzes the COVID-19 time series data provided by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, focusing on the global progression and geographic distribution of confirmed cases and deaths. While the repository includes both U.S. and global datasets, this analysis concentrates primarily on the global datasets to better understand global spread.The analysis begins with the data pre-processing steps, which includes a brief data exploration and data cleaning steps. We then build and evaluation our model using the following approaches: the first, is a geo-spatial mapping visualizing case fatality rates (CFR), we also look at potentially related economic factors; the second approach is a severity index comparing case fatality rates (CFR) across different countries. Next, we discuss potential challenges and limitations faced throughout the project, such as bias, assumptions, data quality concerns, and interpretative limitations. The report concludes by reflecting on key insights derived from the analytical process, potentials recommendations, and suggestions for future works.

Questions of Interest:

- Approach 1 (Geo-Spatial): What does the geographic distribution of global case fatality rates reveal about the spread and impact of COVID-19?

- Approach 2 (Severity Index): Which countries have the highest and lowest case fatality rates (CFR) from COVID-19?

2.1 Data Description

This project uses data from the COVID-19 Data Repository maintained by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The repository compiles global and geo-specific pandemic data from a wide range of official sources and was active managed from January 2020 to March 10, 2023, at which point data collection ceased.

- **US Data:** Confirmed cases (county level), Reported deaths (county level)
- **Global Data:** Confirmed cases (country/province level), Reported deaths (country/province level), Reported recoveries (country/province level)

US data was sourced from the Centers for Disease Control and Prevention (CDC) and individual U.S. state and county public health departments. Global data was aggregated from the World Health Organization (WHO) and regional health ministries worldwide (ie.European Centre for Disease Prevention and Control (ECDC), etc.).

Data Source: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

3 Methodology

This section of the report focuses on an initial exploration of the dataset, including examining its structure, generating summary statistics, and conducting a concise preliminary analysis. The data is then pre-processed and prepared for modeling by removing irrelevant columns, correcting data types, eliminating duplicates, addressing missing values, engineering features, and renaming columns for improved clarity.

3.1 Import Dataset

```r
#Defining base url for the files we want to access
base_url <- paste0("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/",
                   "csse_covid_19_data/csse_covid_19_time_series/")

# Read each CSV file
us_confirmed <- read.csv(paste0(base_url, "time_series_covid19_confirmed_US.csv"))
us_deaths <- read.csv(paste0(base_url, "time_series_covid19_deaths_US.csv"))
global_confirmed <- read.csv(paste0(base_url, "time_series_covid19_confirmed_global.csv"))
global_deaths <- read.csv(paste0(base_url, "time_series_covid19_deaths_global.csv"))
global_recoveries <- read.csv(paste0(base_url, "time_series_covid19_recovered_global.csv"))
```

3.2 Data Integration

In this sections, we're going to merge the related datasets.

Prior to merging, let's pivot the date columns.

```r
#Pivots date columns for affected datasets
us_confirmed <- us_confirmed %>%
  pivot_longer(
    cols = starts_with("X"),
    names_to = "date",
    values_to = "cases") %>%
  mutate(
    date = mdy(str_remove(date, "^X")))

us_deaths <- us_deaths %>%
  pivot_longer(
    cols = starts_with("X"),
    names_to = "date",
    values_to = "deaths") %>%
  mutate(
    date = mdy(str_remove(date, "^X")))

global_confirmed <- global_confirmed %>%
  pivot_longer(
    cols = starts_with("X"),
    names_to = "date",
    values_to = "cases") %>%
  mutate(
    date = mdy(str_remove(date, "^X")))

global_deaths <- global_deaths %>%
  pivot_longer(
    cols = starts_with("X"),
    names_to = "date",
    values_to = "deaths") %>%
  mutate(
    date = mdy(str_remove(date, "^X")))

global_recoveries <- global_recoveries %>%
```

```r
  pivot_longer(
    cols = starts_with("X"),
    names_to = "date",
    values_to = "recoveries") %>%
  mutate(
    date = mdy(str_remove(date, "^X")))
```

Note: This is done because individual datesa are marked as separate columns in the original datasets.

```r
glimpse(global_confirmed)
```

```
## Rows: 330,327
## Columns: 6
## $ Province.State <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ Country.Region <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanist~
## $ Lat            <dbl> 33.93911, 33.93911, 33.93911, 33.93911, 33.93911, 33.93~
## $ Long           <dbl> 67.70995, 67.70995, 67.70995, 67.70995, 67.70995, 67.70~
## $ date           <date> 2020-01-22, 2020-01-23, 2020-01-24, 2020-01-25, 2020-0~
## $ cases          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Merging US datasets . . .

```r
#Confirms the dimensions before merging
dim(us_confirmed)
```

```
## [1] 3819906      13
```

```r
dim(us_deaths)
```

```
## [1] 3819906      14
```

```r
#Merges the us_confirmed and us_deaths datasets
us <- us_confirmed %>%
  full_join(us_deaths)
```

```
## Joining with ‘by = join_by(UID, iso2, iso3, code3, FIPS, Admin2,
## Province_State, Country_Region, Lat, Long_, Combined_Key, date)‘
```

```r
#Checks the new "us" dataset
head(us)
```

```
## # A tibble: 6 x 15
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <int> <chr> <chr> <int> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 3 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 4 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 5 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 6 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## # i 6 more variables: Long_ <dbl>, Combined_Key <chr>, date <date>,
## #   cases <int>, Population <int>, deaths <int>
```

Merging global datasets ...

