Global Patterns and Predictive Modeling of COVID-19: A Data-Driven Analysis

Eric F.

2025-04-29

Contents

1 Introduction

This project presents an exploratory and statistical analysis of the global COVID-19 pandemic, based on a public dataset compiled and maintained by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The dataset, accessible via their GitHub repository (https://github.com/CSSEGISandData), contains comprehensive daily records of confirmed COVID-19 cases, deaths, and recoveries across countries worldwide, from the beginning of the pandemic in early 2020 until 2023.

Each record corresponds to a particular country (or subnational region) and date, reporting cumulative counts of infections, deaths, and recoveries, along with basic geographic information such as the country or province name. To enrich the analysis, we also incorporated population data from auxiliary tables also stored within John Hopkins' repository.

The goal of this project is to import, clean, transform, and analyze the COVID-19 dataset using modern tools of data science in R. The exploratory data analysis (EDA) focuses on global trends over time, the different shares of infected and deaths between countries and comparative indicators like infection rates per capita, mortality rates per capita, and case fatality rates (CFR). Predictive statistical models are also developed to explore relationships between these variables, such as the connection between infection incidence and mortality.

In keeping with the principles of reproducibility and transparency, all steps—from data extraction and transformation to modeling and visualization—are fully documented within code chunks. This ensures that the results can be independently verified and easily updated if newer data becomes available.

In addition to presenting the main findings, we critically discuss potential biases in the dataset, including underreporting, differences in testing capacities, inconsistencies in death attribution methodologies and socio-political reporting issues, all biases that must be recognized in order to correctly interpret the trends and conclusions derived from the analysis.

2 Extract, Transform and Load (ETL) data

First, we install the packages and import the necessary libraries:

```r
packages <- c("tidyverse", "lubridate", "dplyr", "naniar",
              "hms", "scales", "patchwork", "forcats",
              "knitr", "kableExtra")

# Install packages
installed <- packages %in% rownames(installed.packages())
if (any(!installed)) {
  install.packages(packages[!installed])
}
```

```
library(tidyverse)
library(lubridate)
library(dplyr)
library(naniar)
library(hms)
library(scales)
library(patchwork)
library(kableExtra)
```

Then, we copy the data url and create the R containers for it. Now, that will let us see the structure of the datasets.

```
url_in <-
 ↪ "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data

file_names <- c("time_series_covid19_confirmed_US.csv",
 ↪ "time_series_covid19_confirmed_global.csv",
 ↪ "time_series_covid19_deaths_US.csv",
 ↪ "time_series_covid19_deaths_global.csv",
 ↪ "time_series_covid19_recovered_global.csv")

urls <- str_c(url_in, file_names)

us_cases <- read_csv(urls[1])
global_cases <- read_csv(urls[2])
us_deaths <- read_csv(urls[3])
global_deaths <- read_csv(urls[4])
global_recovered <- read_csv(urls[5])

#This is the table where we will look for the countries' populations
uid_lookup_url <-
 ↪ "https://raw.githubusercontent.com/efuentesrico/UID-Data/main/UID_ISO_FIPS_LookUp_Ta

uid <- read_csv(uid_lookup_url)

#This step was not necessary for this dataset, but sometimes we have to do
 ↪ this:
# Convert "" to NA so that vis_miss works
#global_cases[global_cases == ""] <- NA
```

```r
# Check the structure of the datasets
#We will just look at global_cases to save space
head(global_cases, 3)
```

```
## # A tibble: 3 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
## # i 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>, ...
```

```r
#head(global_deaths, 3)
#head(global_recovered, 3)
#head(uid, 3)
#head(us_cases, 3)
```

As we can clearly see, all tables have thousands of columns, since every date appears as a single column. Thus, we will use pivot_longer to turn the dates into rows. We also change the data types appropriately and select only the variables relevant to our analysis, leaving behind things like latitude, longitude, county name and so on.

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

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

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

```
us_cases <- us_cases %>%                         pivot_longer(cols =
 ↪  -c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State",
 ↪  "Country_Region", "Lat", "Long_", "Combined_Key"), names_to = "date",
 ↪  values_to = "cases") %>% mutate(date = mdy(date)) %>%  select(-c(Lat,
 ↪  Long_, UID, iso2, iso3, code3, FIPS))

us_deaths <- us_deaths %>%                          pivot_longer(cols =
 ↪  -c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State",
 ↪  "Country_Region", "Lat", "Long_", "Combined_Key", "Population"),
 ↪  names_to = "date", values_to = "cases") %>% mutate(date = mdy(date))
 ↪  %>%  select(-c(Lat, Long_, UID, iso2, iso3, code3, FIPS))
```

Now, with vis_miss(dataset) we can visualize how many data points are missing for each attribute for each table.

```
#This takes some time, so erase the "#" only to check how the datasets
 ↪  looks like up until this part of the code
#vis_miss(global_cases, warn_large_data = FALSE)
#vis_miss(global_deaths, warn_large_data = FALSE)
#vis_miss(global_recovered, warn_large_data = FALSE)
#vis_miss(us_cases, warn_large_data = FALSE)
#vis_miss(us_deaths, warn_large_data = FALSE)
```

Now we erase negatives, since they are probably typos.

```
global_cases <- global_cases %>% filter(cases >= 0)
global_deaths <- global_deaths %>% filter(cases >= 0)
global_recovered <- global_recovered %>% filter(cases >= 0)
us_cases <- us_cases %>% filter(cases >= 0)
us_deaths <- us_deaths %>% filter(cases >= 0)
```

And now we group the data by county (for US data), country and date.

```
global_cases <- global_cases %>%
  group_by(`Province/State`,`Country/Region`, date) %>%
  summarise(cases = sum(cases), .groups = "drop")

global_deaths <- global_deaths %>%
  group_by(`Province/State`, `Country/Region`, date) %>%
  summarise(cases = sum(cases), .groups = "drop")
```

```
global_recovered <- global_recovered %>%
  group_by(`Province/State`, `Country/Region`, date) %>%
  summarise(cases = sum(cases), .groups = "drop")

us_cases <- us_cases %>%
  group_by(Admin2, Province_State, Country_Region, date) %>%
  summarise(cases = sum(cases), .groups = "drop")

us_deaths <- us_deaths %>%
  group_by(Admin2, Province_State, Country_Region, date) %>%
  summarise(cases = sum(cases), .groups = "drop")
```

And we are finally able to join the tables into one containing both the number of infections and the number of deaths.

```
global_data <- global_cases %>%
  rename(total_infected = cases) %>%
  left_join(global_deaths %>% rename(total_deaths = cases),
            by = c("Province/State", "Country/Region", "date")) %>%
  left_join(global_recovered %>% rename(total_recovered = cases),
            by = c("Province/State", "Country/Region", "date")) %>%
             ↪  rename(Country_Region = `Country/Region`, Province_State =
             ↪  `Province/State`)

US_data <- us_cases %>%
  rename(total_infected_US = cases) %>%
  left_join(us_deaths %>% rename(total_deaths_US = cases),
            by = c("Admin2", "Province_State", "Country_Region", "date"))
```

We group again, now by state (for US data) and by country (for global data), since that is the desired level of granularity for this project.

```
global_data_by_country <- global_data %>%
  group_by(Country_Region, date) %>%
  summarise(total_infected = sum(total_infected), total_deaths =
   ↪  sum(total_deaths), total_recovered = sum(total_recovered), .groups =
   ↪  "drop")

US_data_by_state <- US_data %>%
  group_by(Province_State, Country_Region, date) %>%
  summarise(total_infected_US = sum(total_infected_US), total_deaths_US =
   ↪  sum(total_deaths_US), .groups = "drop")
```

The data is now joined with the UID table, in order to visualize the states' and countries'population.

```
#Table for name of countries
uid_country_pop <- uid %>%
  group_by(Country_Region) %>%
  summarise(Population = max(Population, na.rm = TRUE))

#Table for name of states
uid_state_pop <- uid %>%
  group_by(Province_State) %>%
  summarise(Population = max(Population, na.rm = TRUE))

#We filter North Korea since its data makes no sense, for it is the only
 ↪  country that has more deaths than infected. We also filter the Holy
 ↪  See, for it is NOT  a country

#This gives the COVID timeseries for all countries daily
global_data_by_country <- global_data_by_country %>%
 ↪  left_join(uid_country_pop, by = c("Country_Region")) %>%
 ↪  select(Country_Region, date, total_infected, total_deaths,
 ↪  total_recovered, Population) %>% filter(Country_Region != "Korea,
 ↪  North") %>% filter(Country_Region != "Holy See")

#This gives the cumulative COVID data for all countries
global_data_total <- global_data_by_country %>%
         group_by(date) %>%
         summarise(total_infected = sum(total_infected), total_deaths =
          ↪  sum(total_deaths), total_recovered = sum(total_recovered),
          ↪  .groups = "drop")

#Although we cleaned the US data, we will not use it for anything, for we
 ↪  will only focus on global data
#This gives the COVID timeseries for all US states daily
US_data_by_state <- US_data_by_state %>% left_join(uid_state_pop, by =
 ↪  c("Province_State")) %>% select(Province_State, Country_Region, date,
 ↪  total_infected_US, total_deaths_US, Population)

#This gives the cumulative COVID data for all US states
US_data_total <- US_data_by_state %>%
  group_by(Country_Region, date) %>%
  summarise(total_infected_US = sum(total_infected_US), total_deaths_US =
   ↪  sum(total_deaths_US), .groups = "drop")
```

We can also use the lag function, which does the opposite of the cumsum function and will help us keep track of the new infected and the new deaths daily:

```
global_data_total <- global_data_total %>%
  mutate(new_infected = total_infected - lag(total_infected), new_deaths =
   ↪  total_deaths - lag(total_deaths), new_recovered = total_recovered -
   ↪  lag(total_recovered))

global_data_by_country <- global_data_by_country %>%
  mutate(new_infected = total_infected - lag(total_infected), new_deaths =
   ↪  total_deaths - lag(total_deaths), new_recovered = total_recovered -
   ↪  lag(total_recovered))
```

Now we will also add new columns to the table, all corresponding to time or epidemiological measures calculated from already existing data and which will help us analyze COVID-19 in the EDA section, such as new infected, new deaths, infected per million, deaths per million and cafe fatality rate (CFR).

```
#calculate the total global population
total_global_pop <- uid_country_pop %>%
  filter(is.finite(Population)) %>%
  summarise(total_population = sum(Population)) %>%
  pull(total_population)

# We use the lag function, which does the opposite of the cumsum function
 ↪  and will help us keep track of the new infected and the new deaths
 ↪  daily
global_data_total <- global_data_total %>% mutate(new_infected =
 ↪  total_infected - lag(total_infected), new_deaths = total_deaths -
 ↪  lag(total_deaths), new_recovered = total_recovered -
 ↪  lag(total_recovered), infected_per_million = 1000000*
 ↪  total_infected/total_global_pop, deaths_per_million = 1000000*
 ↪  total_deaths/total_global_pop, case_fatality_rate = 100 *
 ↪  total_deaths/total_infected)

# We use the lag function, which does the opposite of the cumsum function
 ↪  and will help us keep track of the new infected and the new deaths
 ↪  daily
global_data_by_country <- global_data_by_country %>% mutate(new_infected =
 ↪  total_infected - lag(total_infected), new_deaths = total_deaths -
 ↪  lag(total_deaths), new_recovered = total_recovered -
 ↪  lag(total_recovered), infected_per_million = 1000000*
 ↪  total_infected/Population, deaths_per_million = 1000000*
 ↪  total_deaths/Population, case_fatality_rate = 100 *
 ↪  total_deaths/total_infected)

# Get the cumulative results for all countries
```

```r
global_data_by_country_totals <- global_data_by_country %>%
 ↪  group_by(Country_Region) %>% summarize(total_deaths =
 ↪  max(total_deaths), total_infected = max(total_infected), Population =
 ↪  max(Population), infected_per_million = 1000000*
 ↪  total_infected/Population, deaths_per_million = 1000000*
 ↪  total_deaths/Population, case_fatality_rate = 100 *
 ↪  total_deaths/total_infected)

# Filter only territories with population >0
global_data_by_country_totals <- global_data_by_country_totals %>%
 ↪  filter(Population>0)
```

