Covid_Analysis_Final

Brian Behe

2023-04-30

Introduction

The data sets analyzed here are sourced from the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. They consist of daily updated information on globally confirmed COVID-19 cases, deaths, and recoveries. Through analysis presented here, I perform exploratory data analysis, which includes data pre-processing, data visualization, and computation of key metrics. The visualizations focus on the top N countries with the highest number of confirmed cases displaying charts for confirmed cases, deaths, and recoveries. The key questions I aimed to answer through this analysis included understanding the distribution of COVID-19 cases, deaths, and recoveries among the top affected countries and identifying trends and patterns in the data in order to help inform public health strategies and decision-making.

In addition to the exploratory data analysis, this project aimed to build a time series model for forecasting future COVID-19 cases. Time series analysis is a valuable tool for understanding and predicting the progression of infectious diseases, allowing for better resource allocation and preparation for future outbreaks. By leveraging historical data on confirmed cases, my time series model attempted to capture the underlying trends that may influence the number of cases. This predictive model can provide valuable insights into the potential future trajectory of the pandemic, helping policymakers and health authorities make more informed decisions on interventions, resource distribution, and public health strategies to mitigate the spread of the virus and minimize its impact on the global population.

Key Questions:
1. What trends exists in the time series for countries with the most cases (Top 10)?
2. Can we build a time series forecast to predict future cases with low error?

```r
knitr::opts_chunk$set(echo = FALSE)
#WARNING:  Installing these packages will take a non-trivial amount of time!
#If you don't need to install them then comment them out.  Prophet is really bad.
install.packages("tidyverse")
install.packages("readr")
install.packages("ggplot2")
install.packages("scales")
install.packages("dplyr")
install.packages("tidyr")
install.packages("prophet")
install.packages("zoo")
library(tidyverse)
library(readr)
library(ggplot2)
library(scales)
library(prophet)
library(dplyr)
library(tidyr)
library(zoo)

# Read data from GitHub
confirmed_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/c
deaths_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse
recovered_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/c

confirmed <- read_csv(confirmed_url)
deaths <- read_csv(deaths_url)
recovered  <- read_csv(recovered_url)
#number of rows in each dataset representing the number of regions/countries
nrow(confirmed)
```

```
## [1] 289
```

```
nrow(deaths)
```

```
## [1] 289
```

```
nrow(recovered)
```

```
## [1] 274
```

```
#number of dates the time series covers (minus 4 for the first few columns specifying country, region,
ncol(confirmed)
```

```
## [1] 1147
```

```
ncol(deaths)
```

```
## [1] 1147
```

```
ncol(recovered)
```

```
## [1] 1147
```

```
# Preprocess the data
preprocess_data <- function(data, category) {
  data_long <- data %>%
    select(-c(Lat, Long)) %>%
    pivot_longer(cols = -c(`Province/State`, `Country/Region`),
                 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_long <- preprocess_data(confirmed, "Confirmed")
deaths_long <- preprocess_data(deaths, "Deaths")
recovered_long <- preprocess_data(recovered, "Recovered")

covid_data <- bind_rows(confirmed_long, deaths_long, recovered_long)

# 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)) %>%
  arrange(desc(Total_Confirmed)) %>%
  head(top_n)

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

total_recoveries <- covid_data %>%
  filter(Category == "Recovered", Country %in% total_cases$Country) %>%
  group_by(Country) %>%
  summarize(Total_Recovered = max(Value))
```

