# Covid-19 Analysis

Umair Shaikh

2023-12-11

## Introduction

This report presents an analysis of data obtained from the COVID-19 Data Repository, managed by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The dataset is updated daily and includes global figures on confirmed COVID-19 cases, fatalities, and recovery rates. My exploratory data analysis encompasses data cleaning, visualization, and key metric calculation, primarily focusing on countries with the highest number of cases. The visualizations illustrate trends in confirmed cases, deaths, and recoveries. The primary objectives of this analysis are to discern the spread and impact of COVID-19 in severely affected nations and to detect patterns that could inform public health responses and policy decisions.

Furthermore, this study extends beyond exploratory analysis to construct a time series model predicting future COVID-19 case numbers. Time series analysis is crucial for understanding and forecasting the evolution of infectious diseases, thereby enabling more effective planning and response strategies. This model, which is based on historical data of confirmed cases, aims to identify trends affecting case numbers. Such predictive analysis is instrumental in offering insights into the pandemic's future course, thereby assisting government bodies and health officials in making knowledgeable choices regarding interventions, resource allocation, and overall public health measures to curtail the virus's spread and reduce its global impact.

### Load Packages

```
library(tidyverse)
library(readr)
library(ggplot2)
library(scales)
library(prophet)
library(dplyr)
library(tidyr)
library(zoo)
```

### Load Data

The data on laboratory-confirmed COVID-19 cases and reported fatalities were sourced from the COVID-19 Data Repository, managed by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

```
url_folder = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_
file_names = c("time_series_covid19_confirmed_US.csv",  "time_series_covid19_confirmed_global.csv", "ti
urls = str_c(url_folder, file_names)
us_cases = read_csv(urls[1])
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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_cases = 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.
```

```
us_deaths = read_csv(urls[3])
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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[4])
```

```
## 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_recovered = read_csv(urls[5])
```

```
## Rows: 274 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.
```

```r
preprocess_data <- function(data, category) {
  # Identifying columns that are dates
  date_columns <- grep("^[0-9]{1,2}/[0-9]{1,2}/[0-9]{2}$", names(data), value = TRUE)

  # Pivoting the data to a long format
  data_long <- data %>%
    select(-c(Lat, Long), all_of(date_columns)) %>%
    pivot_longer(cols = all_of(date_columns),
                 names_to = "Date",
                 values_to = "Value") %>%
    mutate(Date = as.Date(Date, format = "%m/%d/%y"),
           Category = category,
           `Province/State` = replace_na(`Province/State`, ""),
           Country = ifelse(`Province/State` != "", paste(`Country/Region`, `Province/State`, sep = " -
    group_by(Country, Date, Category) %>%
    summarize(Value = sum(Value), .groups = "drop")
  return(data_long)
}


confirmed <- preprocess_data(global_cases, "Confirmed")
deaths <- preprocess_data(global_deaths, "Deaths")
recovered <- preprocess_data(global_recovered, "Recovered")

covid_data <- bind_rows(confirmed, deaths, recovered)

# Group and summarize covid_data
covid_summary <- covid_data %>%
  group_by(Category, Country, Date) %>%
  summarize(Value = sum(Value), .groups = "drop")


# Set the number of top countries to display
top_n <- 10

# Calculate total cases for each country
total_cases <- covid_data %>%
  filter(Category == "Confirmed") %>%
  group_by(Country) %>%
  summarize(Total_Confirmed = max(Value), .groups = 'drop') %>%
  arrange(desc(Total_Confirmed)) %>%
  slice_head(n = top_n)

# Calculate total deaths for top N countries
total_deaths <- covid_data %>%
  filter(Category == "Deaths", Country %in% total_cases$Country) %>%
  group_by(Country) %>%
  summarize(Total_Deaths = max(Value), .groups = 'drop')

# Calculate total recoveries for top N countries
total_recoveries <- covid_data %>%
  filter(Category == "Recovered", Country %in% total_cases$Country) %>%
  group_by(Country) %>%
```

```
  summarize(Total_Recovered = max(Value), .groups = 'drop')
```