Finally, we can see the summary and structure of the datasets:

```r
str(global_data_total)
```

```
## tibble [1,143 x 10] (S3: tbl_df/tbl/data.frame)
##  $ date                : Date[1:1143], format: "2020-01-22" "2020-01-23" ...
##  $ total_infected      : num [1:1143] 557 657 944 1437 2120 ...
##  $ total_deaths        : num [1:1143] 17 18 26 42 56 82 131 133 172 214 ...
##  $ total_recovered     : num [1:1143] NA NA NA NA NA NA NA NA NA NA ...
##  $ new_infected        : num [1:1143] NA 100 287 493 683 ...
##  $ new_deaths          : num [1:1143] NA 1 8 16 14 26 49 2 39 42 ...
##  $ new_recovered       : num [1:1143] NA NA NA NA NA NA NA NA NA NA ...
##  $ infected_per_million: num [1:1143] 0.0719 0.0848 0.1219 0.1855 0.2737 ...
##  $ deaths_per_million  : num [1:1143] 0.0022 0.00232 0.00336 0.00542 0.00723 ...
##  $ case_fatality_rate  : num [1:1143] 3.05 2.74 2.75 2.92 2.64 ...
```

```r
summary(global_data_total)
```

```
##       date            total_infected       total_deaths     total_recovered
##  Min.   :2020-01-22   Min.   :      557   Min.   :     17   Min.   : NA
##  1st Qu.:2020-11-02   1st Qu.: 47426033   1st Qu.:1282415   1st Qu.: NA
##  Median :2021-08-15   Median :207815422   Median :4388700   Median : NA
##  Mean   :2021-08-15   Mean   :277261828   Mean   :3866855   Mean   :NaN
##  3rd Qu.:2022-05-27   3rd Qu.:528830634   3rd Qu.:6312696   3rd Qu.: NA
##  Max.   :2023-03-09   Max.   :676570119   Max.   :6881796   Max.   : NA
##                                                             NA's   :1143
##   new_infected        new_deaths     new_recovered   infected_per_million
##  Min.   :    100   Min.   :    1   Min.   : NA     Min.   :    0.07
##  1st Qu.: 265246   1st Qu.: 2178   1st Qu.: NA     1st Qu.: 6123.62
##  Median : 473972   Median : 5840   Median : NA     Median :26832.98
##  Mean   : 592443   Mean   : 6026   Mean   :NaN    Mean   :35799.85
##  3rd Qu.: 676774   3rd Qu.: 8777   3rd Qu.: NA    3rd Qu.:68282.23
##  Max.   :4083281   Max.   :60903   Max.   : NA    Max.   :87358.25
##  NA's   :1         NA's   :1       NA's   :1143
##  deaths_per_million case_fatality_rate
##  Min.   :  0.0022   Min.   :1.009
##  1st Qu.:165.5845   1st Qu.:1.194
##  Median :566.6658   Median :2.101
##  Mean   :499.2855   Mean   :2.398
##  3rd Qu.:815.0907   3rd Qu.:2.488
##  Max.   :888.5725   Max.   :7.730
##
```

```
#summary(global_data_by_country)
#str(global_data_by_country)
```

3 Exploratory Data Analysis (EDA)

In this section, we conduct a comprehensive exploratory data analysis (EDA) of global COVID-19 data from early 2020 to 2023. Through a series of visualizations and metrics, we aim to uncover temporal patterns, geographic disparities, and mortality trends associated with the pandemic. The analysis is structured to highlight cumulative and daily dynamics, differences across countries, and a deeper case study focused on Colombia. This exploration provides essential contextual understanding before moving on to predictive modeling.

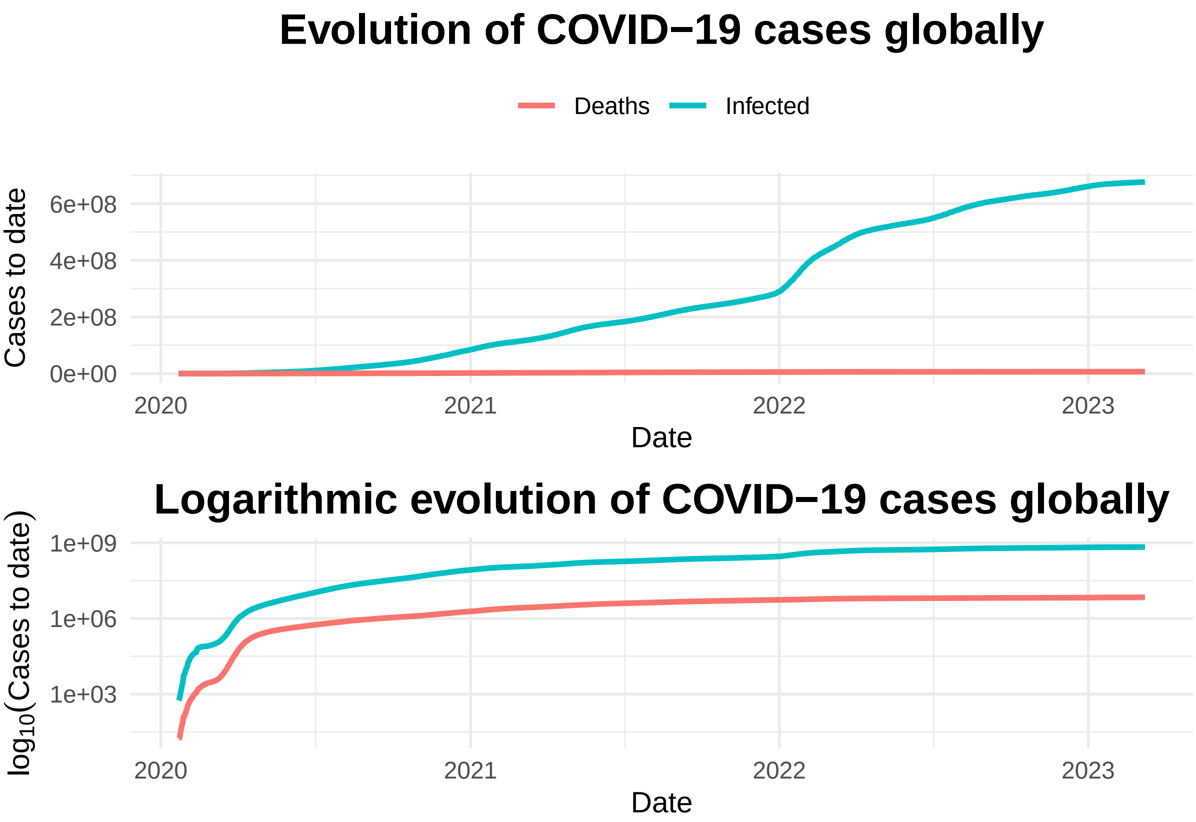
3.1 Global evolution of COVID-19 cases

3.1.1 Cumulative cases over time

The first analysis explores the cumulative number of infections and deaths globally over time. Two versions are provided: one with a linear scale and one with a logarithmic scale (base 10) to better visualize variations during early pandemic stages.

- In the linear plot, a steady increase in cases and deaths is observed, with notable accelerations during major pandemic waves (e.g., late 2021–early 2022).
- The logarithmic plot reveals that while infections and deaths grew rapidly at the beginning, the growth rate gradually decelerated, following an approximately linear pattern in the log scale, characteristic of saturation effects.

```r
# First graph (linear scale)
p1 <- ggplot(global_data_total, aes(x = date)) +
    geom_line(aes(y = total_infected, color = "Infected"),
              size = 1, alpha = 1) +
    geom_line(aes(y = total_deaths, color = "Deaths"),
              size = 1, alpha = 1) +
    labs(
        title = "Evolution of COVID-19 cases globally",
        x = "Date",
        y = "Cases to date"
    ) +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        legend.title = element_blank(),
        legend.position = "top"
    )

# Second graph (log scale)
p2 <- ggplot(global_data_total, aes(x = date)) +
    geom_line(aes(y = total_infected, color = "Infected"),
              size = 1, alpha = 1) +
    geom_line(aes(y = total_deaths, color = "Deaths"),
              size = 1, alpha = 1) +
    labs(
        title = "Logarithmic evolution of COVID-19 cases globally",
        x = "Date",
        y = expression(log[10](Cases~to~date))
    ) +
    scale_y_log10() +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        legend.title = element_blank(),
        legend.position = "top"
    )

# Combine graphs vertically (ncol=1 is 1 column)
p1 + p2 + plot_layout(ncol = 1, guides = "collect") & theme(legend.position
 ↪  = 'top')
```


Evolution of COVID−19 cases globally

Logarithmic evolution of COVID−19 cases globally

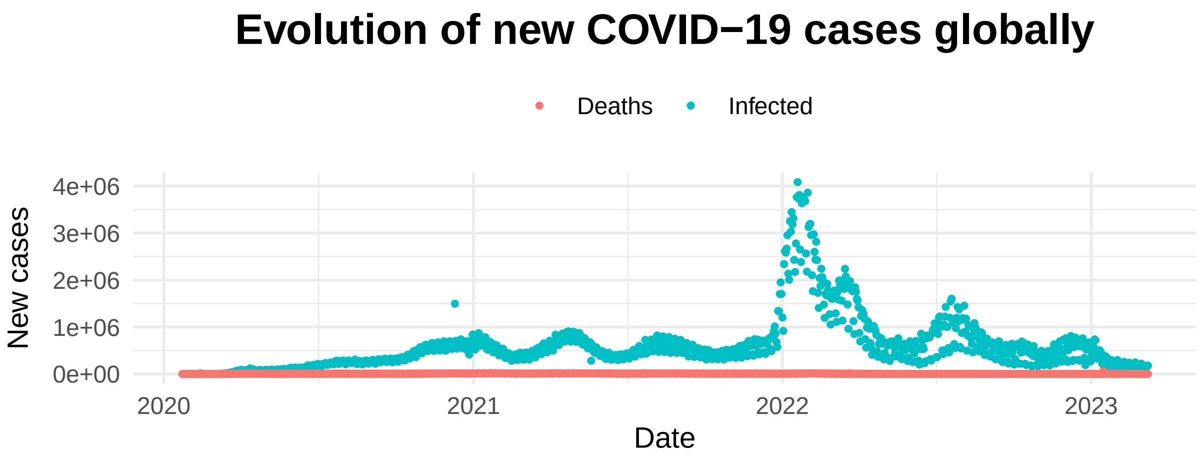
3.1.2 Evolution of new cases over time

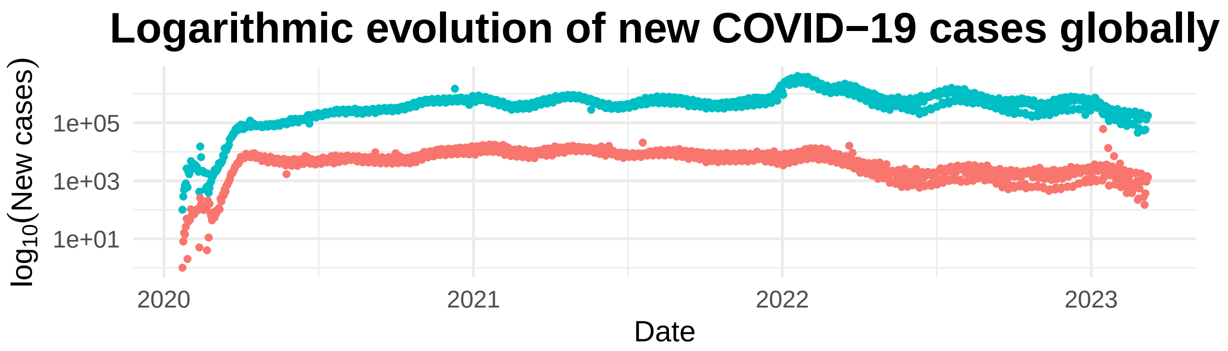
Next, we focus on the daily new cases and deaths:

- In the linear-scale plot, clear pandemic waves are visible, particularly around early 2022, where new infections spiked dramatically.
- The logarithmic-scale plot emphasizes the dynamics of new cases even when their magnitude varies widely, showing that while the baseline level of daily cases remained substantial, mortality rates decreased more steadily compared to infections.