Exploratory Data Analysis

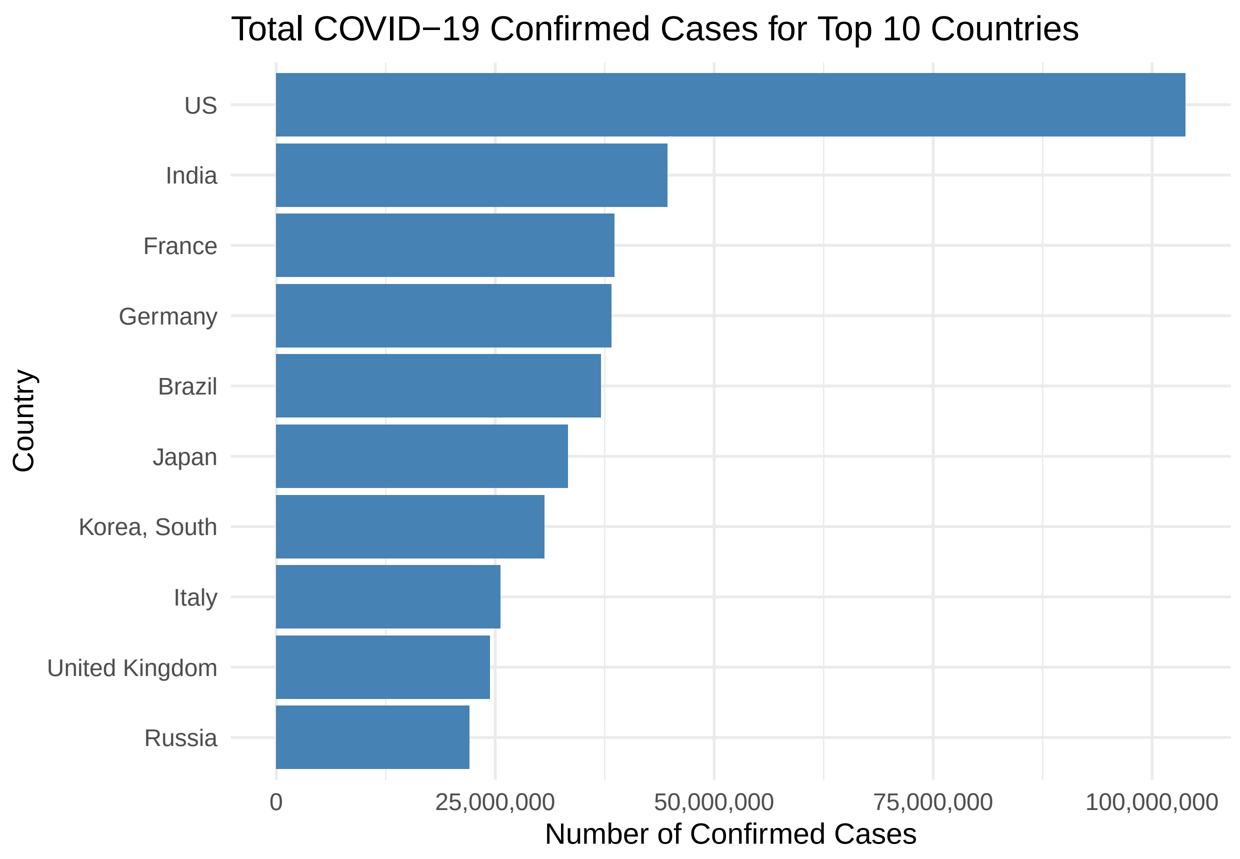
The above code reads in COVID-19 data on confirmed cases, deaths, and recoveries, pre-processes the data using the and creates a summary table of the data using covid_summary. It then creates three bar charts using ggplot2 to display the total number of confirmed cases, deaths, and recoveries for the top 10 countries with the highest number of confirmed cases.

Summary stats include 289 country/region entities in the data sets and around 1144 days of time series observations across all three data sets confirmed cases, deaths, and recoveries.