```r
#Confirms the dimensions before merging (should be the same)
dim(global_confirmed)

## [1] 330327      6

dim(global_deaths)

## [1] 330327      6

dim(global_recoveries)

## [1] 313182      6

#Merges the global_confirmed, global_deaths, global_recoveries and
#global_pop datasets
global <- global_confirmed %>%
  full_join(global_deaths) %>%
  full_join(global_recoveries)

## Joining with `by = join_by(Province.State, Country.Region, Lat, Long, date)`
## Joining with `by = join_by(Province.State, Country.Region, Lat, Long, date)`

#Checks the new "global" dataset
head(global)

## # A tibble: 6 x 8
##   Province.State Country.Region   Lat  Long date       cases deaths recoveries
##   <chr>          <chr>          <dbl> <dbl> <date>     <int>  <int>      <int>
## 1 ""             Afghanistan     33.9  67.7 2020-01-22     0      0          0
## 2 ""             Afghanistan     33.9  67.7 2020-01-23     0      0          0
## 3 ""             Afghanistan     33.9  67.7 2020-01-24     0      0          0
## 4 ""             Afghanistan     33.9  67.7 2020-01-25     0      0          0
## 5 ""             Afghanistan     33.9  67.7 2020-01-26     0      0          0
## 6 ""             Afghanistan     33.9  67.7 2020-01-27     0      0          0
```

Note: Both datasets looks fine, so let's do a little exploration of the datasets

3.3 Data Exploration

First, let's look at the shape of our datasets...

```r
#Shows the # of row and columns in your dataset
dim(global)

[1] 337185      8

dim(us)
```

```
[1] 3819906      15
```

Next, let's take a take a glance at our datasets to learn a bit more

```
#Looks at the first row of each dataset
head(us,1)
```

```
## # A tibble: 1 x 15
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <int> <chr> <chr> <int> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## # i 6 more variables: Long_ <dbl>, Combined_Key <chr>, date <date>,
## #   cases <int>, Population <int>, deaths <int>
```

```
tail(us, 1)
```

```
## # A tibble: 1 x 15
##        UID iso2  iso3  code3  FIPS Admin2 Province_State Country_Region   Lat
##      <int> <chr> <chr> <int> <dbl> <chr>  <chr>          <chr>          <dbl>
## 1 84056045 US    USA     840 56045 Weston Wyoming        US              43.8
## # i 6 more variables: Long_ <dbl>, Combined_Key <chr>, date <date>,
## #   cases <int>, Population <int>, deaths <int>
```

```
head(global, 1)
```

```
## # A tibble: 1 x 8
##   Province.State Country.Region   Lat  Long date       cases deaths recoveries
##   <chr>          <chr>          <dbl> <dbl> <date>     <int>  <int>      <int>
## 1 ""             Afghanistan     33.9  67.7 2020-01-22     0      0          0
```

```
tail(global, 1)
```

```
## # A tibble: 1 x 8
##   Province.State Country.Region   Lat  Long date       cases deaths recoveries
##   <chr>          <chr>          <dbl> <dbl> <date>     <int>  <int>      <int>
## 1 ""             Timor-Leste    -8.87  126. 2023-03-09    NA     NA          0
```

```
#Shows summary statistics for each column in the dataset
summary(us)
```

```
##       UID               iso2               iso3               code3
##  Min.   :      16   Length:3819906     Length:3819906     Min.   : 16.0
##  1st Qu.:84018105   Class :character   Class :character   1st Qu.:840.0
##  Median :84029206   Mode  :character   Mode  :character   Median :840.0
##  Mean   :83429923                                         Mean   :834.5
##  3rd Qu.:84046119                                         3rd Qu.:840.0
##  Max.   :84099999                                         Max.   :850.0
##
##       FIPS          Admin2          Province_State     Country_Region
##  Min.   :   60   Length:3819906     Length:3819906     Length:3819906
##  1st Qu.:19076   Class :character   Class :character   Class :character
```

```
## Median :31012   Mode  :character   Mode  :character   Mode  :character
## Mean   :33043
## 3rd Qu.:47130
## Max.   :99999
## NA's   :11430
##      Lat             Long_          Combined_Key            date
## Min.   :-14.27   Min.   :-174.16   Length:3819906     Min.   :2020-01-22
## 1st Qu.: 33.90   1st Qu.: -97.81   Class :character   1st Qu.:2020-11-02
## Median : 38.01   Median : -89.49   Mode  :character   Median :2021-08-15
## Mean   : 36.72   Mean   : -88.64                      Mean   :2021-08-15
## 3rd Qu.: 41.58   3rd Qu.: -82.31                      3rd Qu.:2022-05-28
## Max.   : 69.31   Max.   : 145.67                      Max.   :2023-03-09
##
##     cases            Population           deaths
## Min.   :  -3073   Min.   :       0   Min.   :  -82.0
## 1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
## Median :   2272   Median :   24892   Median :   37.0
## Mean   :  14088   Mean   :   99604   Mean   :  186.9
## 3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
## Max.   :3710586   Max.   :10039107   Max.   :35545.0
##
```