These plots suggest that although COVID-19 waves continued to emerge, mortality control measures may have improved over time.

```r
# First graph
p3 <- ggplot(global_data_total, aes(x = date)) +
  geom_point(aes(y = new_infected, color = "Infected"),
             size = 0.75, alpha = 1) +
  geom_point(aes(y = new_deaths, color = "Deaths"),
             size = 0.75, alpha = 1) +
  labs(
    title = "Evolution of new COVID-19 cases globally",
    x = "Date",
    y = "New cases"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    legend.title = element_blank(),
    legend.position = "top"
  )

# Second graph
p4 <- ggplot(global_data_total, aes(x = date)) +
  geom_point(aes(y = new_infected, color = "Infected"),
             size = 0.75, alpha = 1) +
  geom_point(aes(y = new_deaths, color = "Deaths"),
             size = 0.75, alpha = 1) +
  labs(
    title = "Logarithmic evolution of new COVID-19 cases globally",
    x = "Date",
    y = expression(log[10](New~cases))
  ) +
  scale_y_log10() +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    legend.title = element_blank(),
    legend.position = "top"
  )

# Combine
p3 + p4 + plot_layout(ncol = 1, guides = "collect") & theme(legend.position
 ↪  = 'top')
```





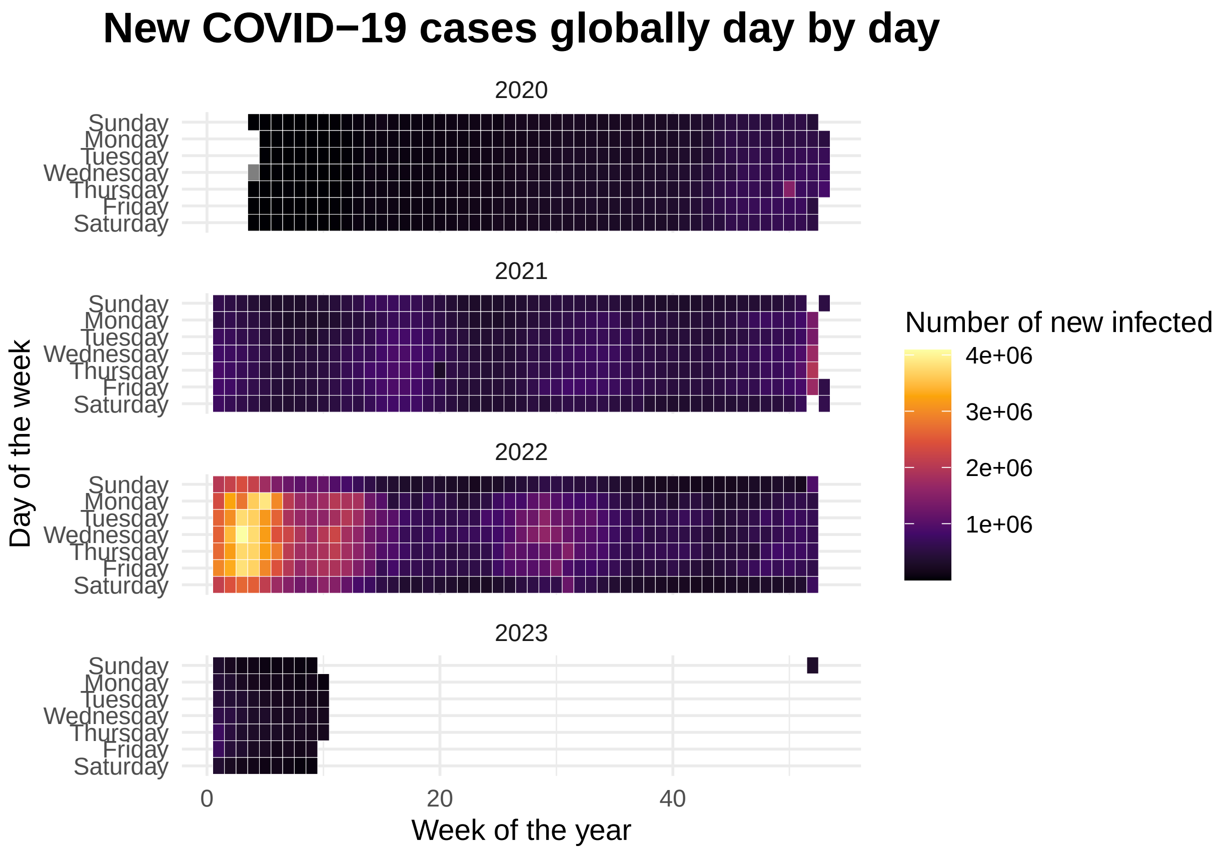
3.1.3 Heatmap: New cases by day of the week

A heatmap was constructed to visualize new infections day-by-day across the weeks and years:

- The most intense spread occurred during the first half of 2022, evidenced by the strong color intensities.
- Some periodic patterns are visible, where new case reporting tended to decrease during weekends (likely due to reporting lags).

The heatmap also visually captures how the pandemic's activity gradually diminished through 2023.

```r
invisible(Sys.setlocale("LC_TIME", "C"))  # Print dates in english

#This extracts weekday, month, year and week from the date data
global_data_total <- global_data_total %>%
  mutate(
    weekday = weekdays(date),
    month = month(date, label = TRUE),
    year = year(date),
    week = isoweek(date)
  )

#Heatmap
global_data_total %>%
  mutate(weekday = wday(date, label = TRUE, abbr = FALSE),
         week = isoweek(date),
         year = year(date)) %>%
  group_by(year, week, weekday) %>%
  summarise(new_cases = mean(new_infected, na.rm = TRUE)) %>%
  ggplot(aes(x = week, y = fct_rev(weekday), fill = new_cases)) +
  geom_tile(color = "white") +
  scale_fill_viridis_c(option = "inferno") +
  labs(title = "New COVID-19 cases globally day by day",
       x = "Week of the year",
       y = "Day of the week",
       fill = "Number of new infected") +
  facet_wrap(~year, ncol = 1) +
  theme_minimal() +
    theme(
       plot.title = element_text(hjust = 0.5, face = "bold", size = 16))
```



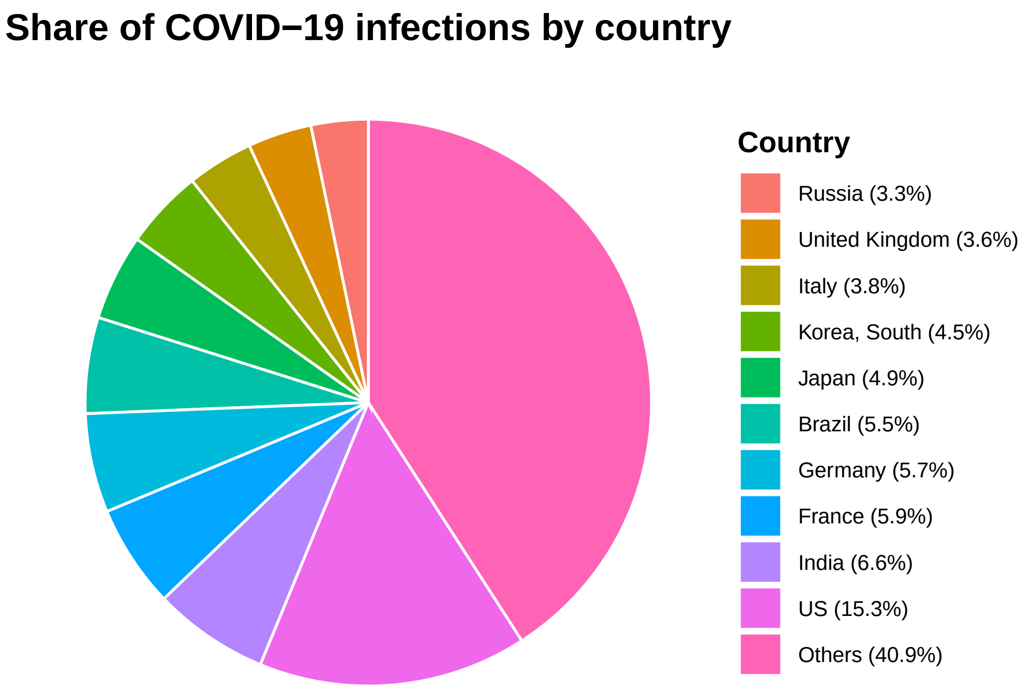
3.2 Geographical distribution of COVID-19 outcomes

3.2.1 Share of global infections by country

A pie chart summarizes the distribution of total infections among countries:

- The United States led with 15.3% of global infections.
- India, France, Germany, and Brazil also contributed significant shares.
- Collectively, the top 10 countries accounted for nearly 59% of all global infections, highlighting a strong geographical concentration.

```
countries_most_infected <- global_data_by_country_totals %>%
 ↪  slice_max(total_infected, n = 10)

global_data_by_country_totals %>%
  mutate(country_group = ifelse(Country_Region %in%
   ↪  countries_most_infected$Country_Region,
                                Country_Region, "Others")) %>%
  group_by(country_group) %>%
  summarise(total_infected = sum(total_infected, na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(
    percentage = total_infected / sum(total_infected) * 100,
    label = paste0(country_group, " (", round(percentage, 1), "%)"),
    label = fct_reorder(label, total_infected)  # <-- esta línea ordena de
     ↪  menor a mayor
  ) %>%
  ggplot(aes(x = "", y = total_infected, fill = label)) +
  geom_col(width = 1, color = "white") +
  coord_polar(theta = "y") +
  labs(
    title = "Share of COVID-19 infections by country",
    fill = "Country"
  ) +
  theme_void() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    legend.title = element_text(face = "bold"),
    legend.position = "right",
    legend.text = element_text(size = 8),
    legend.box = "horizontal",
    legend.justification = "center",
    plot.margin = margin(20, 20, 20, 20))
```

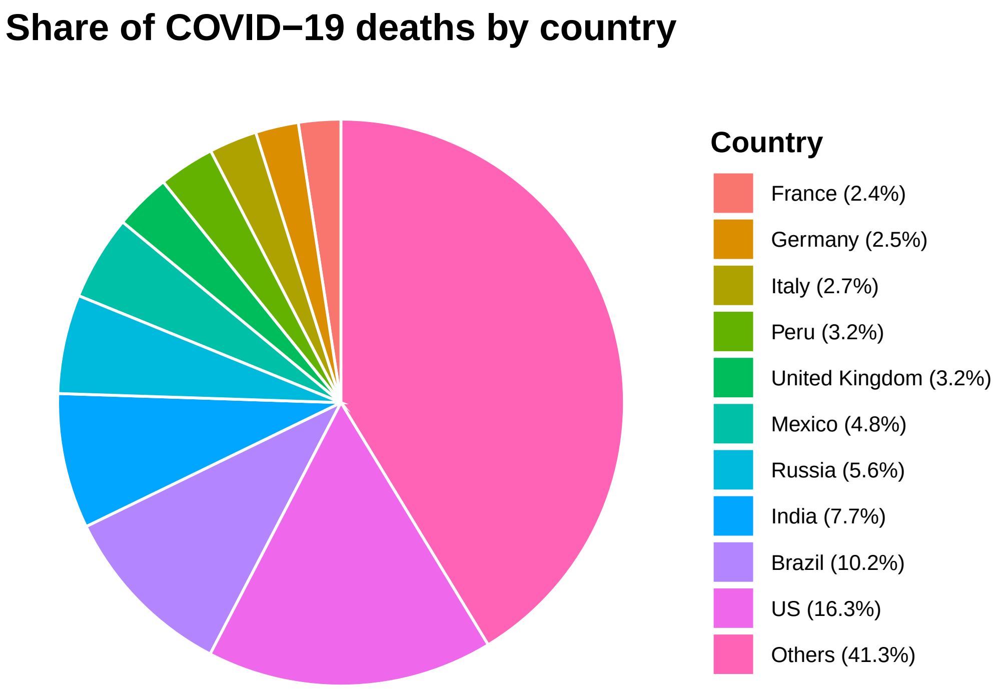


3.2.2 Share of global deaths by country

Similarly, the distribution of total deaths was visualized:

- The United States again recorded the highest share (16.3%), followed by Brazil (10.2%) and India (7.7%).
- The distribution of deaths is slightly different from infections, suggesting variations in mortality rates across countries.

```
countries_most_deaths <- global_data_by_country_totals %>%
 ↪  slice_max(total_deaths, n = 10)

global_data_by_country_totals %>%
  mutate(country_group = ifelse(Country_Region %in%
   ↪  countries_most_deaths$Country_Region,
                                Country_Region, "Others")) %>%
  group_by(country_group) %>%
  summarise(total_deaths = sum(total_deaths, na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(
    percentage = total_deaths / sum(total_deaths) * 100,
    label = paste0(country_group, " (", round(percentage, 1), "%)"),
    label = fct_reorder(label, total_deaths)  # <-- esta línea ordena de
     ↪  menor a mayor
  ) %>%
  ggplot(aes(x = "", y = total_deaths, fill = label)) +
  geom_col(width = 1, color = "white") +
  coord_polar(theta = "y") +
  labs(
    title = "Share of COVID-19 deaths by country",
    fill = "Country"
  ) +
  theme_void() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    legend.title = element_text(face = "bold"),
    legend.position = "right",
    legend.text = element_text(size = 8),
    legend.box = "horizontal",
    legend.justification = "center",
    plot.margin = margin(20, 20, 20, 20))
```



3.3 Spread of the virus: infection rate per million

The spread of the virus was estimated using the country's infected per million,

$$\text{Infected per million} = \frac{\text{Total Infected}}{\text{Population}} \times 1000000,$$

which gives a population-independent measure of how much the COVID-19 spread within the country.