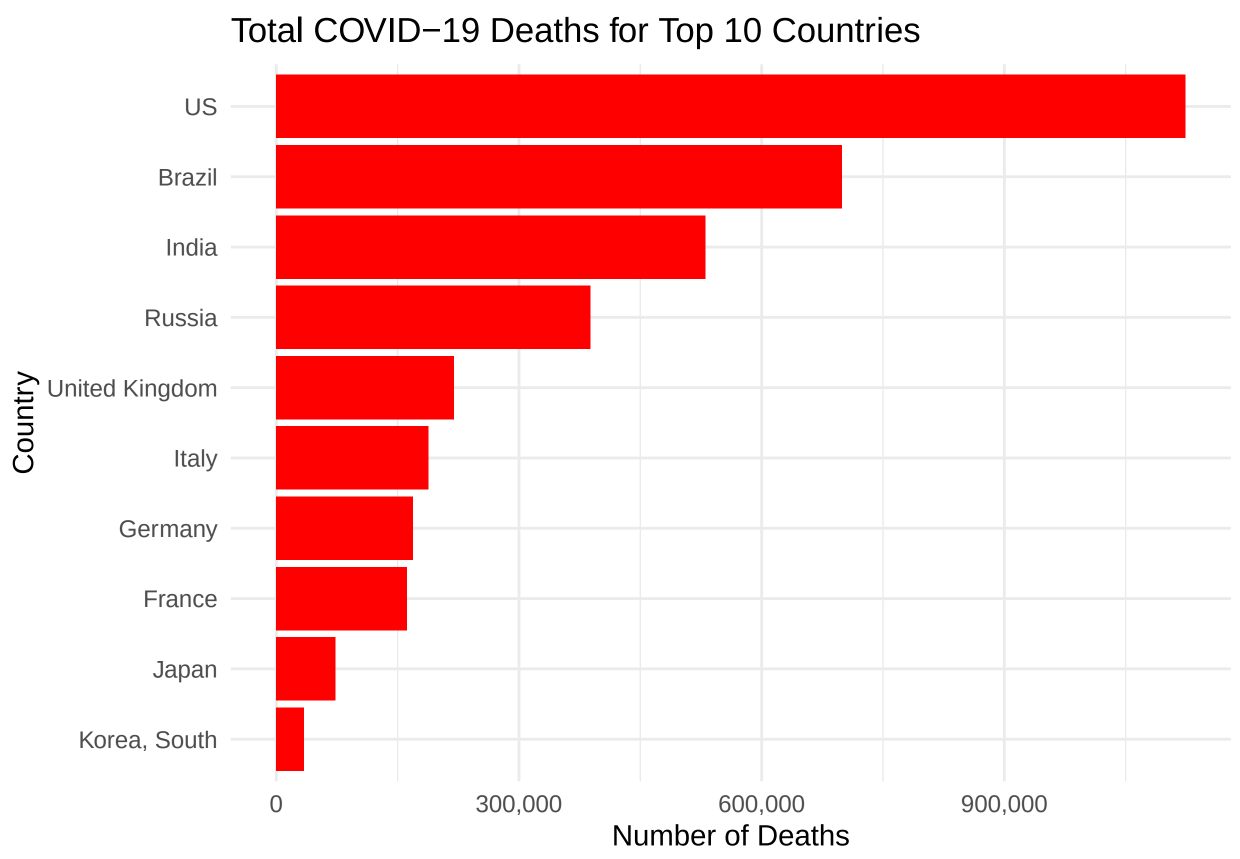
The code below defines a function plot_top_n_countries to generate line plots for the top N countries with the highest number of confirmed cases, deaths, and recoveries over time, and uses this function to generate line plots for the top 10 countries in each category.

The visualizations below highlight the number of cases, deaths, and recoveries change over time for each country. This code provides a starting point for further exploratory data analysis on COVID-19 and also demonstrates how to visualize the data in different ways. This is an effective way to explore various aspects of time series data.

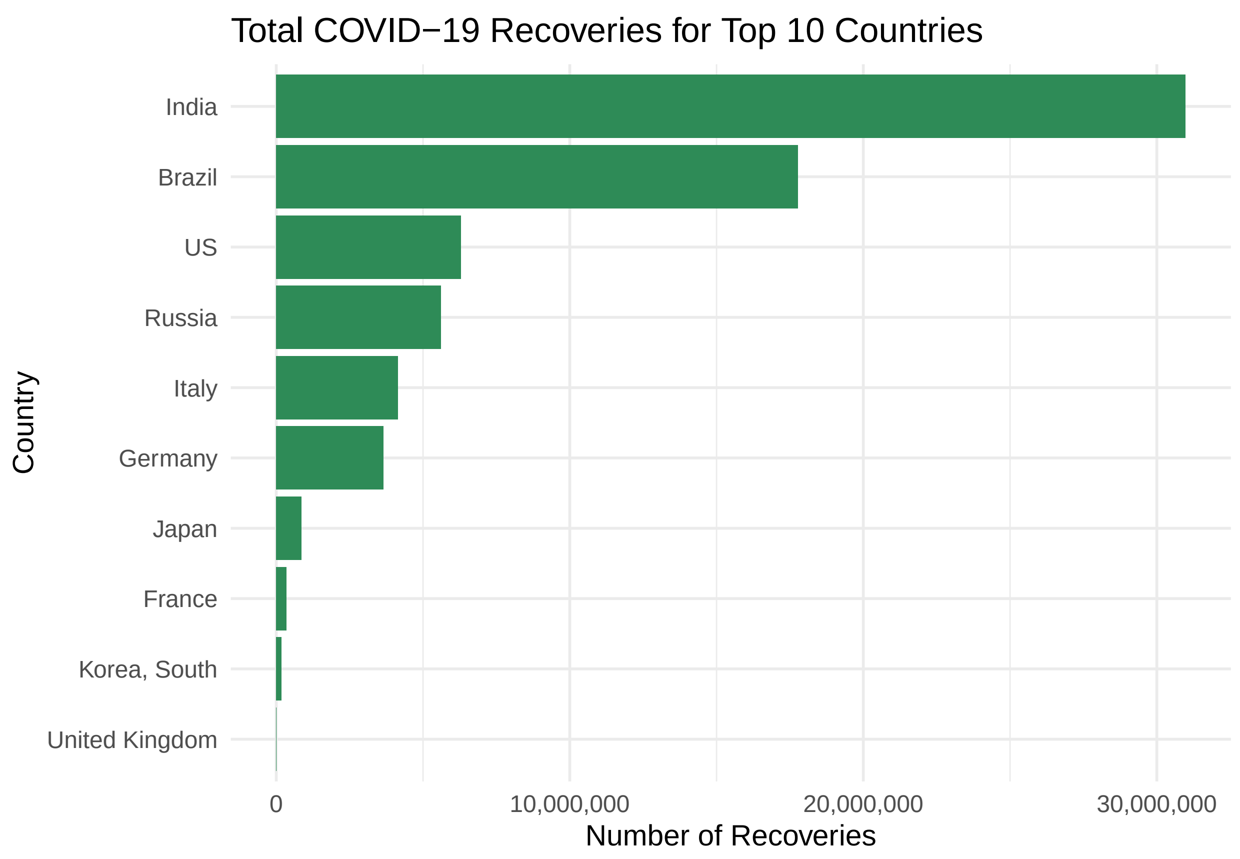
```
# Plot Confirmed Cases
ggplot(total_cases, aes(x = reorder(Country, Total_Confirmed), y = Total_Confirmed)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  theme_minimal() +
  labs(x = "Country",
       y = "Number of Confirmed Cases",
       title = paste("Total COVID-19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