```
summary(global)
```

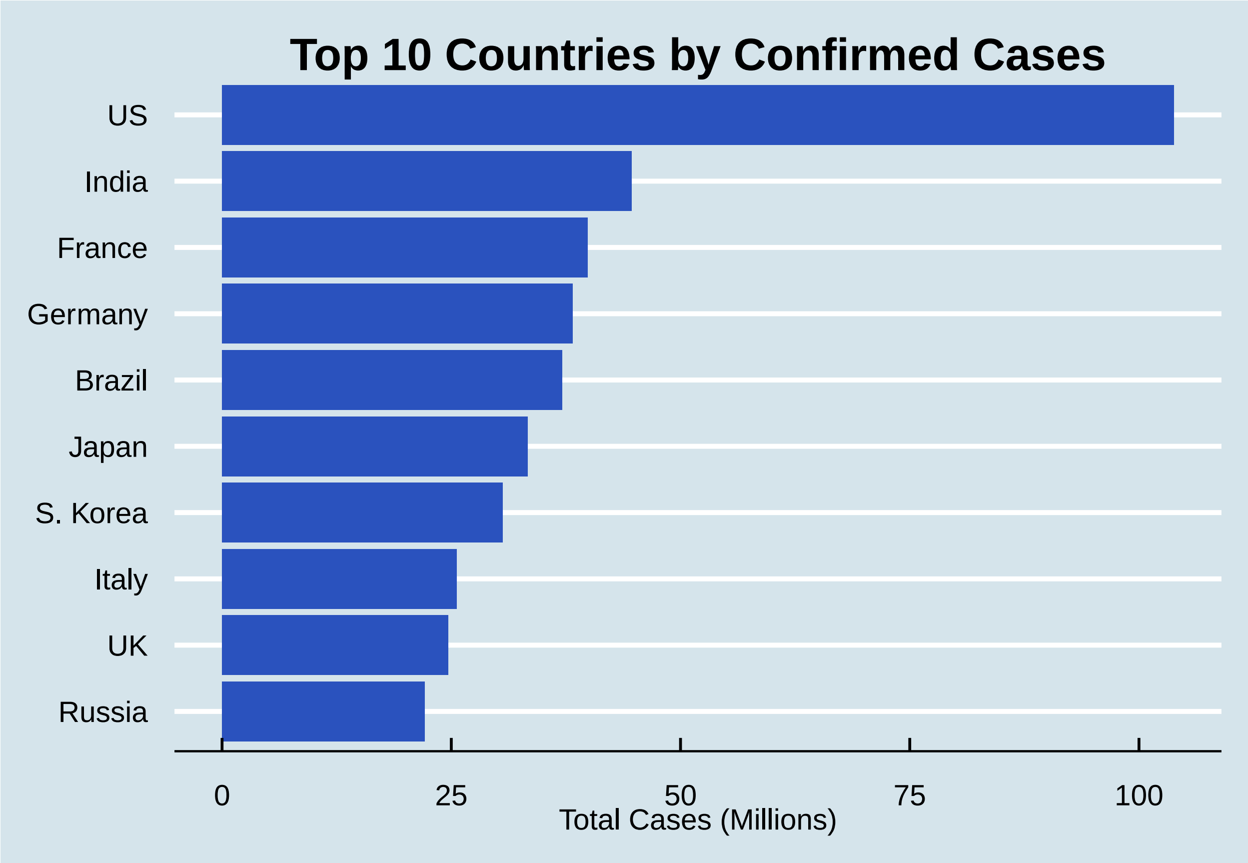
```
## Province.State     Country.Region          Lat               Long
## Length:337185      Length:337185      Min.   :-71.950   Min.   :-178.12
## Class :character   Class :character   1st Qu.:  3.934   1st Qu.: -23.04
## Mode  :character   Mode  :character   Median : 21.522   Median :  21.75
##                                       Mean   : 19.776   Mean   :  22.83
##                                       3rd Qu.: 40.340   3rd Qu.:  90.43
##                                       Max.   : 71.707   Max.   : 178.06
##                                       NA's   :2286      NA's   :2286
##      date                cases               deaths            recoveries
## Min.   :2020-01-22   Min.   :        0   Min.   :      0   Min.   :      -1
## 1st Qu.:2020-11-02   1st Qu.:      680   1st Qu.:      3   1st Qu.:       0
## Median :2021-08-15   Median :    14429   Median :    150   Median :       0
## Mean   :2021-08-15   Mean   :   959384   Mean   :  13380   Mean   :   75009
## 3rd Qu.:2022-05-28   3rd Qu.:   228517   3rd Qu.:   3032   3rd Qu.:     934
## Max.   :2023-03-09   Max.   :103802702   Max.   :1123836   Max.   :30974748
##                      NA's   :6858        NA's   :6858      NA's   :24003
```

Now, let's perform a simple analysis before processing the data.