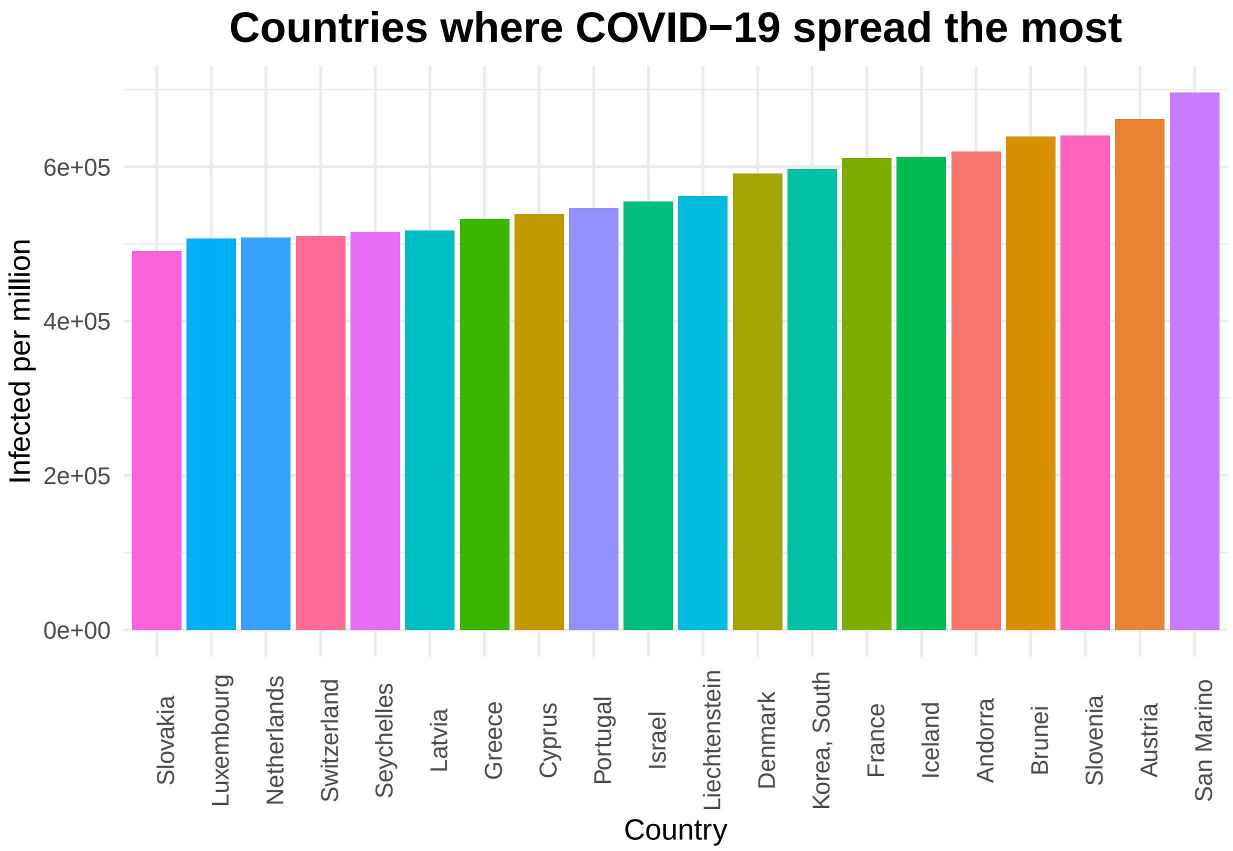
3.3.1 Countries with highest infection rates

-Countries like San Marino, Austria, and Slovenia recorded the highest numbers of infections relative to their population.

-Several small European countries dominate this list, possibly due to higher testing rates or dense urban populations facilitating faster virus spread.

```
most_infected_countries <- global_data_by_country_totals %>%
 ↪  slice_max(infected_per_million, n = 20)
```

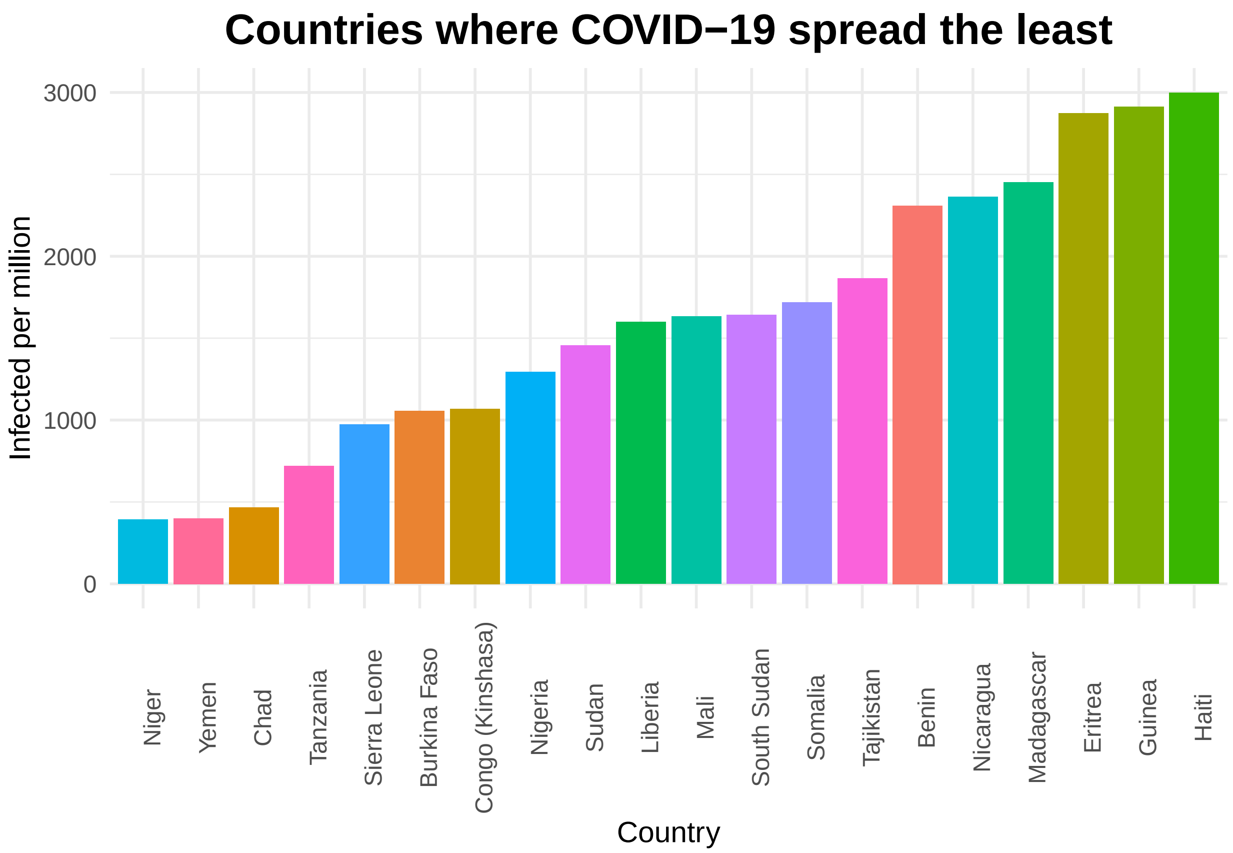
```
ggplot(most_infected_countries, aes(x = fct_reorder(Country_Region,
 ↪  infected_per_million), y = infected_per_million, fill = Country_Region))
 ↪  +
    geom_bar(stat = "identity", show.legend = FALSE) +
    labs(title = "Countries where COVID-19 spread the most",
         x = "Country",
         y = "Infected per million") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         ↪  legend.position ="top",                                   axis.text.x =
         ↪  element_text(angle = 90))
```



3.3.2 Countries with lowest infection rates

- Nations such as Haiti, Guinea, and Eritrea exhibited very low infection rates.
- These lower rates could be due to limited testing, underreporting, demographic characteristics (younger populations), or stricter containment measures.

```
least_infected_countries <- global_data_by_country_totals %>%
 ↪ slice_min(infected_per_million, n = 20)

ggplot(least_infected_countries, aes(x = fct_reorder(Country_Region,
 ↪ infected_per_million), y = infected_per_million, fill = Country_Region))
 ↪ +
    geom_bar(stat = "identity", show.legend = FALSE) +
    labs(title = "Countries where COVID-19 spread the least",
         x = "Country",
         y = "Infected per million") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         ↪ legend.position ="top",                            axis.text.x =
         ↪ element_text(angle = 90))
```



3.4 Mortality: death rate per million

The mortality of the virus was estimated using the country’s deaths per million,

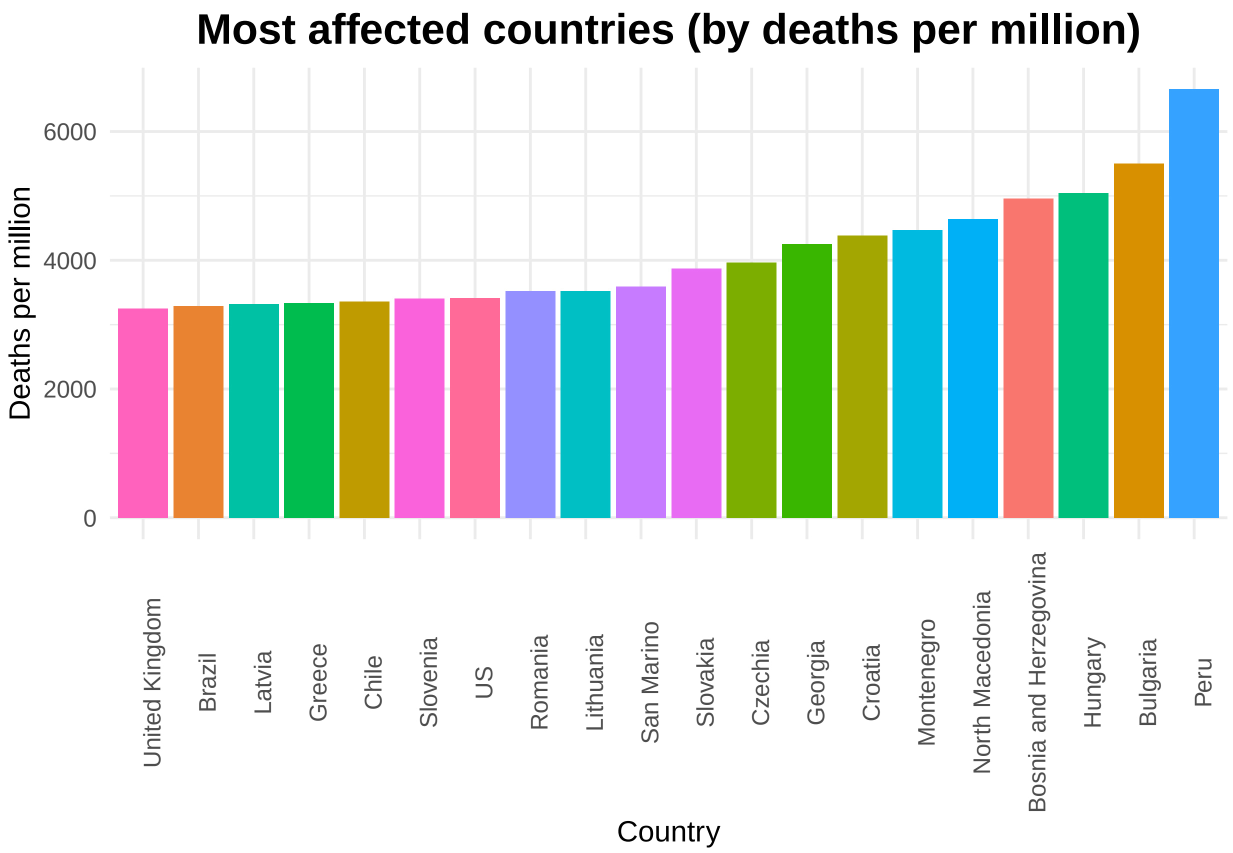
$$\text{Deaths per million} = \frac{\text{Total Deaths}}{\text{Population}} \times 1000000,$$

which gives a population-independent measure of how much the country's population was affected and killed by COVID-19.