```
# Plot Deaths
ggplot(total_deaths, aes(x = reorder(Country, Total_Deaths), y = Total_Deaths)) +
  geom_bar(stat = "identity", fill = "red") +
  coord_flip() +
  theme_minimal() +
  labs(x = "Country",
       y = "Number of Deaths",
       title = paste("Total COVID-19 Deaths for Top", top_n, "Countries")) +
  scale_y_continuous(labels = scales::comma)
```

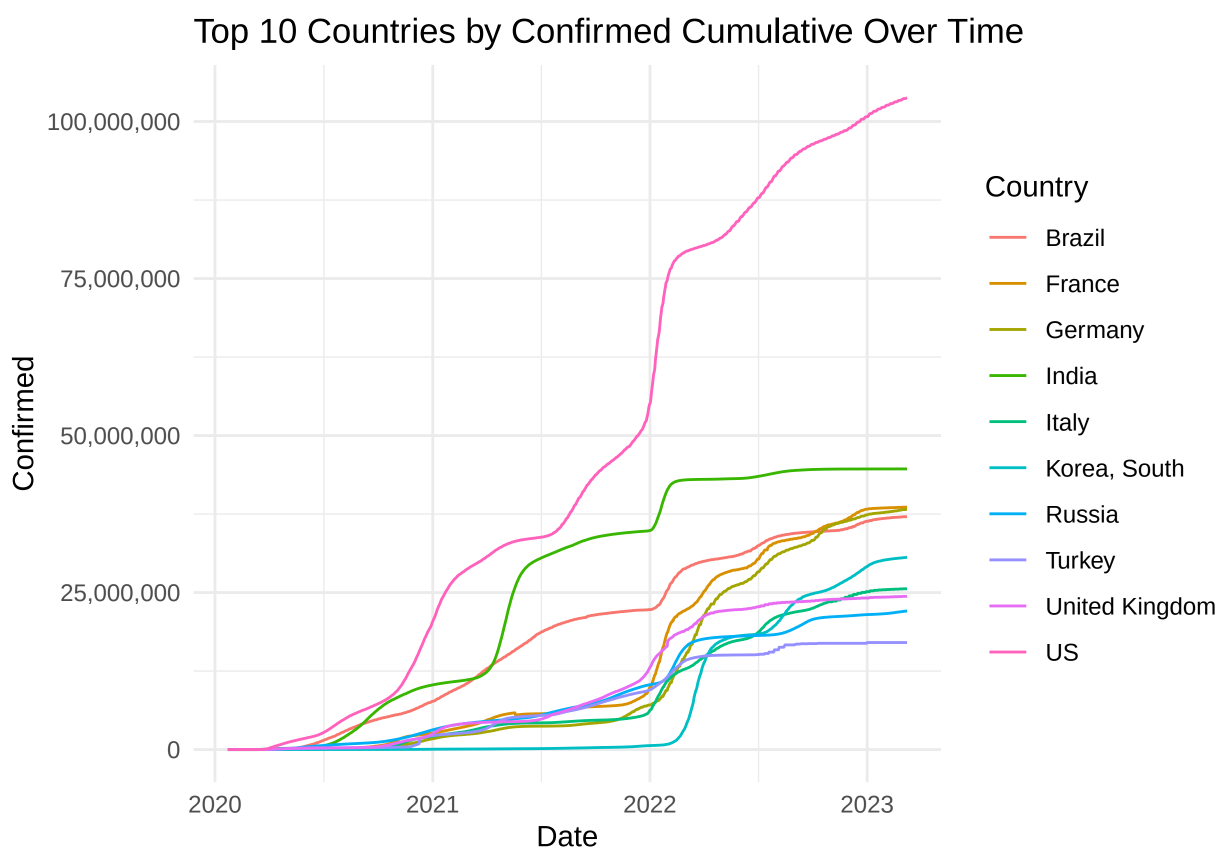