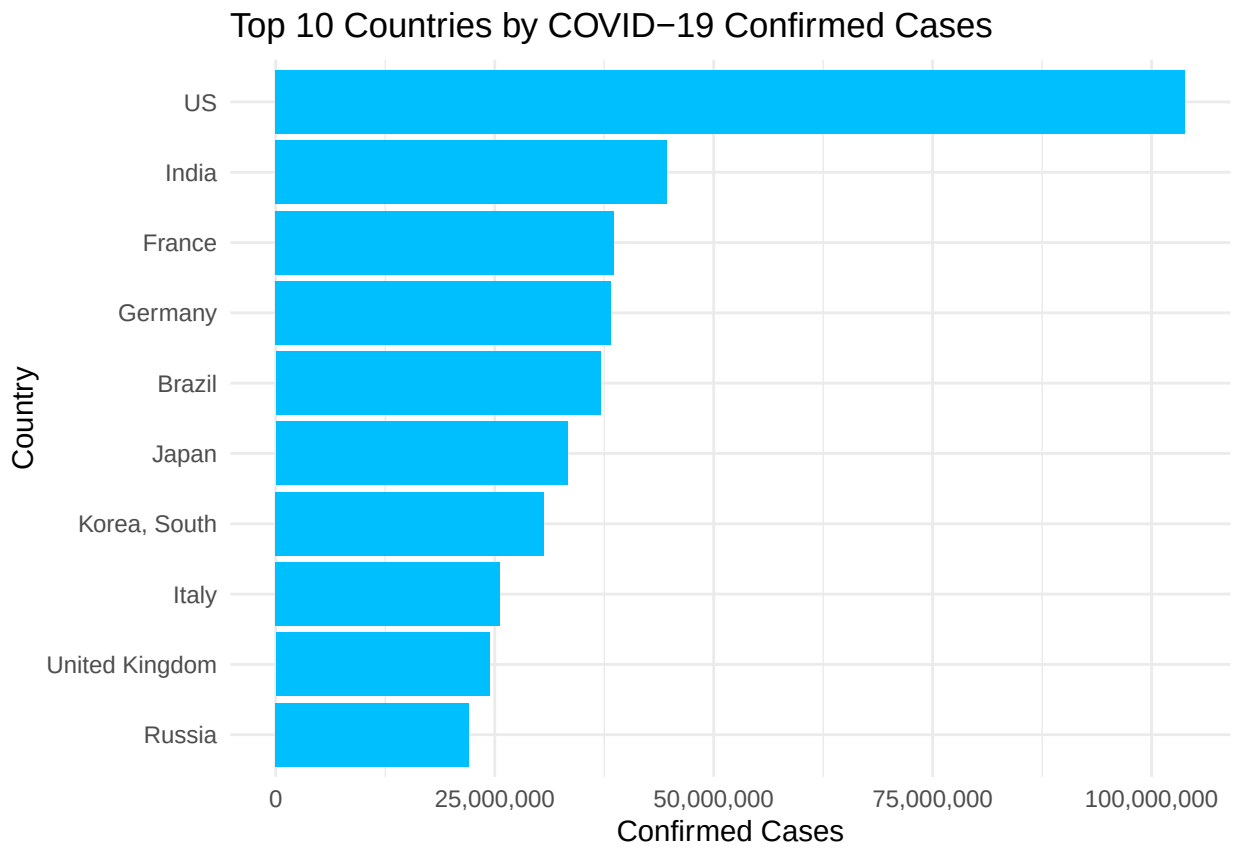
## Data Analysis

A comprehensive analysis and visualization of COVID-19 data, focusing on confirmed cases, deaths, and recoveries. Initially, it loads and processes the data, involving records from 289 regions or countries over approximately 1,144 days. The processing step organizes this vast dataset into a clear, concise format. Once the data is prepped, the code employs the ggplot2 package to create informative bar charts. These charts display the total counts of confirmed cases, deaths, and recoveries in the top 10 most heavily impacted countries, offering a snapshot of the pandemic's severity in these areas.