3.4.1 Countries with highest death rates

Most affected countries (e.g., Peru, Bulgaria, Bosnia and Herzegovina) exhibited extremely high mortality rates per capita, exceeding 6,000 deaths per million inhabitants.

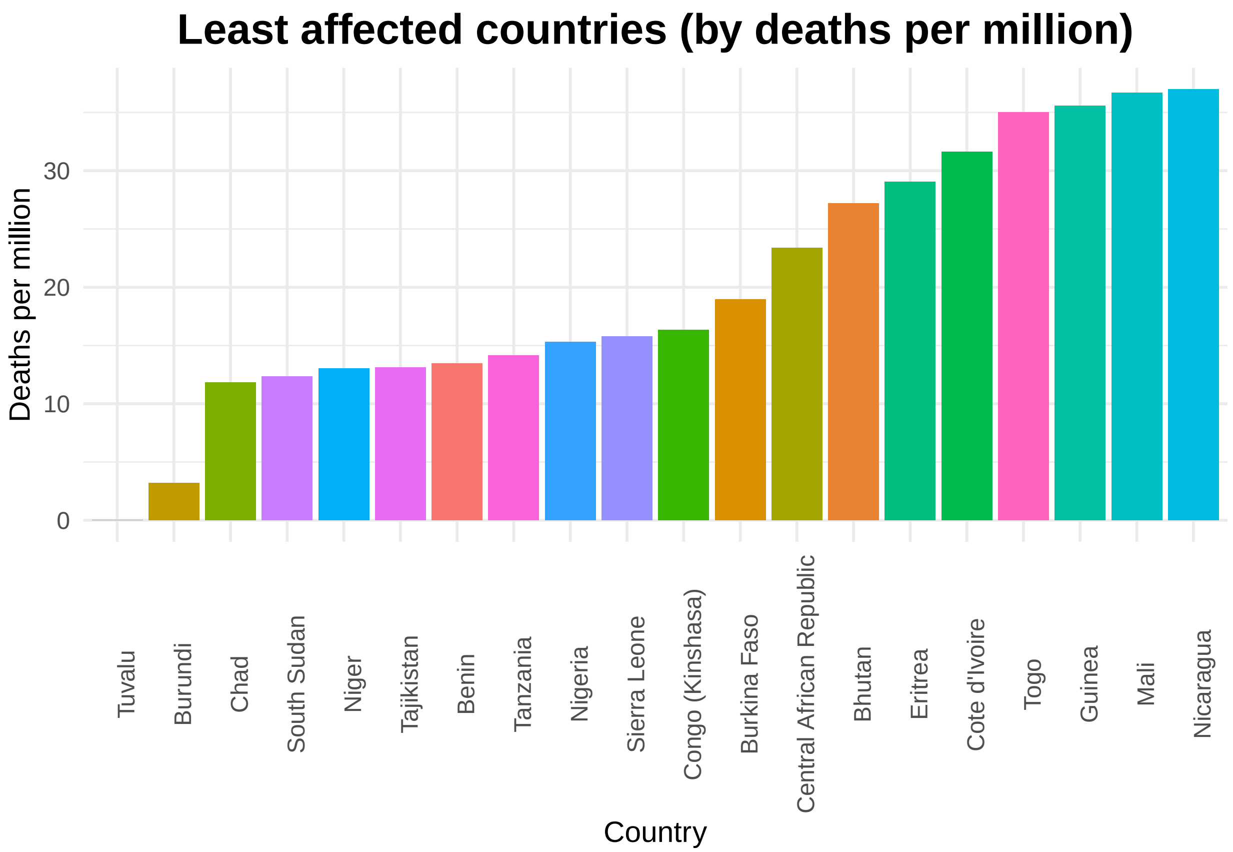
```
worst_countries <- global_data_by_country_totals %>%
 ↪  slice_max(deaths_per_million, n = 20)

ggplot(worst_countries, aes(x = fct_reorder(Country_Region,
 ↪  deaths_per_million), y = deaths_per_million, fill = Country_Region)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    labs(title = "Most affected countries (by deaths per million)",
         x = "Country",
         y = "Deaths per million") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         ↪  legend.position ="top",                              axis.text.x =
         ↪  element_text(angle = 90))
```



3.4.2 Countries with lowest death rates

Least affected countries (e.g., Tuvalu, Burundi, Chad) showed remarkably low mortality rates, often fewer than 30 deaths per million.

```
best_countries <- global_data_by_country_totals %>% filter(Population>0,
 ↪  total_infected>0) %>% slice_min(deaths_per_million, n = 20)

ggplot(best_countries, aes(x = fct_reorder(Country_Region,
 ↪  deaths_per_million), y = deaths_per_million, fill = Country_Region)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    labs(title = "Least affected countries (by deaths per million)",
         x = "Country",
         y = "Deaths per million") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         ↪  legend.position ="top",                            axis.text.x =
         ↪  element_text(angle = 90))
```



This analysis highlights strong inequalities in the pandemic's human cost across countries.

3.5 Medical system effectiveness: Case Fatality Rate (CFR)

The Case Fatality Rate (CFR) was calculated as:

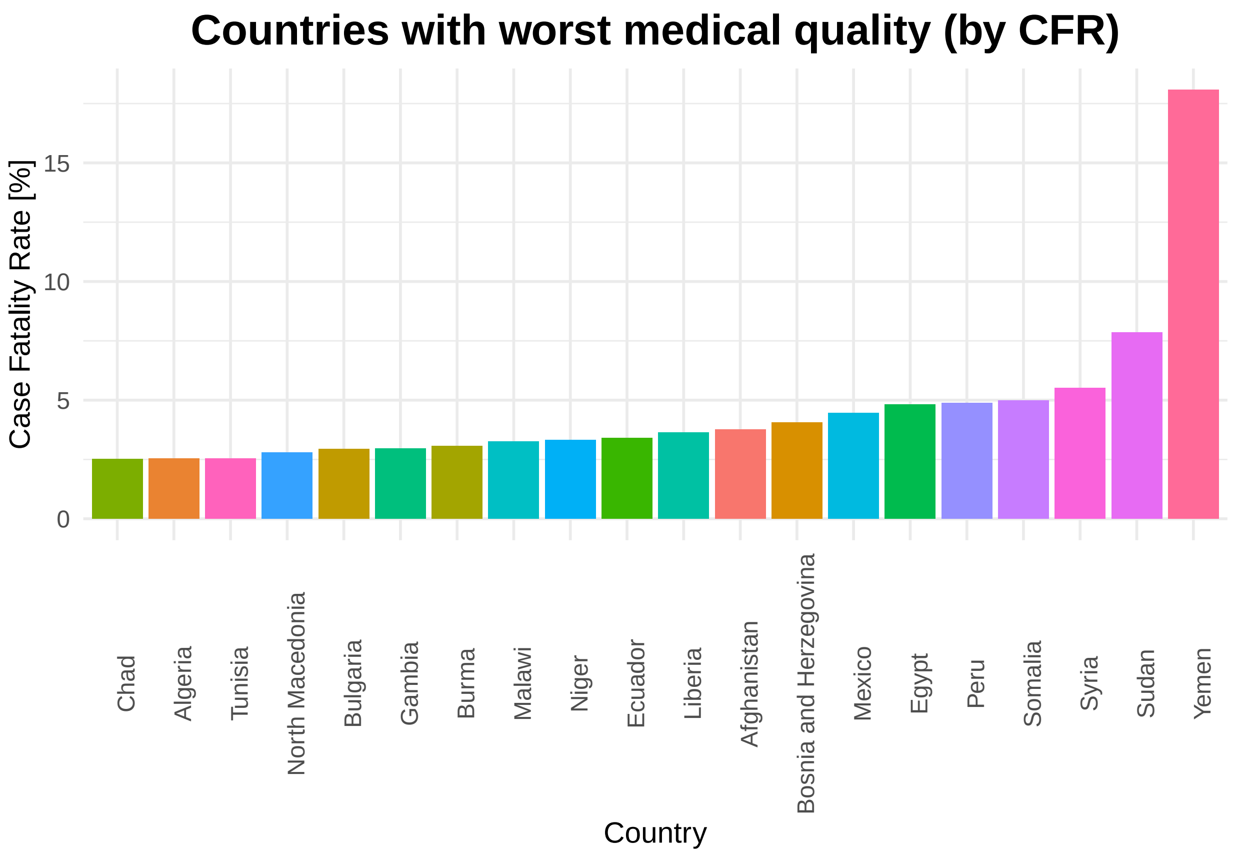
$$\text{CFR} = \frac{\text{Total Deaths}}{\text{Total Infected}} \times 100$$

This metric helps evaluate how well countries managed to prevent deaths among infected individuals.

3.5.1 Countries with worst CFR (suggesting poorer medical quality)

- Yemen, Sudan, and Syria reported the highest CFRs, exceeding 5% in several cases.
- High CFRs generally reflect factors such as weaker healthcare infrastructure, delayed medical attention, limited access to treatments, or data underreporting (especially of mild cases).

```
worst_cfr_countries <- global_data_by_country_totals %>%
 ↪  slice_max(case_fatality_rate, n = 20)

ggplot(worst_cfr_countries, aes(x = fct_reorder(Country_Region,
 ↪  case_fatality_rate), y = case_fatality_rate, fill = Country_Region)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    labs(title = "Countries with worst medical quality (by CFR)",
         x = "Country",
         y = "Case Fatality Rate [%]") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         ↪  legend.position ="top",                          axis.text.x =
         ↪  element_text(angle = 90))
```