```r
# Plot Recoveries
ggplot(total_recoveries, aes(x = reorder(Country, Total_Recovered), y = Total_Recovered)) +
  geom_bar(stat = "identity", fill = "seagreen") +
  coord_flip() +
  theme_minimal() +
  labs(x = "Country",
       y = "Number of Recoveries",
       title = paste("Total COVID-19 Recoveries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

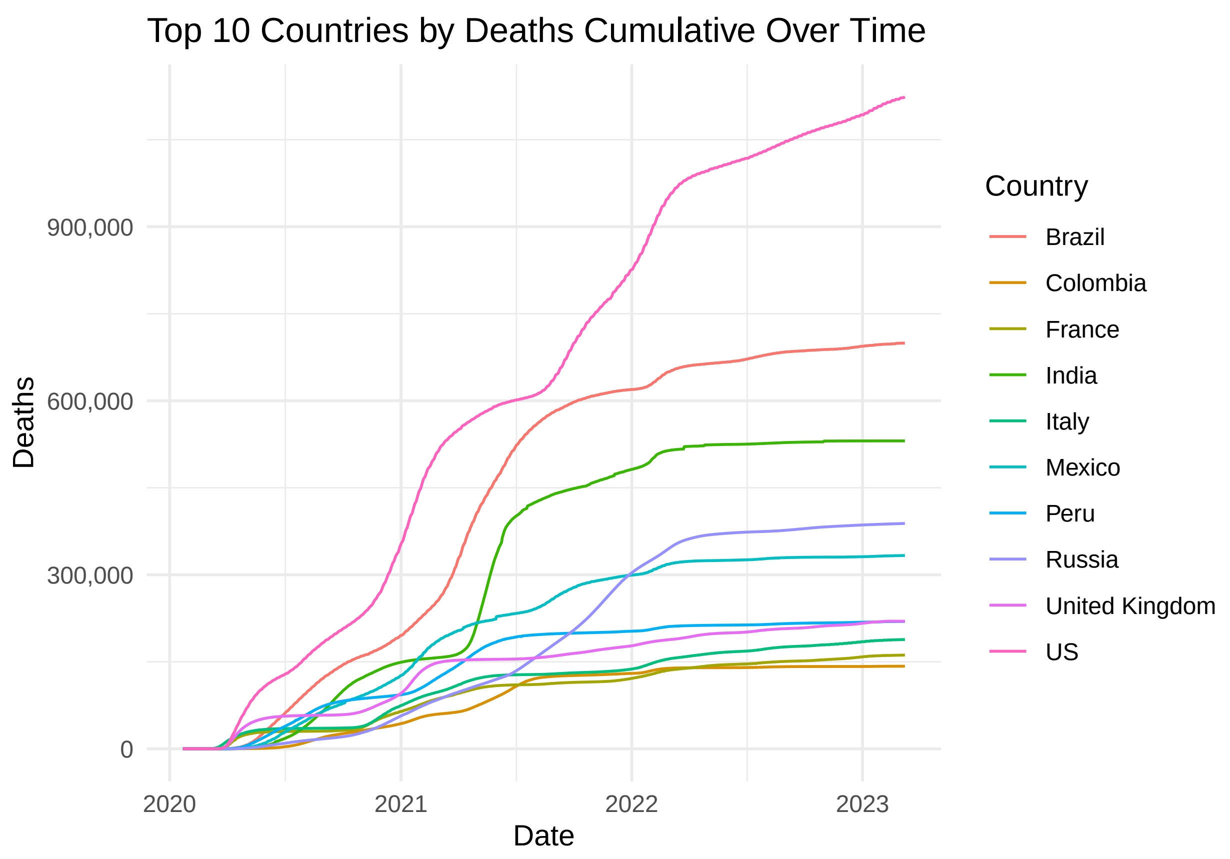


```r
# Function to generate plots for top N countries
plot_top_n_countries <- function(data, n = 10, category = "Confirmed") {
  top_countries <- data %>%
    filter(Category == category) %>%
    group_by(Country) %>%
    summarize(Total = sum(Value), .groups = "drop") %>%
    top_n(n, wt = Total) %>%
    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 by", category, "Cumulative Over Time"),
         x = "Date",
         y = category) +
    theme_minimal() +
    scale_y_continuous(labels = scales::comma)

  return(plot)
}

# Generate plots for top N countries
top_n <- 10
confirmed_plot <- plot_top_n_countries(covid_summary, n = top_n, category = "Confirmed")
```