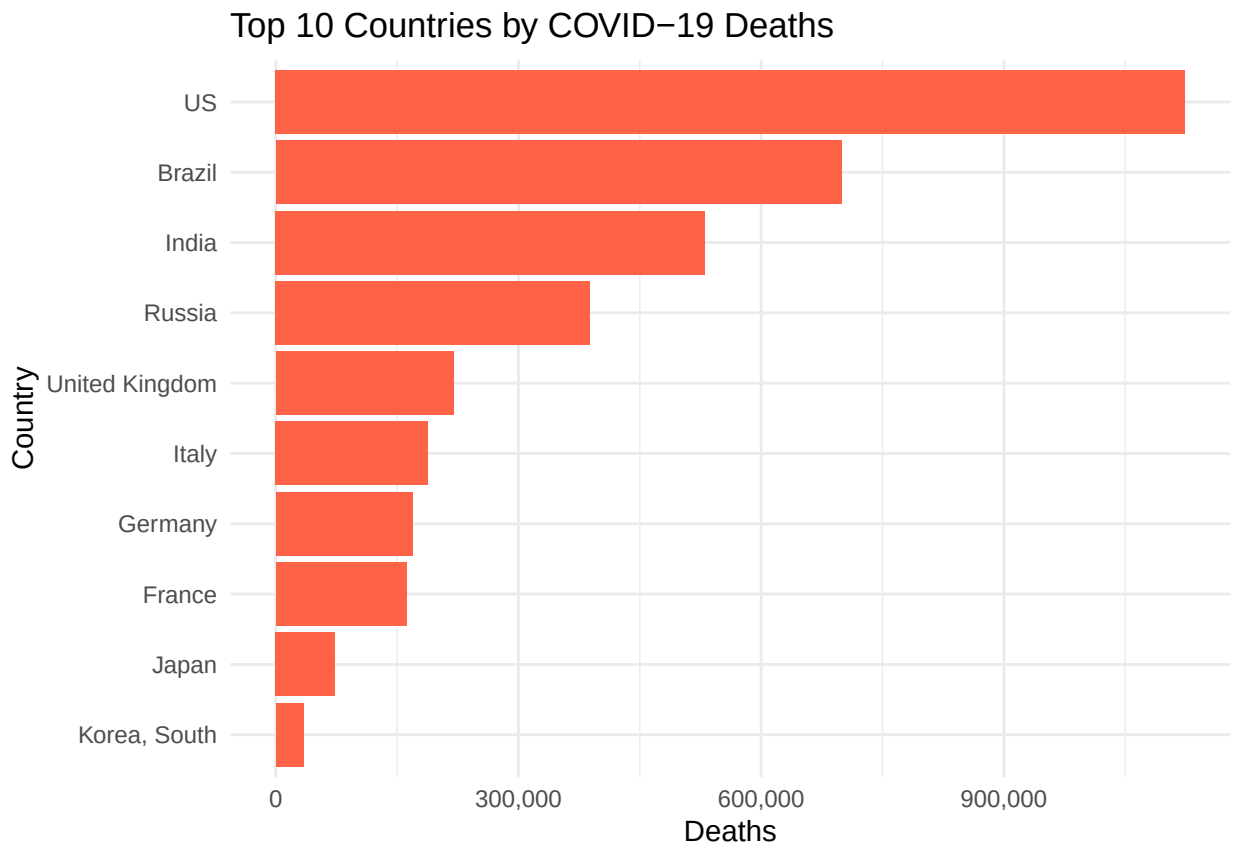
Additionally, it includes a specialized function, plot_top_n_countries, designed to generate line plots that track the pandemic's progression over time in these countries. By applying this function, it creates revealing line plots for the top 10 countries across each category—confirmed cases, deaths, and recoveries. These visualizations are particularly useful in showing how the situation has evolved and fluctuated over time in different parts of the world.

Overall, this is an excellent starting point for further exploratory data analysis on COVID-19. It not only helps in obtaining a clear picture of the pandemic's current state but also showcases various techniques for visualizing time series data. This approach is effective in exploring and presenting different aspects of the pandemic's impact, making complex data more accessible and understandable.