```
#Compiles top 10 countries by total cases
top10_confirm <- global %>%
  filter(date == max(date)) %>%
  group_by(Country.Region) %>%
  summarise(total_cases = sum(cases)/1000000) %>%
   mutate(Country.Region =
            recode(Country.Region,
            "United Kingdom" = "UK",
            "Korea, South" = "S. Korea")) %>%
  arrange(desc(total_cases)) %>%
```

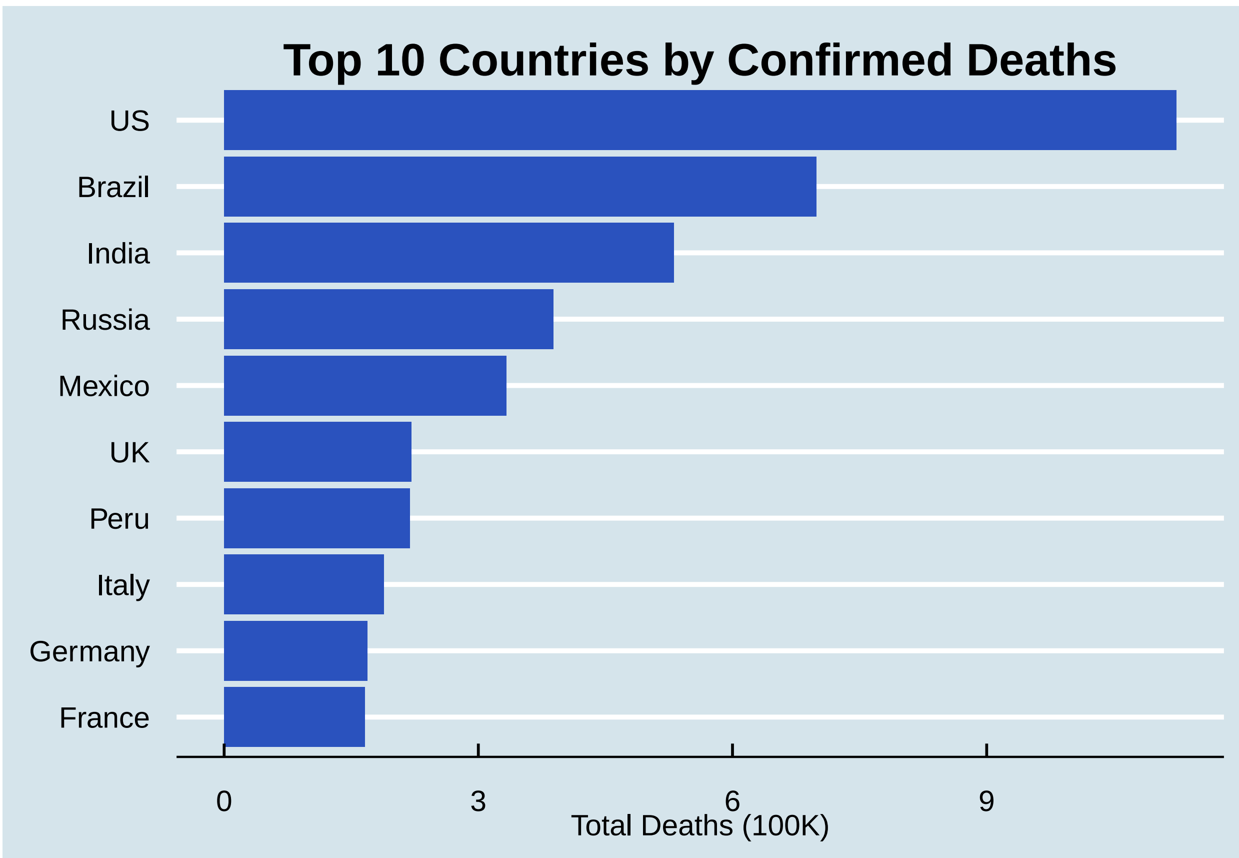
```r
  slice_head(n = 10)

#Shows bar plots of top 10 countries by total cases
ggplot(top10_confirm, aes(x = reorder(Country.Region, total_cases), y = total_cases)) +
  geom_bar(stat = "identity", fill = "#2a52be") +
  coord_flip() +
  labs(title = "Top 10 Countries by Confirmed Cases",
       x = NULL,
       y = "Total Cases (Millions)") +
  theme_economist() +
  theme(
      axis.text.y = element_text(angle = 0, hjust = 1),
      plot.title = element_text(hjust = 0.5))
```




```r
#Compiles top 10 countries by total deaths
top10_deaths<- global %>%
  filter(date == max(date)) %>%
  group_by(Country.Region) %>%
  summarise(total_deaths = sum(deaths)/100000) %>%
  mutate(Country.Region =
           ifelse(Country.Region ==
           "United Kingdom", "UK", Country.Region)) %>%
  arrange(desc(total_deaths)) %>%
  slice_head(n = 10)
```

```r
#Shows bar plots of top 10 countries by total deaths
ggplot(top10_deaths, aes(x = reorder(Country.Region, total_deaths), y = total_deaths)) +
  geom_bar(stat = "identity", fill = "#2a52be") +
  coord_flip() +
  labs(title = "Top 10 Countries by Confirmed Deaths",
       x = NULL,
       y = "Total Deaths (100K)") +
  theme_economist() +
  theme(
      axis.text.y = element_text(angle = 0, hjust = 1),
       plot.title = element_text(hjust = 0.5))
```



Note: Looking at the charts, we observe that the US had the highest number of reported COVID-19 cases and deaths. This could be potentially attributed to policy decisions made during the early stages of the pandemic, but it may also reflect the country's large population, given higher raw numbers are expected in more populous nations. However, when we compare this to India, which has a population over three times that of the US, its reported case and death counts are significantly lower. At first glance, this might suggest more effective pandemic management. Yet, it also raises questions about the accuracy of the data. Limited access to testing and underreporting may have contributed to lower official figures, meaning the true scale of the outbreak could be much larger than reported.

Overall, it's difficult to determine exactly why the numbers appear as they do, so one must be careful not to jump to conclusions, as there could be influencing factors that are unaccounted for. Also, I must state that correlation != causation.

3.4 Data Cleaning

In this segment of the report, we're going to get the data ready for modeling. Specifically, we'll be doing the following:

- Drop irrelevant columns
- Check for duplicate rows
- Check & deal w/ missing values
- Refactor column names
- Feature Engineering

For the cleaning steps, we'll focus solely on the `global` dataset, which will be used for the subsequent analyses.

- Drop irrelevant columns