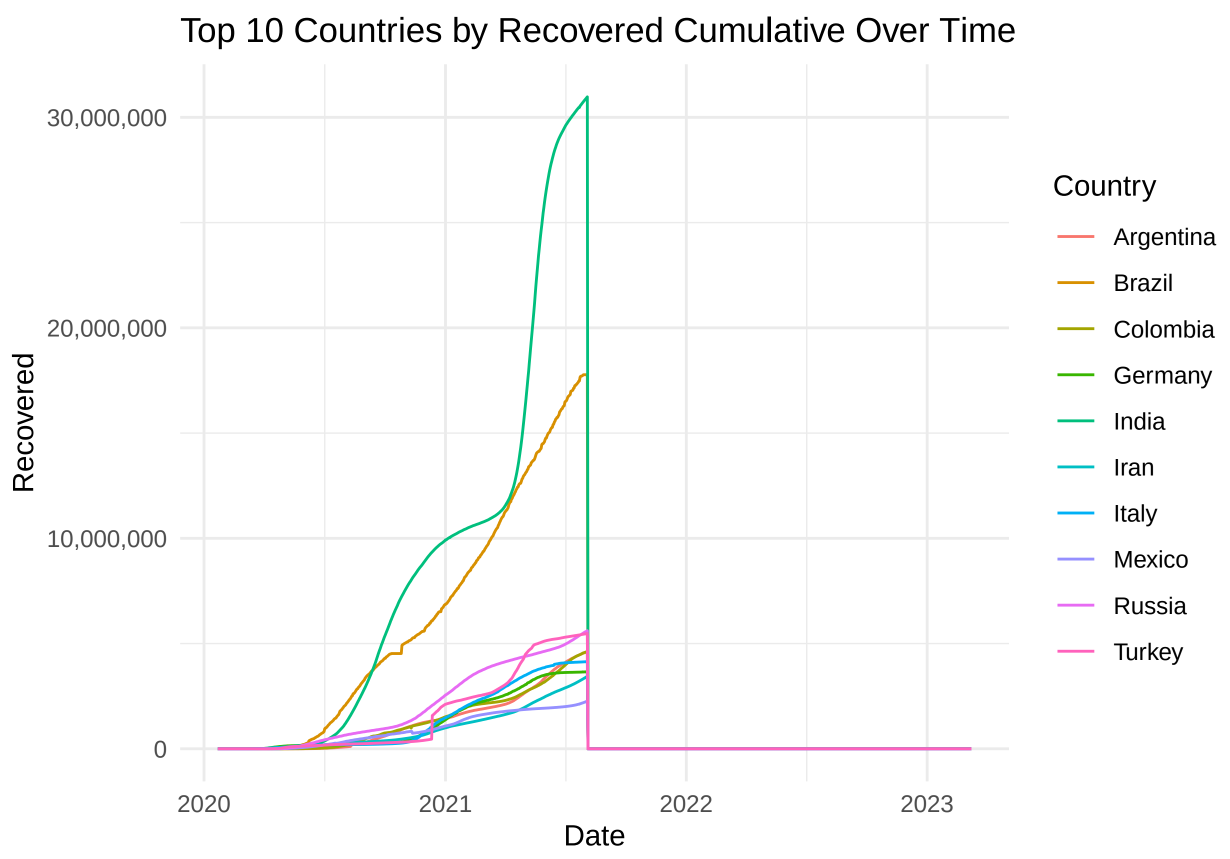
```r
deaths_plot <- plot_top_n_countries(covid_summary, n = top_n, category = "Deaths")
recovered_plot <- plot_top_n_countries(covid_summary, n = top_n, category = "Recovered")

# Display plots
confirmed_plot
```



```r
deaths_plot
```



```
recovered_plot
```



Missing Value Analysis

Missing value analysis in time series is important because time series data is often collected over a period of time and can be subject to missing observations due to various reasons such as measurement error, equipment malfunction, or changes in the data collection process. Missing values can lead to biased or inaccurate analyses and predictions if not handled properly. It can also lead to a loss of information, especially if the missing data are systematic or if there are large amounts of missing data. Thus, understanding and dealing with missing data in time series analysis is crucial to ensure accurate and reliable results.