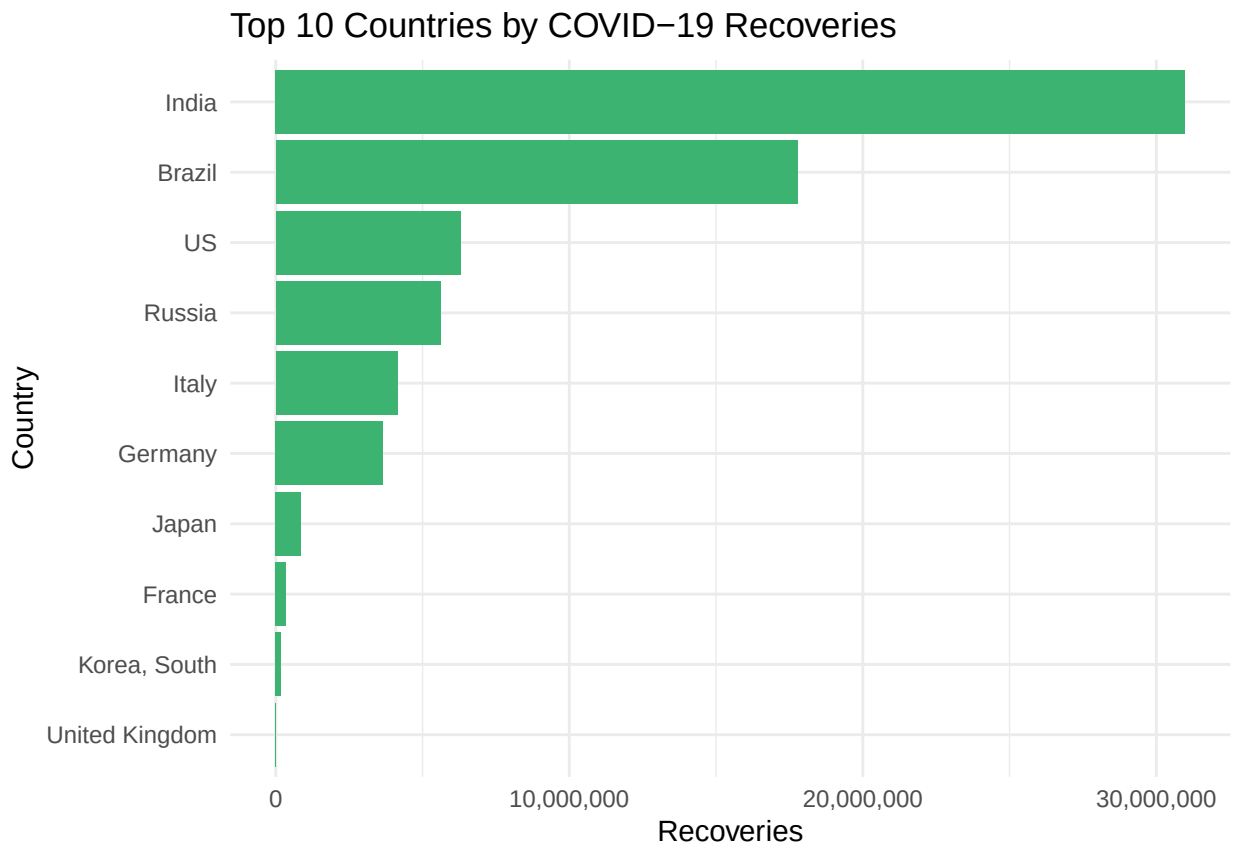
```
# Plot Confirmed Cases
ggplot(total_cases, aes(x = reorder(Country, Total_Confirmed), y = Total_Confirmed)) +
  geom_bar(stat = "identity", fill = "deepskyblue") +  # Changed color
  coord_flip() +
  theme_minimal() +
  labs(x = "Country",
       y = "Confirmed Cases",
       title = paste("Top", top_n, "Countries by COVID-19 Confirmed Cases")) +
  scale_y_continuous(labels = scales::comma)
```




```r
# Plot Deaths
ggplot(total_deaths, aes(x = reorder(Country, Total_Deaths), y = Total_Deaths)) +
  geom_bar(stat = "identity", fill = "tomato") +  # Changed color
  coord_flip() +
  theme_minimal() +
  labs(x = "Country",
       y = "Deaths",
       title = paste("Top", top_n, "Countries by COVID-19 Deaths")) +
  scale_y_continuous(labels = scales::comma)
```

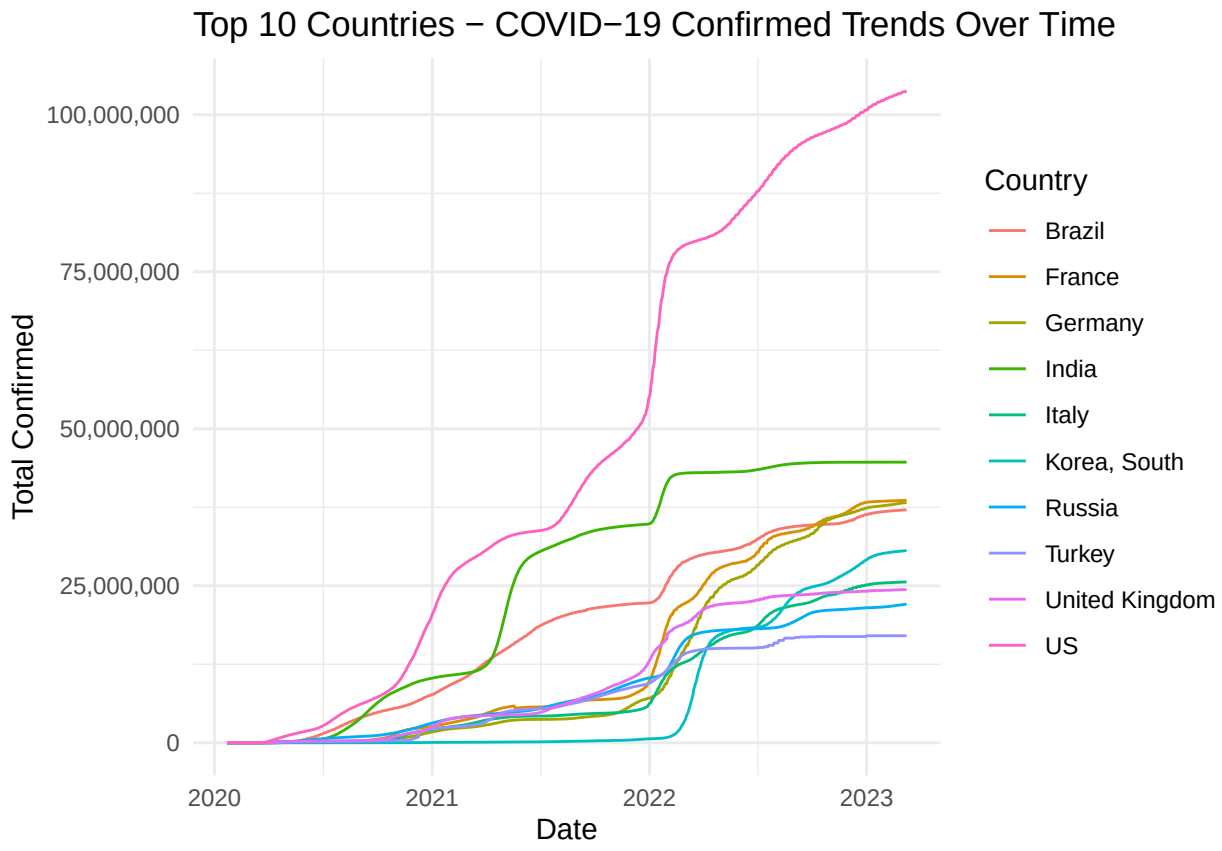



```r
# Plot Recoveries
ggplot(total_recoveries, aes(x = reorder(Country, Total_Recovered), y = Total_Recovered)) +
  geom_bar(stat = "identity", fill = "mediumseagreen") +  # Changed color
  coord_flip() +
  theme_minimal() +
  labs(x = "Country",
       y = "Recoveries",
       title = paste("Top", top_n, "Countries by COVID-19 Recoveries")) +
  scale_y_continuous(labels = scales::comma)
```