```
glimpse(global)
```

```
## Rows: 337,185
## Columns: 8
## $ Province.State <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ Country.Region <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanist~
## $ Lat            <dbl> 33.93911, 33.93911, 33.93911, 33.93911, 33.93911, 33.93~
## $ Long           <dbl> 67.70995, 67.70995, 67.70995, 67.70995, 67.70995, 67.70~
## $ date           <date> 2020-01-22, 2020-01-23, 2020-01-24, 2020-01-25, 2020-0~
## $ cases          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ deaths         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ recoveries     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
#Removed column(s) irrelevant to planned analysis
global <- subset(global, select = -c(recoveries))

#Eliminates rows with case counts equaling zero
global <- global %>%
  filter(cases != 0)
```

Note: This reduces clutter, and allows your code to run faster

- Check for duplicate rows

```
sum(duplicated(global))
```

```
## [1] 0
```

Note: No duplicate rows. No further steps required.

- Check & deal w/ missing values

```
colSums(is.na(global))
```

```
## Province.State Country.Region            Lat           Long           date
##              0              0           1910           1910              0
##          cases         deaths
##              0              0
```

```r
#Shows total number of missing values
sum(is.na(global$Lat))
```

```
## [1] 1910
```

```r
sum(is.na(global$Long))
```

```
## [1] 1910
```

```r
#Compute mean values for Latitude and Longitude by BORO
mean_lat_geo <- global %>%
  group_by(Country.Region) %>%
  summarise(mean_lat = mean(Lat, na.rm = TRUE)) #Finds mean Latitude by Country

mean_lon_geo <- global %>%
  group_by(Country.Region) %>%
  summarise(mean_lon = mean(Long, na.rm = TRUE)) #Finds mean Longitude by Country

#Impute missing value using calculated means
global <- global %>%
  left_join(mean_lat_geo, by = "Country.Region") %>%
  mutate(Lat = ifelse(is.na(Lat), mean_lat, Lat)) %>%
  select(-mean_lat) #Imputes missing Latitude values with means from prev. computations

global <- global %>%
  left_join(mean_lon_geo, by = "Country.Region") %>%
  mutate(Long = ifelse(is.na(Long), mean_lon, Long)) %>%
  select(-mean_lon) #Imputes missing Longitude values with means from prev. computations
```

- Refactor column names

```r
#Renaming variables to match existing variable naming convention (screaming snake case)
global <- global %>%
  rename(
    prov_state = Province.State,
    country_region = Country.Region,
    latitude = Lat,
    longitude = Long)
```

- Feature Engineering

Let's check the column data types.

```r
#Shows information about the # of rows and columns, columns labels along
#with their data types and contents
glimpse(global)
```

```
## Rows: 306,827
## Columns: 7
## $ prov_state     <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ country_region <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanist~
## $ latitude       <dbl> 33.93911, 33.93911, 33.93911, 33.93911, 33.93911, 33.93~
## $ longitude      <dbl> 67.70995, 67.70995, 67.70995, 67.70995, 67.70995, 67.70~
## $ date           <date> 2020-02-24, 2020-02-25, 2020-02-26, 2020-02-27, 2020-0~
## $ cases          <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 8, 8, 8, 8, 11, 11,~
## $ deaths         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Note: Data types look fine. No need to change.

```
#Lists non-country or territory entries
non_countries <- c(
  "Diamond Princess",
  "MS Zaandam",
  "Summer Olympics 2020",
  "Winter Olympics 2022")

#Filters out the non-country rows
global <- global %>%
  filter(!country_region %in% non_countries)

#Creates case fatality rate (CFR) column
global <- global %>%
  mutate(case_fatal_rate = (deaths / cases) * 100)

#Add ISO3 country codes to your dataset
global <- global %>%
  mutate(country_code = countrycode(`country_region`,
                              origin = 'country.name',
                              destination = 'iso3c')) %>%
  mutate(country_code = case_when(`country_region` == "Kosovo" ~ "XKX",
                                  `country_region` == "Micronesia" ~ "FSM",
                                  TRUE ~ country_code))

unmatched <- global %>% filter(is.na(country_code)) %>% distinct(`country_region`)

print(unmatched)

## # A tibble: 0 x 1
## # i 1 variable: country_region <chr>
```

4 Model Evaluaion

In this segment we aim to understand the impact of COVID-19 across global communities. To quantify this impact we computed case fatality rate, referred to as CFR for short, which is quotient of the total reported deaths and total reported cases (see formula below). We then map the geospatial distribution of CFR by country and compare it to economic data provided by the International Monetary Fund (IMF); and visualize the most and least severely impacted countries and try to investigate possible influencing factor. Finally, we build our analysis by interpreting the resulting visualization and determine what they communicate to us about the global impact of the pandemic.