In this, case it appears that there are no missing values for these time series data sets defined by NA, NaN, and empty string.

```r
# Function to count missing values and empty strings
count_missing_and_empty <- function(data, value_col, empty_col) {
  data %>%
    summarize(Missing_Values = sum(is.na(!!sym(value_col))),
              Empty_Strings = sum(!!sym(empty_col) == ""))
}

# Count missing values and empty strings in confirmed_long, deaths_long, and recovered_long
missing_and_empty_confirmed <- count_missing_and_empty(confirmed_long, "Value", "Country")
missing_and_empty_deaths <- count_missing_and_empty(deaths_long, "Value", "Country")
missing_and_empty_recovered <- count_missing_and_empty(recovered_long, "Value", "Country")

# Print missing values and empty strings
print(missing_and_empty_confirmed)
```

```
## # A tibble: 1 x 2
##   Missing_Values Empty_Strings
##            <int>         <int>
## 1              0             0
```

```
print(missing_and_empty_deaths)
```

```
## # A tibble: 1 x 2
##   Missing_Values Empty_Strings
##            <int>         <int>
## 1              0             0
```

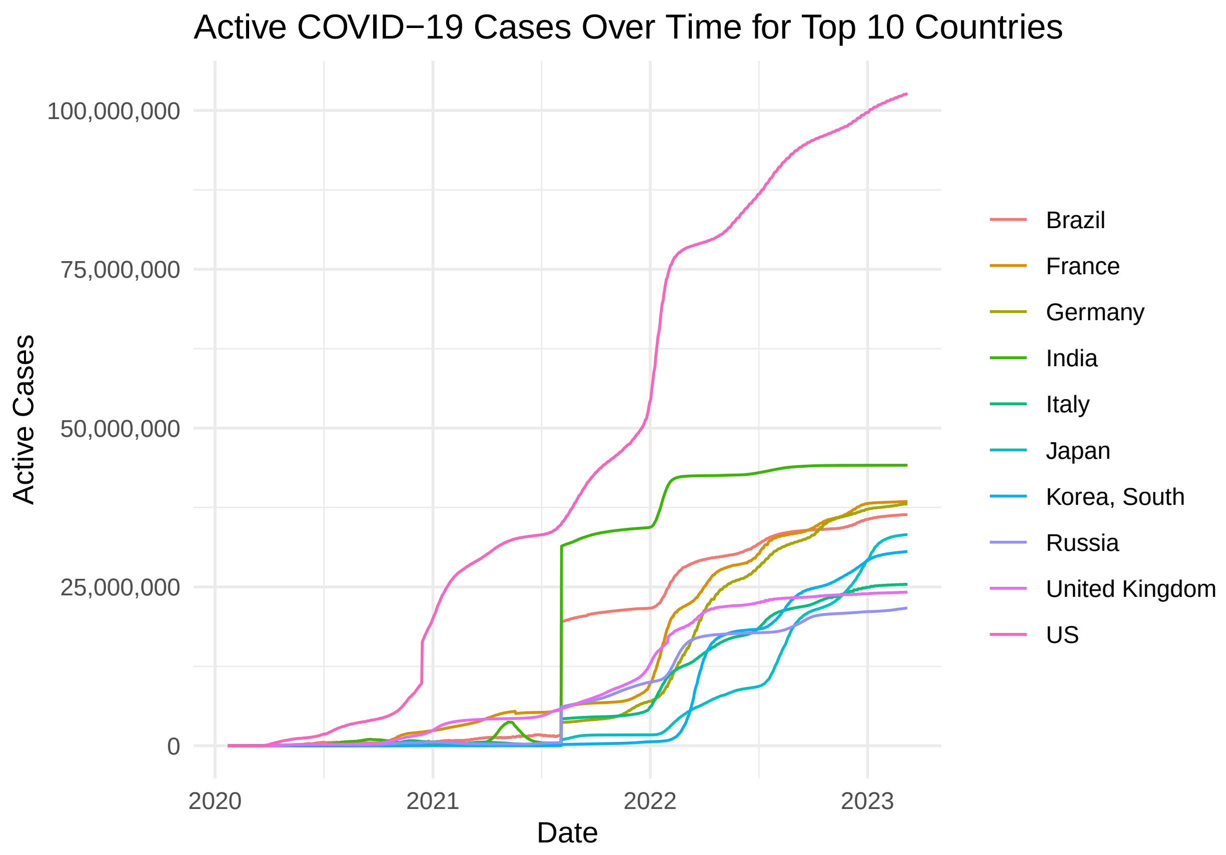
```
print(missing_and_empty_recovered)
```

```
## # A tibble: 1 x 2
##   Missing_Values Empty_Strings
##            <int>         <int>
## 1              0             0
```