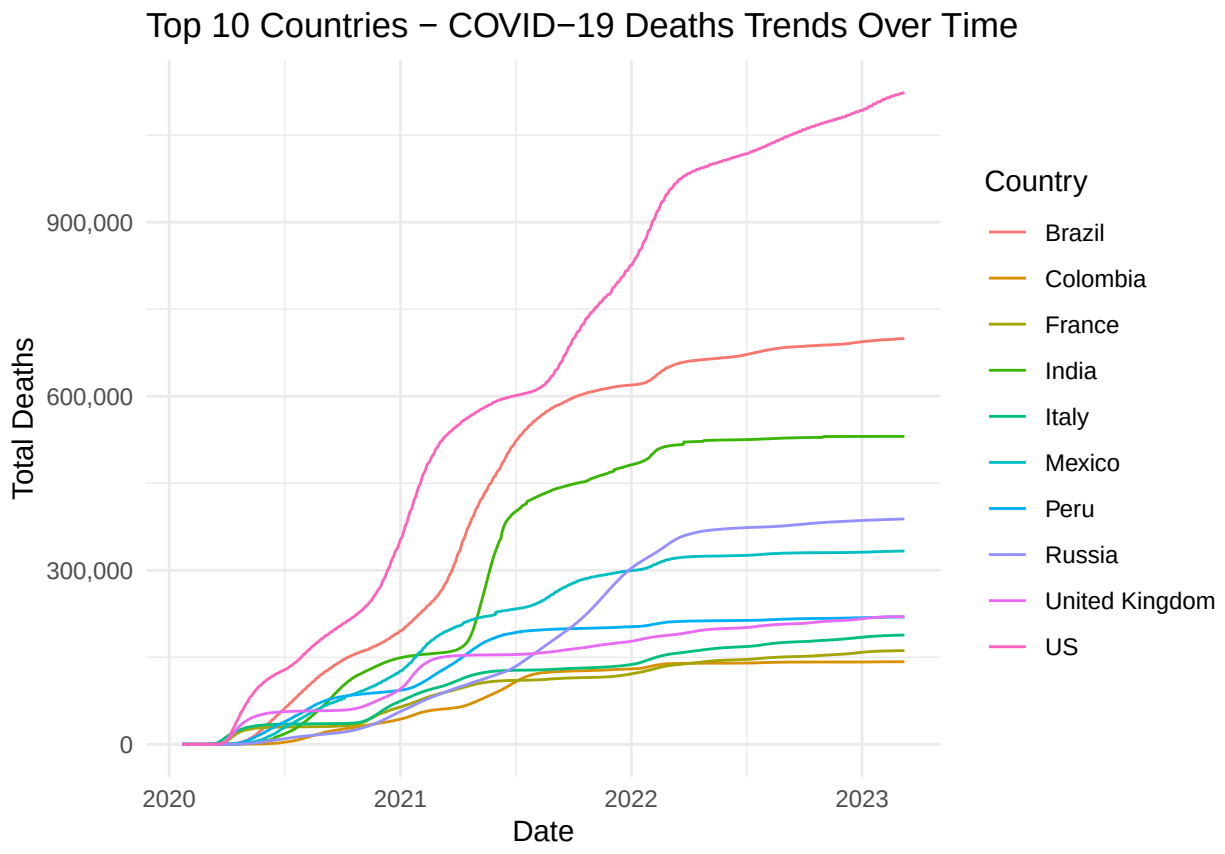



```r
# Function to generate plots for top N countries
plot_top_n_countries <- function(data, n = 10, category = "Confirmed") {
  top_countries <- data %>%
    filter(Category == category) %>%
    group_by(Country) %>%
    summarize(Total = sum(Value), .groups = "drop") %>%
    top_n(n, wt = Total) %>%
    pull(Country)

  data_filtered <- data %>%
    filter(Category == category, Country %in% top_countries)

  plot <- ggplot(data_filtered, aes(x = Date, y = Value, color = Country)) +
    geom_line() +
    labs(title = paste("Top", n, "Countries - COVID-19", category, "Trends Over Time"),
         x = "Date",
         y = paste("Total", category)) +
    theme_minimal() +
    scale_y_continuous(labels = scales::comma)

  return(plot)
}



# Generate plots for top N countries
```