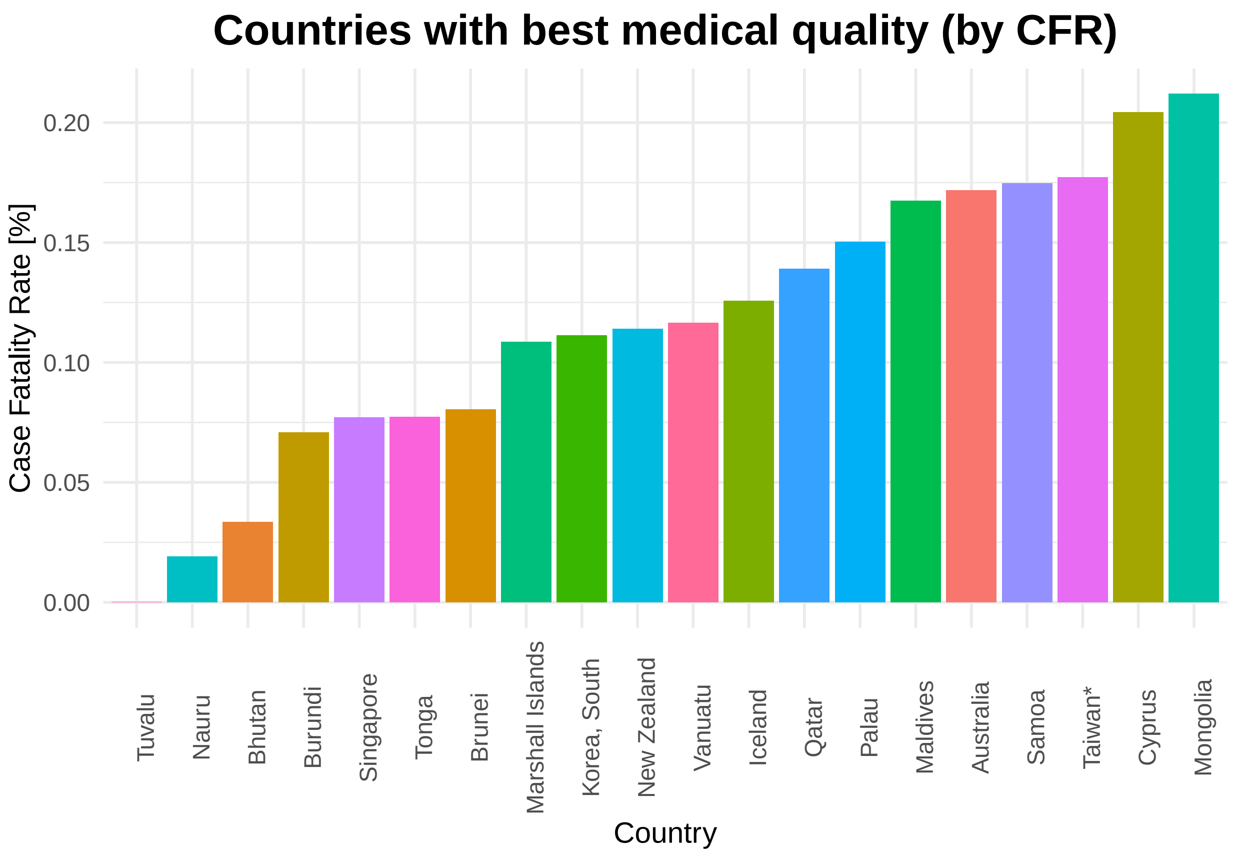


3.5.2 Countries with best CFR (suggesting better medical quality)

- Countries like Tuvalu, Nauru, and Singapore exhibited exceptionally low CFRs, often under 0.1%.

- This suggests excellent healthcare responses, efficient vaccination campaigns, and/or robust early intervention systems.

```
best_cfr_countries <- global_data_by_country_totals %>%
 ↪  slice_min(case_fatality_rate, n = 20)

ggplot(best_cfr_countries, aes(x = fct_reorder(Country_Region,
 ↪  case_fatality_rate), y = case_fatality_rate, fill = Country_Region)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    labs(title = "Countries with best medical quality (by CFR)",
         x = "Country",
         y = "Case Fatality Rate [%]") +
    theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         ↪  legend.position ="top",                         axis.text.x =
         ↪  element_text(angle = 90))
```



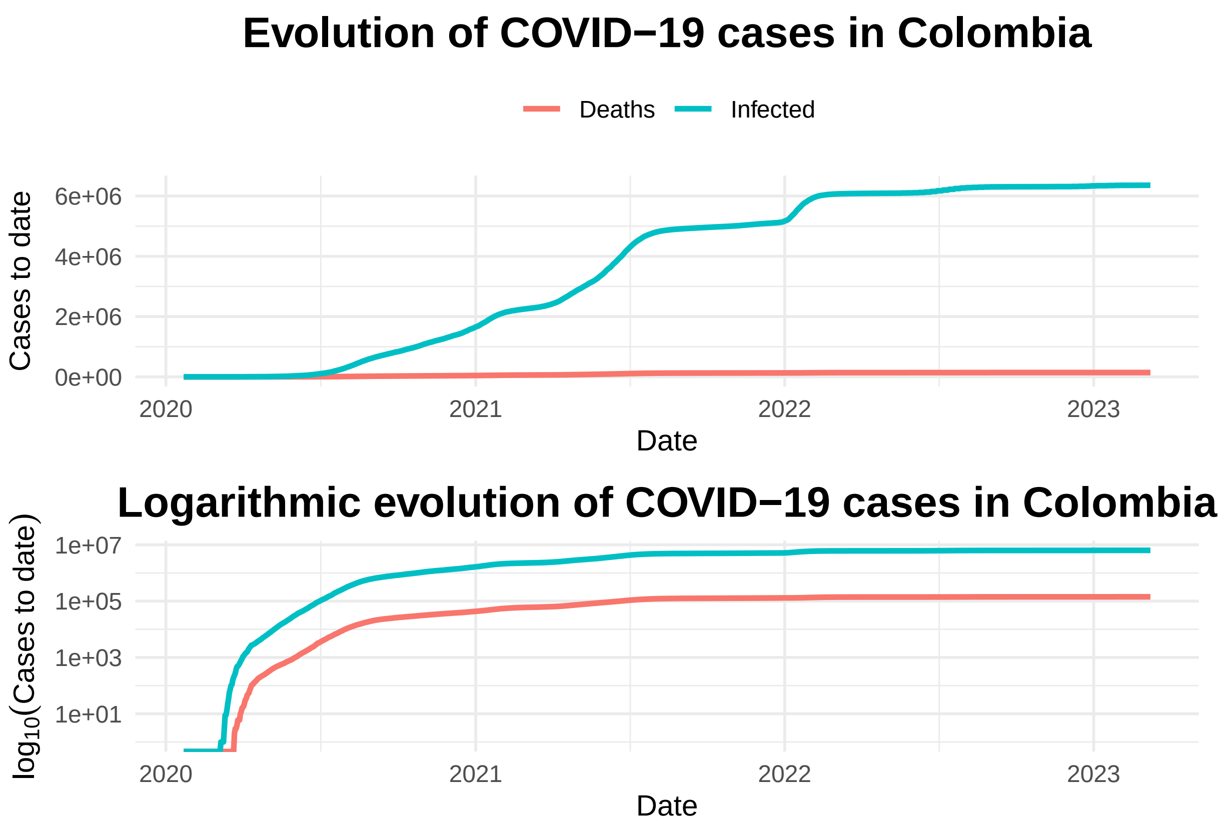
3.6 Case Study: Colombia

A more detailed focus was placed on Colombia's pandemic evolution, since it is the author's country of origin.

3.6.1 Cumulative evolution

- The total number of infections and deaths steadily increased, with major inflection points visible during early 2021 and early 2022, corresponding to global waves.
- Logarithmic visualization confirms the slowing growth rate after successive waves.

```r
# Filter data for Colombia
colombia_data_totals <- global_data_by_country %>%
  filter(Country_Region == "Colombia",
         total_infected >= 0,
         total_deaths >= 0,
         total_recovered >= 0)

# First graph
p5 <- ggplot(colombia_data_totals, aes(x = date)) +
  geom_line(aes(y = total_deaths, color = "Deaths"),
            size = 1, alpha = 1) +
  geom_line(aes(y = total_infected, color = "Infected"),
            size = 1, alpha = 1) +
  labs(
    title = "Evolution of COVID-19 cases in Colombia",
    x = "Date",
    y = "Cases to date"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    legend.title = element_blank(),
    legend.position = "top"
  )

# Second graph
p6 <- ggplot(colombia_data_totals, aes(x = date)) +
  geom_line(aes(y = total_deaths, color = "Deaths"),
            size = 1, alpha = 1) +
  geom_line(aes(y = total_infected, color = "Infected"),
            size = 1, alpha = 1) +
  labs(
    title = "Logarithmic evolution of COVID-19 cases in Colombia",
    x = "Date",
    y = expression(log[10](Cases~to~date))
  ) +
  scale_y_log10() +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    legend.title = element_blank(),
    legend.position = "top"
  )

# Combine
p5 + p6 + plot_layout(ncol = 1, guides = "collect") & theme(legend.position
↪   = 'top')
```

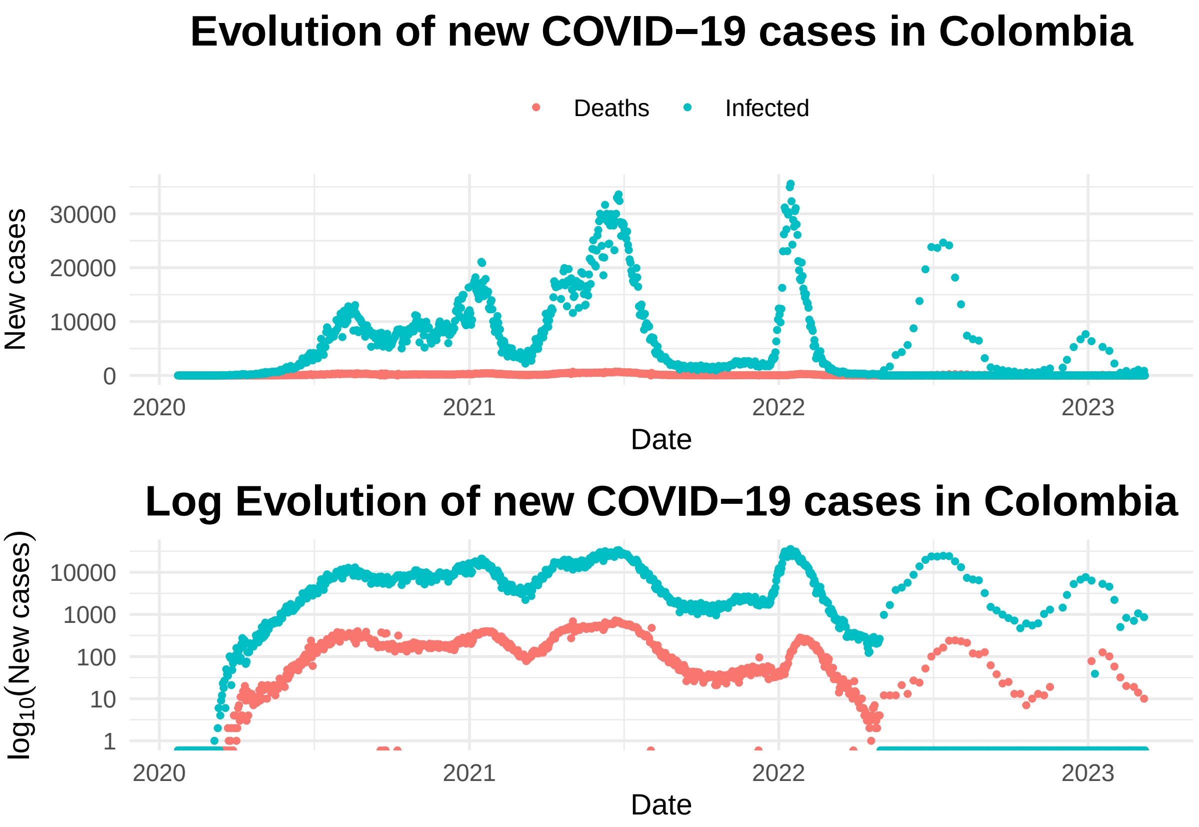


3.6.2 Evolution of new cases and deaths

- Significant peaks in new infections were observed in 2021 and 2022, reflecting national pandemic waves.

- Notably, new deaths showed much smaller peaks relative to infections after mid-2021, suggesting improvements in clinical management, vaccination, or both.