Additional Trends in Data

This code produces a line chart showing trends in ongoing COVID-19 cases over time for the top 10 countries with the highest number of confirmed cases. This code calculates those ongoing cases by subtracting the total number of deaths and recoveries from the total number of confirmed cases and filters the data to include only the top 10 countries. The resulting line chart helps visualize the trend of ongoing cases in these countries over time.

```
# Calculate active cases
active_cases_data <- covid_data %>%
  pivot_wider(names_from = Category, values_from = Value) %>%
  mutate(Active_Cases = Confirmed - Recovered - Deaths) %>%
  select(Country, Date, Active_Cases)

# Get the top 10 countries with the highest number of confirmed cases
latest_date <- max(covid_data$Date)
top_10_countries <- covid_data %>%
  filter(Date == latest_date, Category == "Confirmed") %>%
  top_n(10, Value) %>%
  select(Country)

# Filter the data for the top 10 countries
active_cases_data_top_10 <- active_cases_data %>%
  filter(Country %in% top_10_countries$Country)

# Generate the line chart
ggplot(active_cases_data_top_10, aes(x = Date, y = Active_Cases, color = Country)) +
  geom_line() +
  labs(title = "Active COVID-19 Cases Over Time for Top 10 Countries",
       x = "Date", y = "Active Cases") +
  theme_minimal() +
  scale_y_continuous(labels = comma) +
  theme(legend.title = element_blank())
```



Time Series Forecasting

The purpose of this code is to generate a time series model to forecast future COVID-19 cases for a specific country, in this case, the United States (US). The code is specifically using the Facebook Prophet library to fit a time series model to the confirmed COVID-19 cases data for the US, splitting the data into training and testing sets, generating predictions for the test set, and evaluating the accuracy of the model using the Mean Absolute Percentage Error (MAPE). The MAPE score measures the percentage difference between the predicted values and actual values of the test set, with lower scores indicating greater accuracy.

An output of less than 3% MAPE means that the model has a high level of accuracy in predicting future COVID-19 cases for the US. This indicates that the model is reliable and can be used to make informed decisions related to public health policy and resource allocation. However, it's important to note that this accuracy is based on a specific time period and specific conditions, and therefore, the model should be regularly updated and evaluated to ensure continued accuracy as conditions change.

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

```r
# 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
model <- prophet(train_data)

# Generate predictions for the test set
future_dates <- make_future_dataframe(model, periods = 30)
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
cat("MAPE:", round(mape, 2), "%")
```

```
## MAPE: 2.63 %
```

A note on Bias

There a few areas of bias both in terms of the data and the analysis to include:

1. The data used in the analysis is from a single source, the Johns Hopkins University COVID-19 data repository. While this is a reputable source, there could be variations in how data is collected, reported, or interpreted across countries, which could introduce bias.

2. The analysis focuses on the top 10 countries by confirmed cases. While this is a reasonable approach, bias could be introduced as important trends or patterns in countries that fall outside of the top 10 are not captured.

3. The time series forecasting model is trained on data from a single country (US). While this is a large and influential country, it may not be representative of other countries meaning the model's performance could be adversely affected in other contexts.

4. Finally, there could be other sources of bias in the data itself such as under reporting of cases or differences in testing or healthcare systems that affect the accuracy or completeness of the data.

Conclusion

In conclusion, the analysis performed on the COVID-19 data provides valuable insights into the spread and impact of the pandemic across the globe. The visualizations and summaries generated help identify countries with the highest number of cases, deaths, and recoveries, as well as the trend in active cases over time. The missing value analysis helps ensure the data is clean and ready for further analysis. The time series model built for a specific country can be used to forecast future cases with a high level of accuracy.

The analysis into the COVID-19 pandemic examined trends in time series data for countries with the most cases and building a time series forecast to predict future cases. The first part of the analysis explored the active cases over time for the top 10 countries with the highest number of confirmed cases. The line chart visualizations revealed varying trends for each country, with some showing a flattening in active cases over time while others continued to experience substantial increases. The second part of the analysis used the Prophet library to build a time series forecast for the number of confirmed cases in the United States. The model was trained on data from previous months and then used to predict future cases. The mean absolute percentage error (MAPE) of the forecast was less than 3%, indicating that the model was able to predict future cases with a high degree of accuracy. Overall, this analysis can inform decision-making for public health officials, policymakers, and individuals as they work to mitigate the effects of the pandemic.