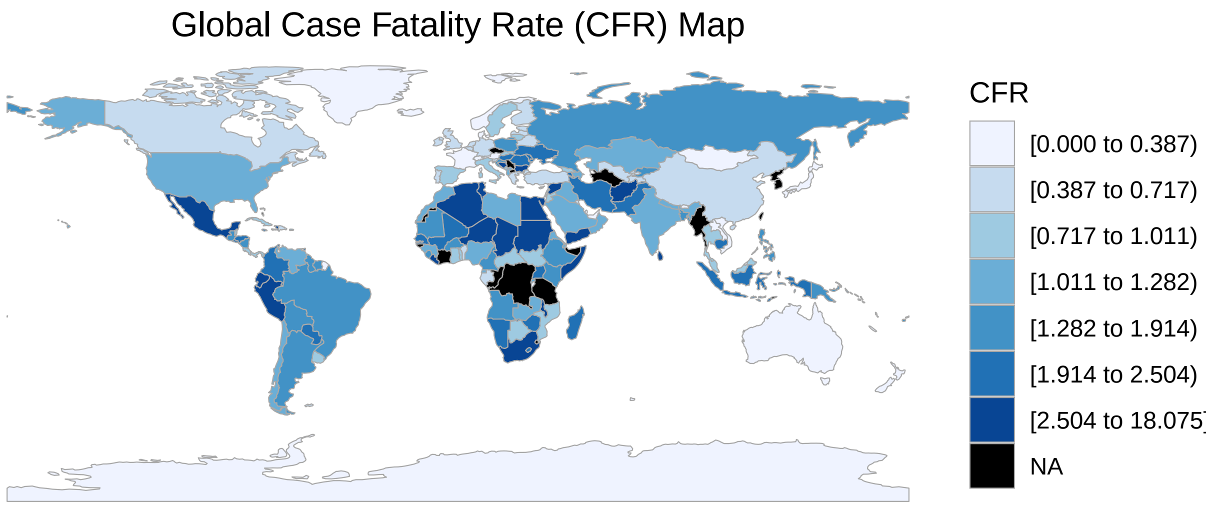
$$CFR = \frac{\text{Deaths}}{\text{Confirmed Cases}} \tag{1}$$

4.1 Geo-Spatial Mapping: Global Case Fatality Rate and Economic Indicators

```
#Filter for latest date
latest_date <- max(global$date, na.rm = TRUE)

#Prepare data for choropleth
global_choro_data <- global %>%
  filter(date == latest_date) %>%
  group_by(country_region) %>%
  summarize(value = mean(case_fatal_rate, na.rm = TRUE)) %>%
  ungroup() %>%
  rename(region = country_region) %>%
  mutate(region = tolower(region)) %>%
  mutate(region = case_when(
    region == "us" ~ "united states of america",
    region == "usa" ~ "united states of america",
    region == "united states" ~ "united states of america",
    TRUE ~ region
  ))

#Plot map
country_choropleth(global_choro_data,
                   title = "Global Case Fatality Rate (CFR) Map",
                   legend = "CFR",
                   num_colors = 7) +
  theme(plot.title = element_text(hjust = 0.5))
```



Analysis

Looking at the global map of avg. case fatality rates (CFR), we observe that countries in Eastern Europe, Africa, South and Central America, and parts of Asia tended to have higher fatality rates. To explore a possible explanation, I compared these CFR patterns to economic indicators, specifically GDP per capita. When examining GDP per capita maps from the International Monetary Fund (IMF) between 2020 and 2022, one can notice a geographic distribution that resembles that of CFR. This suggests that wealthier nations, which generally have higher GDP per capita, may have been better equipped to respond to the pandemic through more accessible healthcare systems, implementing policy, earlier interventions and treatment availibility. While correlation does not imply causation, this parallel supports the broader notion that economic capacity plays a role in a country's ability to manage epidemic health crises.

IMF Data: https://www.imf.org/external/datamapper/NGDPDPC@WEO/OEMDC/ADVEC/WEOWORLD

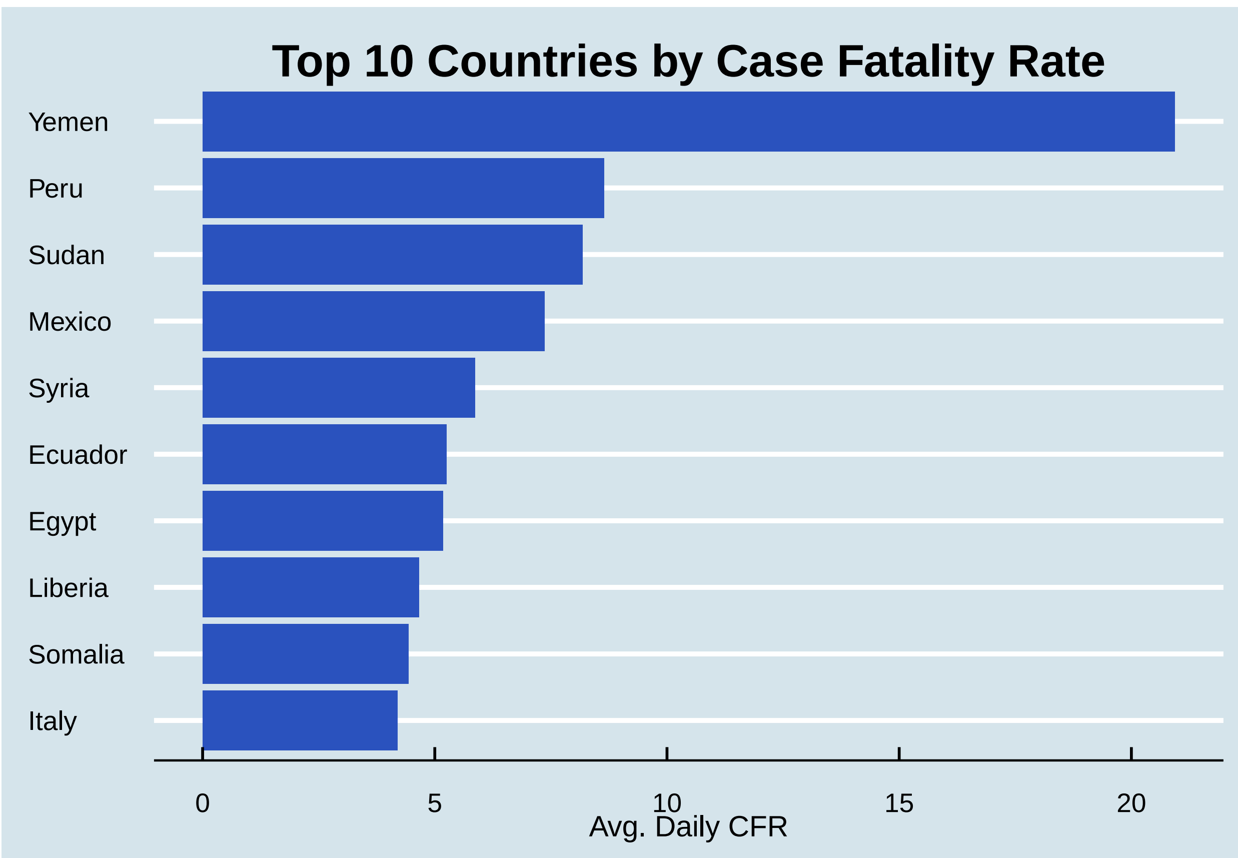
4.2 Severity Index: Global Case Fatality Rate

```
# Load required libraries

# Step 1: Calculate average daily CFR per country (excluding NA and infinite)
daily_cfr_avg <- global %>%
  filter(!is.na(case_fatal_rate),
         is.finite(case_fatal_rate)) %>%
  group_by(country_region) %>%
  summarize(avg_case_fatal_rate = mean(case_fatal_rate, na.rm = TRUE)) %>%
  ungroup() %>%
```

```r
  filter(country_region != "Korea, North", #numbers do not seem supported, suppress for now
         country_region != "Vanuatu") %>%
  arrange(desc(avg_case_fatal_rate)) %>%
  slice_head(n = 10)

# Step 2: Plot the bar chart
ggplot(daily_cfr_avg, aes(x = reorder(country_region, avg_case_fatal_rate), y = avg_case_fatal_rate)) +
  geom_col(fill = "#2a52be") +
  coord_flip() +
  labs(
    title = "Top 10 Countries by Case Fatality Rate",
    x = NULL,
    y = "Avg. Daily CFR") +
  theme_economist() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.text = element_text(size = 10))
```