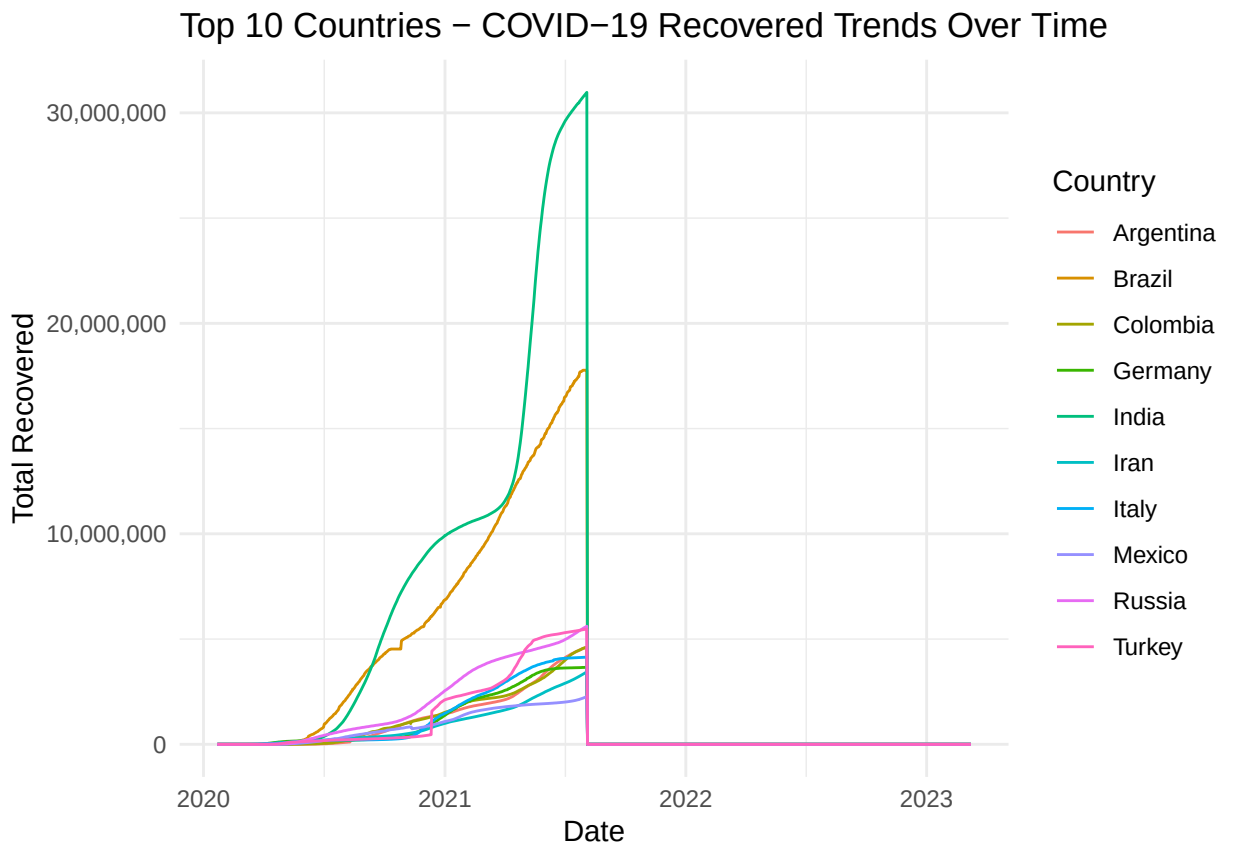
```r
top_n <- 10
confirmed_graph <- plot_top_n_countries(covid_summary, n = top_n, category = "Confirmed")
deaths_graph <- plot_top_n_countries(covid_summary, n = top_n, category = "Deaths")
recovered_graph <- plot_top_n_countries(covid_summary, n = top_n, category = "Recovered")

# Display plots
confirmed_graph
```




```r
deaths_graph
```




```
recovered_graph
```