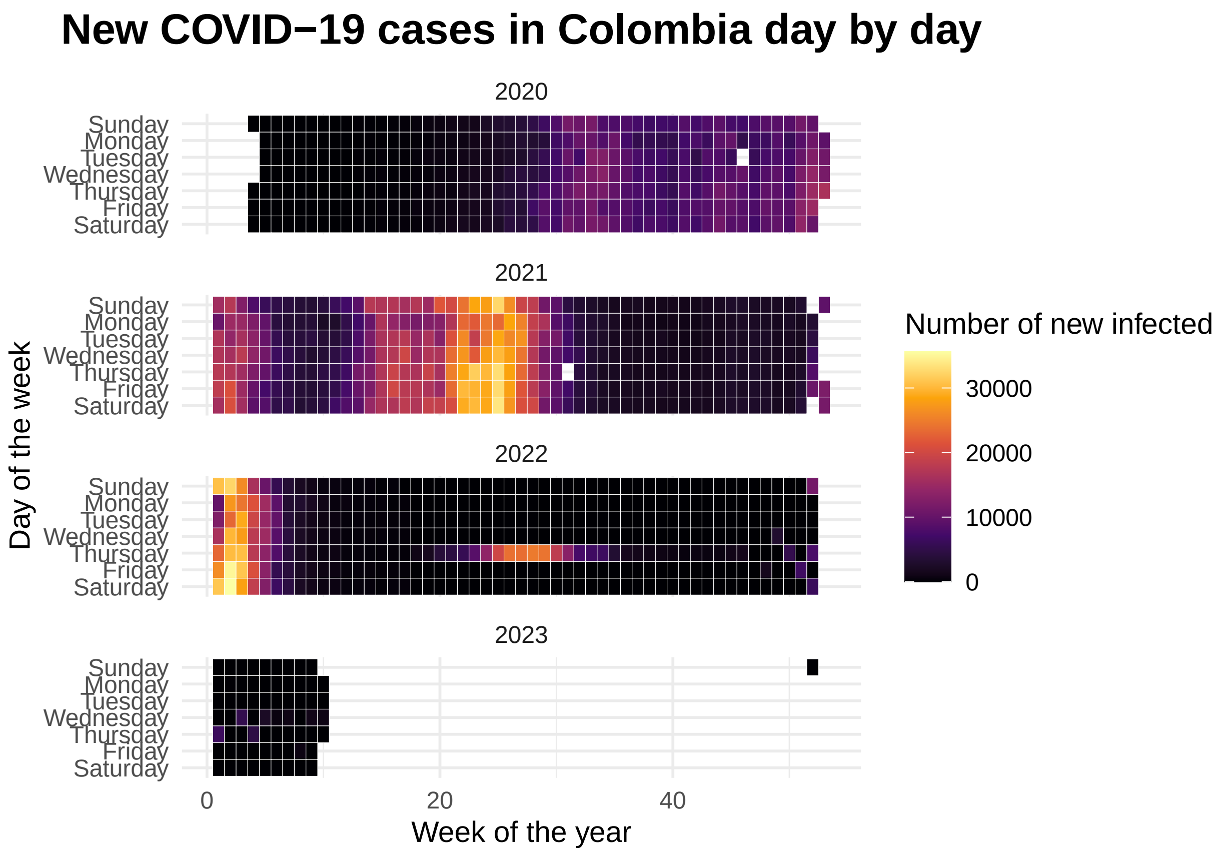
```r
# Filter
colombia_data_totals <- global_data_by_country %>%
  filter(Country_Region == "Colombia",
         new_infected >= 0,
         new_deaths >= 0,
         new_recovered >= 0)

# First graph
p7 <- ggplot(colombia_data_totals, aes(x = date)) +
  geom_point(aes(y = new_deaths, color = "Deaths"),
             size = 0.75, alpha = 1) +
  geom_point(aes(y = new_infected, color = "Infected"),
             size = 0.75, alpha = 1) +
  labs(
    title = "Evolution of new COVID-19 cases in Colombia",
    x = "Date",
    y = "New cases"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    legend.title = element_blank(),
    legend.position = "top"
  )

#Second graph
p8 <- ggplot(colombia_data_totals, aes(x = date)) +
  geom_point(aes(y = new_deaths, color = "Deaths"),
             size = 0.75, alpha = 1) +
  geom_point(aes(y = new_infected, color = "Infected"),
             size = 0.75, alpha = 1) +
  labs(
    title = "Log Evolution of new COVID-19 cases in Colombia",
    x = "Date",
    y = expression(log[10](New~cases))
  ) +
  scale_y_log10() +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    legend.title = element_blank(),
    legend.position = "top"
  )

# Combine
p7 + p8 + plot_layout(ncol = 1, guides = "collect") & theme(legend.position
↪   = 'top')
```



3.6.3 Heatmap: new cases day by day

- Colombia's heatmap shows the heaviest transmission during mid-2021, with evident periodicities and reporting dips over weekends.
- By 2023, COVID-19 activity had substantially diminished in Colombia, following a global trend.

```
invisible(Sys.setlocale("LC_TIME", "C"))  # Impose english to dates

colombia_data_totals <- colombia_data_totals %>%
  mutate(
    weekday = weekdays(date),
    month = month(date, label = TRUE),
    year = year(date),
    week = isoweek(date)
  )

colombia_data_totals %>%
  mutate(weekday = wday(date, label = TRUE, abbr = FALSE),
         week = isoweek(date),
         year = year(date)) %>%
  group_by(year, week, weekday) %>%
  summarise(new_cases = mean(new_infected, na.rm = TRUE)) %>%
  ggplot(aes(x = week, y = fct_rev(weekday), fill = new_cases)) +
  geom_tile(color = "white") +
  scale_fill_viridis_c(option = "inferno") +
  labs(title = "New COVID-19 cases in Colombia day by day",
       x = "Week of the year",
       y = "Day of the week",
       fill = "Number of new infected") +
  facet_wrap(~year, ncol = 1) +
  theme_minimal() +
    theme(
        plot.title = element_text(hjust = 0.5, face = "bold", size = 16))
```



3.6.4 Summary statistics for Colombia

The following tables summarize key COVID-19 statistics for Colombia as of the final dataset update:

```
# Filter for Colombia
colombia_summary <- global_data_by_country_totals %>%
  filter(Country_Region == "Colombia") %>%
  select(Country_Region, Population, total_infected, total_deaths,
         infected_per_million, deaths_per_million, case_fatality_rate)

# Adjust labels of tables
colombia_summary_formatted <- colombia_summary %>%
  rename(
    Country = Country_Region,
    Population = Population,
    `Total Infected` = total_infected,
    `Total Deaths` = total_deaths,
    `Infected per Million` = infected_per_million,
    `Deaths per Million` = deaths_per_million,
```

```r
    `Case Fatality Rate (%)` = case_fatality_rate
  )

# Table 1: General data
general_data <- colombia_summary_formatted %>%
  select(Country, Population, `Total Infected`, `Total Deaths`)

knitr::kable(
  general_data,
  caption = "General COVID-19 Statistics for Colombia",
  digits = 2,
  align = "c",
  booktabs = TRUE
) %>%
  kableExtra::kable_styling(
    latex_options = c("striped", "hold_position", "HRule"),
    full_width = FALSE,
    position = "center",
    stripe_color = "gray!15"
  )
```

Table 1: General COVID-19 Statistics for Colombia

Country	Population	Total Infected	Total Deaths
Colombia	50882884	6359093	142339

```r
# Table 2: Infection, mortality and CFR rates
rates_data <- colombia_summary_formatted %>%
  select(`Infected per Million`, `Deaths per Million`, `Case Fatality Rate
   ↪   (%)`)

knitr::kable(
  rates_data,
  caption = "COVID-19 Infection and Mortality Rates for Colombia",
  digits = 2,
  align = "c",
  booktabs = TRUE
) %>%
  kableExtra::kable_styling(
    latex_options = c("striped", "hold_position", "HRule"),
    full_width = FALSE,
    position = "center",
    stripe_color = "gray!15"
  )
```

Table 2: COVID-19 Infection and Mortality Rates for Colombia

Infected per Million	Deaths per Million	Case Fatality Rate (%)
124975.1	2797.38	2.24

Compared to global averages, Colombia reported a higher-than-expected mortality rate and higher-than-expected case fatality rate based on its infection incidence. While the country experienced a relatively widespread transmission (with ~12.5% of the population officially infected), its death rate suggests greater strain on healthcare capacity or other socio-demographic vulnerabilities that impacted outcomes. Despite extensive infection spread, the elevated CFR points to challenges in clinical management, healthcare accessibility, or delayed interventions during peak transmission periods.

Overall, Colombia's pandemic management faced notable mortality challenges relative to infection volume, underscoring the need for improvements in emergency healthcare response, early detection, and treatment capacities for future pandemics.

4 Predictive or statistical model

4.1 Relationship between incidence and mortality rates

In order to better understand the relationship between the spread of COVID-19 and its associated mortality at the country level, a simple linear regression model was fitted. In this first approach, the dependent variable is the number of deaths per million inhabitants and the independent variable is the number of infections per million inhabitants.

The fitted model provides a first-order approximation of how mortality scales with incidence across countries, assuming a linear relationship. After training the model, the predicted values were appended to the dataset for visualization purposes. The resulting plot displays both the observed data points and the regression line representing the model's prediction.

As expected, a positive relationship is observed: countries with higher infection rates tend to also have higher mortality rates. However, a significant dispersion around the fitted line suggests that other factors—such as healthcare quality, population demographics, public health interventions and vaccination coverage strongly modulate the ultimate mortality outcomes beyond mere infection counts.

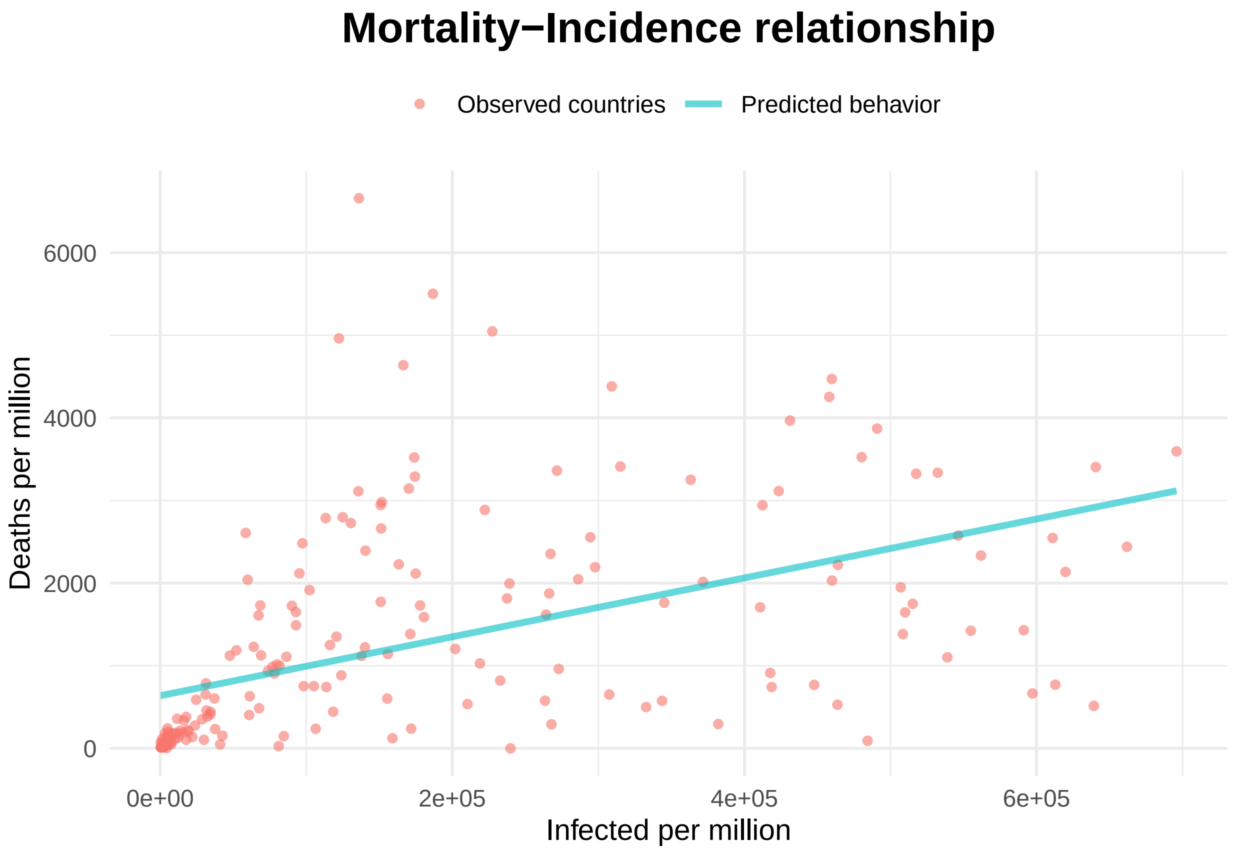
```
#create the linear model
mod <- lm(deaths_per_million ~ infected_per_million, data =
 ↪  global_data_by_country_totals)
```

```r
#append preds to the table
global_data_by_country_totals <- global_data_by_country_totals %>%
 ↪  mutate(pred_deaths_per_million = predict(mod))

#show summary of fitted model
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_million ~ infected_per_million, data = global_data_by_country
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2401.7  -615.8  -391.5   472.4  5535.7
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.375e+02  1.117e+02   5.708 4.28e-08 ***
## infected_per_million 3.564e-03  4.382e-04   8.135 5.04e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1149 on 192 degrees of freedom
## Multiple R-squared:  0.2563, Adjusted R-squared:  0.2524
## F-statistic: 66.17 on 1 and 192 DF,  p-value: 5.038e-14
```

```r
#Plot model
ggplot(global_data_by_country_totals, aes(x = infected_per_million)) +
     geom_point(aes(y = deaths_per_million, color = "Observed countries"),
                size = 1.2, alpha = 0.6) +
     geom_line(aes(y = pred_deaths_per_million, color = "Predicted
      ↪  behavior"),
                size = 1.2, alpha = 0.6) +
     labs(
         title = "Mortality-Incidence relationship",
         x = "Infected per million",
         y = "Deaths per million"
     ) +
     theme_minimal() +
     theme(
         plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         legend.title = element_blank(),   #hide legend title
```

```
        legend.position = "top"
    )
```



4.2 Relationship between incidence and case fatality rates

A second regression model was developed to explore a complementary perspective: the relationship between the case fatality rate (CFR) and the infection rate per capita. In this case, the dependent variable is the percentage of confirmed cases that resulted in death (CFR), while the independent variable remains the number of infections per million inhabitants.