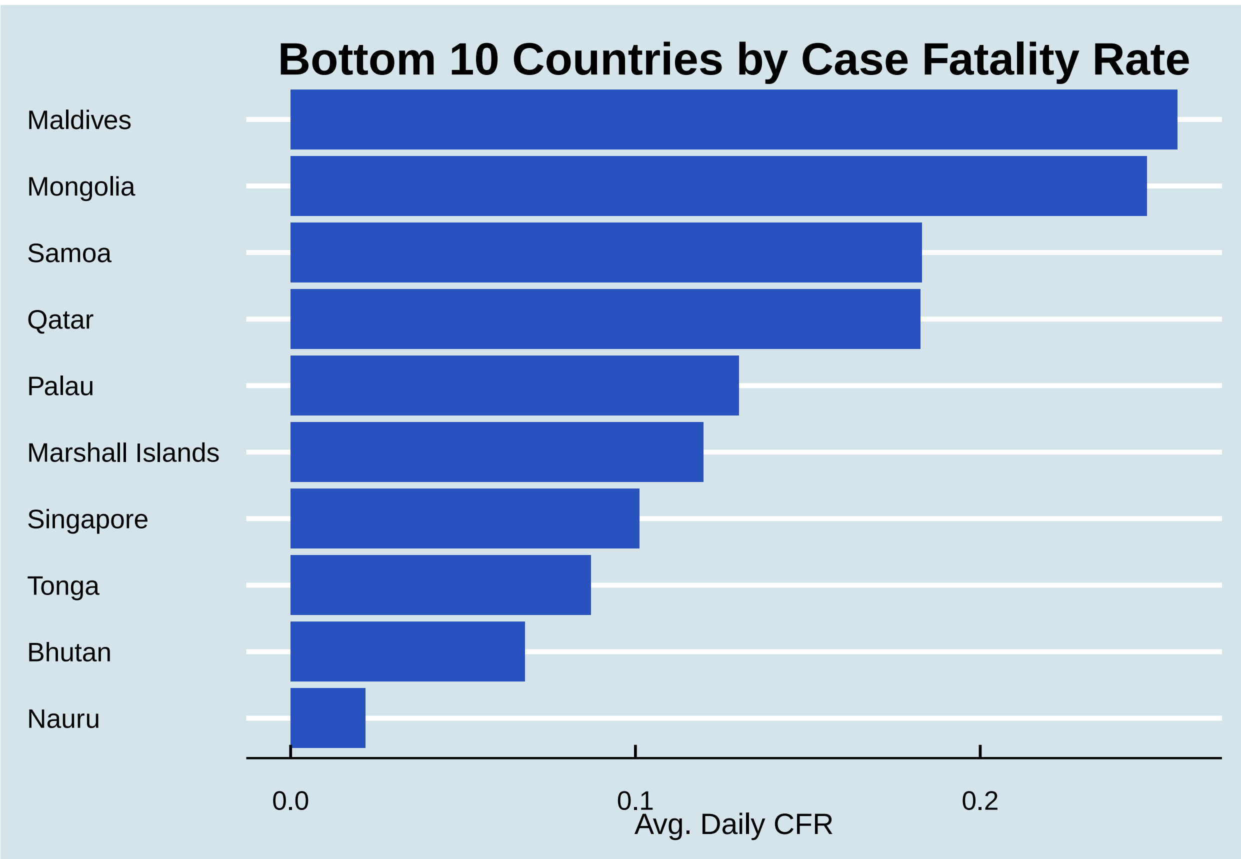


Note: Removed North Korea from the top 10, because CFR was abnormally high. For context, it was more than 30 times higher than Yemen's CFR (which is the new top country). Unaware what caused this extremely high CFR, so I opted to remove it from the graph.

```r
#Computes average daily CFR per country (excluding NA, 0s)
bottom10 <- global %>%
  filter(!is.na(case_fatal_rate),
         case_fatal_rate > 0,
```

```
         is.finite(case_fatal_rate)) %>%
  group_by(country_region) %>%
  summarize(avg_cfr = mean(case_fatal_rate, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(avg_cfr) %>%
  slice_head(n = 10)

#Plots bottom 10 countries by CFR
ggplot(bottom10, aes(x = reorder(country_region, avg_cfr), y = avg_cfr)) +
  geom_col(fill = "#2a52be") +
  coord_flip() +  # horizontal bars for readability
  labs(
    title = "Bottom 10 Countries by Case Fatality Rate",
    x = NULL,
    y = "Avg. Daily CFR") +
  theme_economist() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.text = element_text(size = 10))
```



Analysis

Amongst the top 10 countries, 4 of those countries are either were considered unstable or at war during the period of the pandemic: Yemen, Sudan, Syria, and Somalia. Peru These countries have limited healthcare infrastructure, and and likely did not enforce global public health guidelines consistently. Furthermore, limited testing capacity, underreporting, and challenges in distributing vaccines or enforcing lockdowns may have contributed to higher case fatality rates. Some of the other countries on the list, such as Egypt and

Mexico, have densely crowded urban areas and are tourist hotspots, which potentially could have caused the disease spread more quickly. Generally, the high CFR, could result from an inability to control the spread of the disease or to provide adequate treatment.

Looking at the bottom 10 countries by case fatality rate, half of them are remote island nations in Oceania, including Nauru, Tonga, the Marshall Islands, Palau, and Samoa. It is likely that their geographic isolation played a protective role early in the pandemic. These countries may have experienced fewer initial COVID-19 introductions, especially as global tourism slowed and many potential visitors were under lockdown in their home countries. Additionally some of these countries adhered to strict quarantine policies, and had high vaccination rates.

Our analysis here is mostly speculative, as we do not have the exact data to support any of the insights extracted and it would be very difficult to do so.

5 Limitations & Challenges

Next, we discussn biases, challenges, and limitations encountered throughout this report.

5.1 Bias

I approached this project with a degree of skepticism about the accuracy of reported COVID-19 numbers. I questioned the reliability of data collection methods, especially in developing nations. I wondered whether people had access to hospitals, whether those hospitals were recording cases consistently, and whether that data was being accurately shared with global health organizations. Whilst my concerns were valid, I came to realize they represent only part of the picture. Over time, I shifted my focus toward understanding what the data can reliably tell us, rather than fixating solely on its limitations.