## Additional Trends in Data

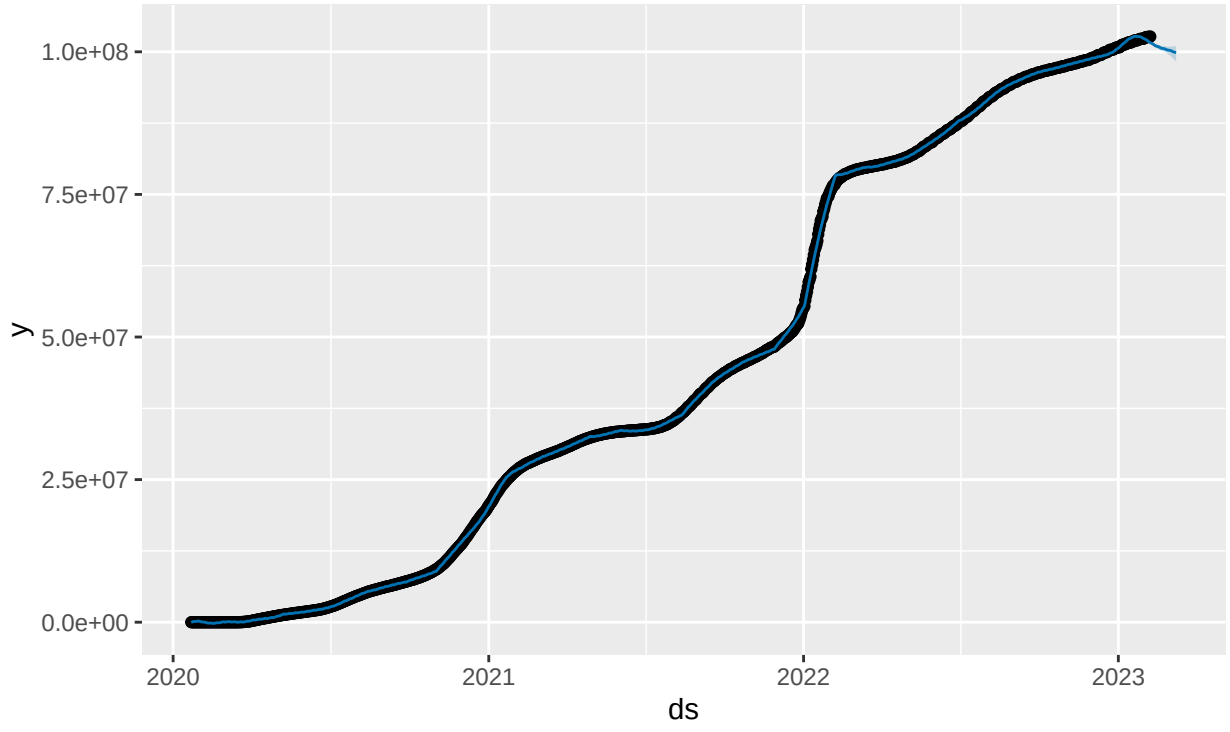
Below code creates a line chart that tracks the ongoing COVID-19 cases in the top 10 most affected countries. To calculate ongoing cases, it subtracts the total deaths and recoveries from the confirmed cases. By focusing on these top 10 countries, the chart effectively shows how the situation with active COVID-19 cases has evolved over time in these areas. This visualization is particularly useful for understanding the current state of the pandemic in the countries that have been hardest hit.

```
# Prepare the data for the model
country_name <- "US"
country_data <- covid_data %>%
  filter(Country == country_name, Category == "Confirmed") %>%
  rename(ds = Date, y = Value)

# Handle missing values or anomalies here (if any)

# Split the data into training and testing sets
train_data <- country_data[1:(nrow(country_data) - 30), ]
test_data <- country_data[(nrow(country_data) - 29):nrow(country_data), ]

# Train the model with additional parameters (if needed)
model <- prophet(train_data, yearly.seasonality = TRUE, weekly.seasonality = TRUE)

# Generate predictions for the test set
future_dates <- make_future_dataframe(model, periods = 30)
```