This model investigates whether countries with higher rates of infection also exhibited changes in lethality (perhaps due to healthcare system strain or selection biases in case detection). The model's predictions were again appended to the dataset and visualized alongside the observed data.

Interestingly, the fitted regression reveals a slightly negative relationship: countries with a greater cumulative infection burden tended to have a lower observed case fatality rate. Several plausible interpretations exist for this pattern, including:

- Broader testing in highly affected countries (leading to detection of milder cases).
- Learning effects and improved treatments over time in more exposed regions.

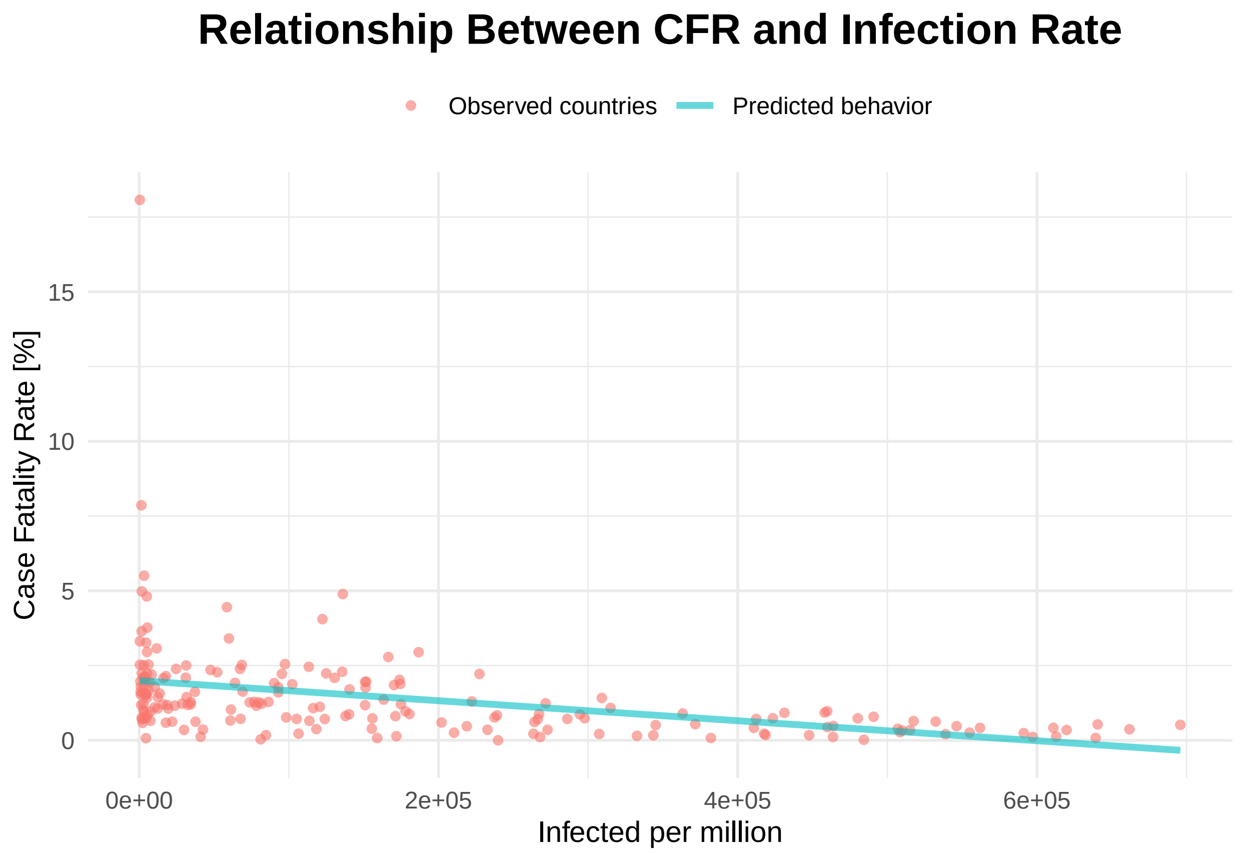
- Potential underreporting biases in countries with lower infection rates.

Nonetheless, the relatively weak slope and high dispersion of points emphasize the complexity of pandemic dynamics and the caution needed when drawing causal conclusions.

```r
global_data_by_country_totals <- global_data_by_country_totals %>%
 ↪  filter(Population>0)

#create the linear model
mod_2 <- lm(case_fatality_rate ~ infected_per_million, data =
 ↪  global_data_by_country_totals)

#append preds to the table
global_data_by_country_totals <- global_data_by_country_totals %>%
 ↪  mutate(pred_cfr = predict(mod_2))

#show summary of fitted model
summary(mod_2)
```

```
##
## Call:
## lm(formula = case_fatality_rate ~ infected_per_million, data = global_data_by_country
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9079 -0.7265 -0.2012  0.3779 16.0820
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.994e+00  1.491e-01  13.369  < 2e-16 ***
## infected_per_million -3.347e-06  5.851e-07  -5.721 4.02e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.535 on 192 degrees of freedom
## Multiple R-squared:  0.1456, Adjusted R-squared:  0.1412
## F-statistic: 32.72 on 1 and 192 DF,  p-value: 4.021e-08
```

```r
#Plot model
ggplot(global_data_by_country_totals, aes(x = infected_per_million)) +
     geom_point(aes(y = case_fatality_rate, color = "Observed countries"),
                size = 1.2, alpha = 0.6) +
     geom_line(aes(y = pred_cfr, color = "Predicted behavior"),
                size = 1.2, alpha = 0.6) +
     labs(
         title = "Relationship Between CFR and Infection Rate",
         x = "Infected per million",
         y = "Case Fatality Rate [%]"
     ) +
     theme_minimal() +
     theme(
         plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
         legend.title = element_blank(),   #hide legend title
         legend.position = "top"
     )
```



4.3 Comparison and Purpose

Together, the two linear models offer complementary views into the pandemic's impact:

The Mortality–Incidence model highlights how raw death counts scaled with infections across countries, serving as a first approximation of overall pandemic severity.

The CFR–Incidence model focuses on healthcare performance and outcome severity conditional on infection, offering indirect insights into medical system efficacy, testing policies, and population vulnerability.

While both models are simple and do not account for confounding factors, they serve as valuable exploratory tools to identify broad global patterns. Future work could extend these models by incorporating additional predictors (e.g., median age, healthcare expenditures, vaccination rates) to refine the understanding of the key drivers behind international COVID-19 mortality differences.

5 Conclusions

This analysis provided a comprehensive overview of the global evolution of COVID-19 cases, combining cumulative and daily perspectives, per-country evaluations, and basic predictive models. By analyzing both total infections and total deaths per million inhabitants, the study offered a balanced view of the pandemic's spread and lethality.

The Exploratory Data Analysis (EDA) revealed that COVID-19 exhibited highly heterogeneous patterns across countries and time periods. Some nations, such as Peru and Bulgaria, experienced extremely high death rates relative to their populations, while others, such as Tuvalu and Burundi, recorded minimal mortality impacts. Similarly, countries such as San Marino and Austria exhibited the highest incidence rates, whereas nations like Niger and Haiti had some of the lowest.

Furthermore, the Case Fatality Rate (CFR) analysis suggested a strong link between a country's medical capacity and its ability to manage the pandemic. Countries with lower CFRs, such as Singapore and Iceland, generally have stronger healthcare infrastructures, while countries with the highest CFRs, like Yemen and Sudan, often face systemic medical vulnerabilities.

The predictive models developed in Section 4, while simple, provided key insights:

- The linear relationship between infections and deaths confirmed that, as expected, more widespread infection generally led to higher mortality, though with significant variability.
- The inverse relationship between infection rate and CFR suggested that countries with widespread testing and infection detection tended to report lower fatality rates, possibly due to early diagnosis and better treatment availability.

In summary, this study highlighted not only the epidemiological dynamics of COVID-19, but also the crucial role played by healthcare quality, governance, and socio-economic factors in shaping pandemic outcomes globally.

6 Bias discussion

While the present study provides important insights, several biases and limitations must be acknowledged:

- **Data accuracy and underreporting:** COVID-19 data collection varied widely across countries. Underreporting of infections, deaths, or both is likely, especially in low-resource settings or in the early stages of the pandemic. This affects the reliability of infection rates, mortality rates, and CFRs.

- **Differences in testing rates:** Countries that implemented mass testing programs may appear to have higher infection rates but lower CFRs. Conversely, countries with limited testing likely missed asymptomatic or mild cases, artificially inflating their CFR.

- **Population structure differences:** Variations in age distribution, urbanization, comorbidities, and socio-economic factors across countries can confound comparisons of mortality and infection rates. Older populations, for instance, tend to experience higher fatality rates independently of healthcare quality.

- **Temporal shifts:** The pandemic evolved through distinct phases, including emergence, multiple waves, and vaccination rollout. Aggregating data across multiple years may obscure important temporal dynamics.

- **Simplification in predictive modeling:** The linear regression models assume linear relationships and do not account for complex interactions between healthcare interventions, social behavior, and virus variants. More sophisticated models (e.g., multivariate regressions, machine learning) would provide richer predictive insights.

- **Definition of Metrics:** CFR is sensitive to how cases and deaths are defined and reported. Some countries may include probable COVID-19 deaths; others may not. Similarly, deaths “with COVID-19” versus “due to COVID-19” can alter mortality statistics.

Despite these biases, the findings provide a valuable approximation of global COVID-19 dynamics, helping to identify broad trends and national differences that merit further study.

7 Recommendations for future studies

Building upon the findings and limitations discussed, several recommendations can be made to enhance future analyses of pandemic-related data:

- **Incorporate socioeconomic and health system variables:** Future models should integrate variables such as hospital bed density, healthcare expenditure per capita, median age, GDP per capita, vaccination rates, and pre-existing burden of chronic diseases. Including these covariates would allow for multivariate modeling, providing a more nuanced understanding of mortality and infection dynamics.

- **Adjust for testing rates and reporting quality:** Developing correction factors based on testing coverage and data transparency indices would help mitigate biases caused by underreporting and variable diagnostic capacity across countries.

- **Time series analysis:** Instead of analyzing only cumulative totals, future studies should apply time series models (e.g., ARIMA, Prophet, or LSTM models) to capture the evolution of infection and death rates across different pandemic waves, identifying inflection points, seasonal effects, and long-term trends.

- **Variant-specific analysis:** Considering the major impact of viral variants (e.g., Delta, Omicron) on transmission and lethality, future studies should aim to disaggregate data by predominant variant periods, allowing for a clearer understanding of biological and epidemiological shifts.

- **Refined metrics:** Beyond the Case Fatality Rate (CFR) and infection rates per million, other epidemiological metrics such as the Infection Fatality Rate (IFR), hospitalization rate, excess mortality, and basic reproduction number (R_0) should be estimated when possible, providing more robust indicators of pandemic severity and health system response.

- **Geographical and regional granularity:** Future analyses could move beyond country-level data to incorporate state, province, or city-level information, capturing within-country heterogeneity, which often exceeds between-country differences.

- **Use of machine learning techniques:** Machine learning models, such as random forests, gradient boosting machines, or neural networks, could be applied to predict mortality or infection surges based on multi-factorial inputs, possibly uncovering non-linear relationships missed by traditional regressions.