5.2 Other Factors

Limitations

- Lack context - While the data told us about reported cases, it did not provide a lot of context with regards to global and local policies in effect, or the data gathering practices. However this is an common issue when dealing with large-scale global data.
- Lack infrustructure - Countries with limited healthcare infrastructure or centralized health organization, may have underreported cases or deaths due to lack of testing, data systems. Also, these countries might rely on data collected or aggregated by foreign health organizations, which can result in delayed or incomplete reporting.
- Lack of standards - There were no uniform global reporting

Challenges

- I initially planned to include global population data, so that my spatial analysis also include reported cases per capita. I was unable to succesfully merge the global population dataset, and have decided to exclude it from my analysis.

6 Conclusion

This analysis of global case fatality rates (CFR) reveals stark differences in how countries experienced the COVID-19 pandemic. High CFRs were observed in regions grappling with conflict or limited healthcare infrastructure, such as Yemen, Sudan, Syria, Somalia, and Peru, highlighting the compounding impact of instability on public health outcomes. In contrast, countries with greater geographic isolation, such as island nations in Oceania, appeared to benefit from natural and policy-driven barriers that reduced exposure and allowed for more effective containment. A broader geographic pattern also emerged: countries in Eastern Europe, Africa, South and Central America, and parts of Asia often faced higher fatality rates. When viewed alongside GDP per capita data, these trends suggest a possible correlation between national wealth and pandemic response effectiveness. While we cannot establish causation, the similarities between CFR and GDP distributions support the idea that stronger economies with better healthcare systems, earlier interventions, and greater policy enforcement, may have been better equipped to handle the pandemic.

Future Works

- Observe spatial maps based on per capita cases and deaths.
- Observe time series trends on global data.
- Observe spatial and time series patterns in the US datasets.
- Further deep dive the relationship between economic factors and case fatality rate.

7 References

[1] J. Hopkins University, Center for Systems Science and Engineering (CSSE), “CSSE COVID-19 Time Series Data,” GitHub, [Online]. Available: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series. [Accessed: Jun. 22, 2025].

[2] SAP, “What is the difference between Country and Country ISO Code?,” SAP Knowledge Base Article 2518366, [Online]. Available: https://userapps.support.sap.com/sap/support/knowledge/en/2518366. [Accessed: 24-Jun-2025].

[3] International Organization for Standardization, “FM - Micronesia (Federated States of),” ISO Online Browsing Platform, [Online]. Available: https://www.iso.org/obp/ui/#iso:code:3166:FM. [Accessed: 24-Jun-2025].