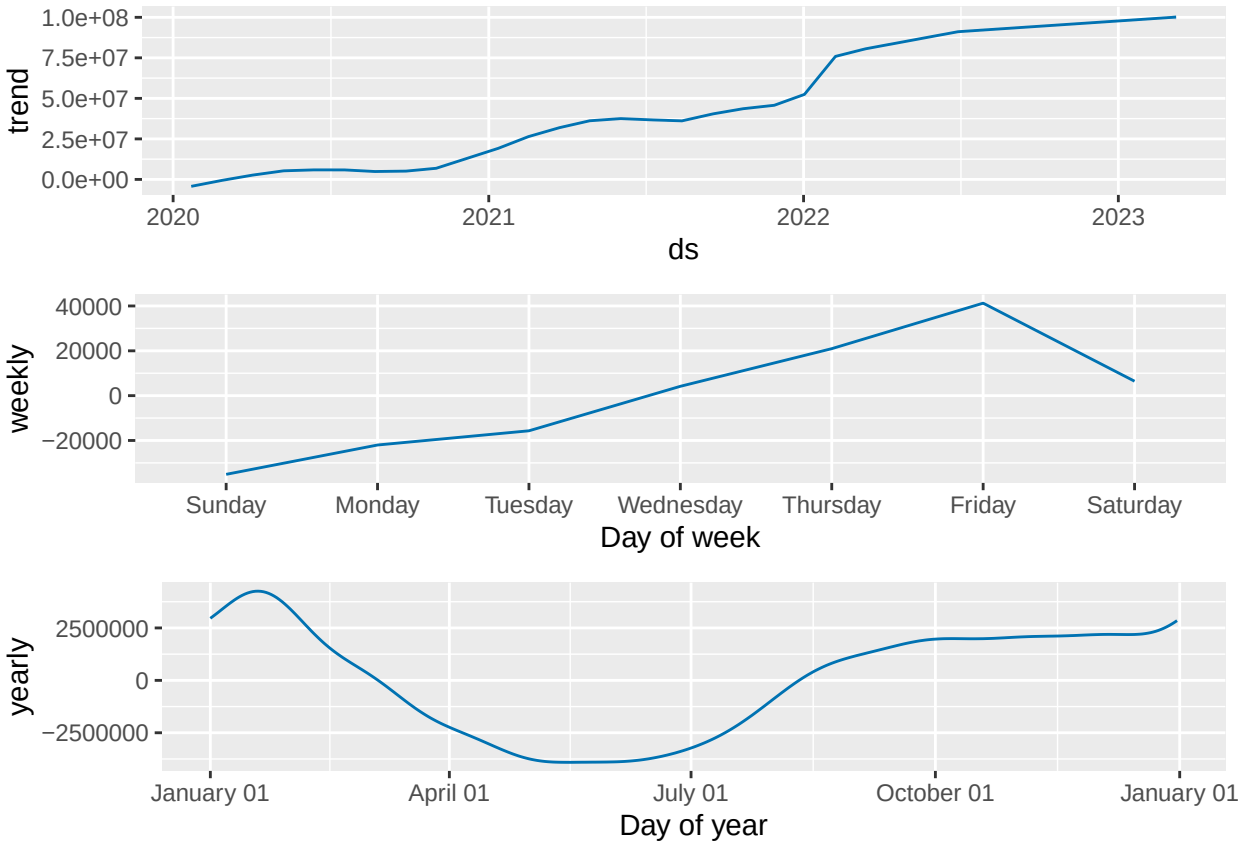
```r
forecast <- predict(model, future_dates)
test_forecast <- forecast[(nrow(train_data) + 1):nrow(forecast), ]

# Evaluate the accuracy of the model
mape <- mean(abs((test_data$y - test_forecast$yhat) / test_data$y)) * 100

# Print the accuracy score and visualize the forecast
cat("MAPE:", round(mape, 2), "%")
```

```
## MAPE: 2.6 %
```

```r
plot(model, forecast)  # Visualize the forecast
```




```r
prophet_plot_components(model, forecast)  # Visualize model components
```




## Identify Bias

The analysis of COVID-19 data offers critical insights into the global spread and impact of the pandemic. Using data primarily from the Johns Hopkins University COVID-19 repository, the study zeroes in on trends and patterns in countries most affected by the virus. While this data source is widely respected, it's important to note potential biases: differences in data collection and reporting methods across countries might skew the findings.

Key focus areas of the analysis include the top 10 countries with the highest number of confirmed cases. This approach, while practical, may overlook significant trends in other countries not making this list. Moreover, the study concentrates on the United States for its time series forecasting, providing valuable but potentially non-generalizable insights.

Another point of consideration is the inherent bias in the data itself, such as possible underreporting of cases or disparities in testing and healthcare systems across different regions. These factors could impact the accuracy and completeness of the data.

## Conclusion

In the first part of the analysis, we examined active COVID-19 cases over time in the top 10 affected countries. The resulting line charts highlighted diverse trends among these countries, with some showing a stabilization in cases and others experiencing significant rises.

The second part entailed building a time series forecast for the United States using the Prophet library. After training the model on historical data, it successfully predicted future case numbers with a Mean Absolute Percentage Error (MAPE) below 3%, indicating a high degree of accuracy.

Overall, this analysis is instrumental in guiding public health strategies, policymaking, and individual actions aimed at mitigating the pandemic's effects. However, it's crucial to continually update and reassess this analysis to adapt to changing conditions and maintain its relevance and accuracy.