MSDS-5301: Week 5, Analysis on COVID-19 Data

2025-03-03

Introduction

In this analysis, we will import, tidy, and analyze COVID-19 data from Johns Hopkins University. The dataset includes information on confirmed cases, deaths, and recoveries from around the world. We will explore trends in the data, visualize them, fit a model to the data, and discuss potential biases.

Data Description

Source: JOHN HOPKINS COVID-19 DATA(click to see)

Description: The dataset contains daily time series summary tables, including confirmed, deaths and recovered COVID-19 cases. All data is read in from the daily case report.

Two time series tables are for the US confirmed cases and deaths, reported at the county level. Three time series tables are for the global confirmed cases, recovered cases, and deaths.

All the tables are given in CSV files. We are going to use four time series tables: two having the US confirmed cases and deaths and two having the global confirmed cases and deaths. There are a lot of columns in some of the tables. Therefore, we will provide the column level information for each table after we cleaned and formatted the tables.

Import Libraries

Let's import the necessary libraries to run this markdown file.

```r
# Turned off some flags to hide library load messages
# by using {r results='hide', message=FALSE, warning=FALSE}

# To install the packages uncomment the following two lines
# options(repos = c(CRAN = "https://cloud.r-project.org/"))
# install.packages(c("tidyverse", "lubridate", "ggplot2", "caret", "tinytex", "gridExtra"))
library(tidyverse)
library(lubridate)
library(ggplot2)
library(caret)
library(gridExtra) # Needed for grid
```

Read Data

To read the CSV files, first we collected the raw urls of the files by browsing the github repository stated in Data Description. All these files are under a parent url. We augmented each file name with the parent url and used a CSV reader to read the data.

```
# Parent url
# url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
# csse_covid_19_data/csse_covid_19_time_series/"

# Breaking down the url to fit in output window
url_part1 <- "https://raw.githubusercontent.com/"
url_part2 <- "CSSEGISandData/COVID-19/master/"
url_part3 <- "csse_covid_19_data/csse_covid_19_time_series/"

url_in <- paste0(url_part1, url_part2, url_part3)

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"
)

# Augment file names to parent url
urls <- str_c(url_in, file_names)

# Read data into R
us_cases <- read_csv(urls[1])
global_cases <- read_csv(urls[2])
us_deaths <- read_csv(urls[3])
global_deaths <- read_csv(urls[4])

# Check the first few rows of the U.S. confirmed data to ensure correct import
# colnames(us_cases)
```

Clean Data

In this step, we clean and format the data we loaded to R. The following operations are performed mostly like the way the Professor showed in the video “Tidying and Transforming Data” of Week 3.

```
# Filter global cases, drop unnecessary columns
global_cases <- global_cases %>%
  pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "cases") %>%
  select((-c(Lat, Long)))

# Filter global deaths, drop unnecessary columns
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select((-c(Lat, Long)))

# Join global_cases and global_deaths
global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%
  mutate(date=mdy(date))

# Filter rows where we have at least 1 case
global %>% dplyr::filter(cases > 0)
```

```
## # A tibble: 306,827 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-02-24     5      0
##  2 <NA>           Afghanistan    2020-02-25     5      0
##  3 <NA>           Afghanistan    2020-02-26     5      0
##  4 <NA>           Afghanistan    2020-02-27     5      0
##  5 <NA>           Afghanistan    2020-02-28     5      0
##  6 <NA>           Afghanistan    2020-02-29     5      0
##  7 <NA>           Afghanistan    2020-03-01     5      0
##  8 <NA>           Afghanistan    2020-03-02     5      0
##  9 <NA>           Afghanistan    2020-03-03     5      0
## 10 <NA>           Afghanistan    2020-03-04     5      0
## # i 306,817 more rows
```

```
# Filter US cases, drop unnecessary columns
# Mutate date to appropriate date format
us_cases <- us_cases %>%
  pivot_longer(cols = -c(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  mutate(date=mdy(date)) %>%
  select(-c(Lat, Long_))

# Filter US deaths, drop unnecessary columns
# Mutate date to appropriate date format
us_deaths <- us_deaths %>%
  pivot_longer(cols = -c(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date=mdy(date)) %>%
  select(-c(Lat, Long_))

# Join US cases and US deaths
us_all <- us_cases %>%
  full_join(us_deaths)

# Create a combined key for global data
global <- global %>%
  unite("Combined_Key",
    c(Province_State, Country_Region),
    sep = ", ",
    na.rm = TRUE,
    remove = FALSE)
```

```r
# Current global data do not include population
# uid_lookup_url <-"https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/
# heads/master/csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv"

# Breaking down the url to fit in output window
url_part1 <- "https://raw.githubusercontent.com/"
url_part2 <- "CSSEGISandData/COVID-19/refs/heads/master/"
url_part3 <- "csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv"

uid_lookup_url <- paste0(url_part1, url_part2, url_part3)

# Read filfe having population
uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, code3, iso2, iso3, Admin2))

# Join population from uid based on keys: "Province_State", "Country_Region", "Combined_Key"
# Select necessary columns only
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region", "Combined_Key")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths, Population, Combined_Key)
```

Summary of Cleaned Data

```r
# Summary of US cases and deaths
summary(us_all)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases            Population           deaths
##  Min.   :2020-01-22   Min.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-11-02   1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
##  Median :2021-08-15   Median :   2272   Median :   24892   Median :   37.0
##  Mean   :2021-08-15   Mean   :  14088   Mean   :   99604   Mean   :  186.9
##  3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

```r
# Summary of global cases and deaths
summary(global)
```

```
##  Province_State     Country_Region          date              cases
##  Length:330327      Length:330327      Min.   :2020-01-22   Min.   :        0
##  Class :character   Class :character   1st Qu.:2020-11-02   1st Qu.:      680
##  Mode  :character   Mode  :character   Median :2021-08-15   Median :    14429
##                                        Mean   :2021-08-15   Mean   :   959384
##                                         3rd Qu.:2022-05-28   3rd Qu.:   228517
##                                         Max.   :2023-03-09   Max.   :103802702
##
##      deaths           Population         Combined_Key
##  Min.   :      0   Min.   :6.700e+01   Length:330327
##  1st Qu.:      3   1st Qu.:5.866e+05   Class :character
##  Median :    150   Median :6.625e+06   Mode  :character
##  Mean   :  13380   Mean   :2.779e+07
##  3rd Qu.:   3032   3rd Qu.:2.655e+07
##  Max.   :1123836   Max.   :1.380e+09
##                    NA's   :11430
```

Exploratory Data Analysis

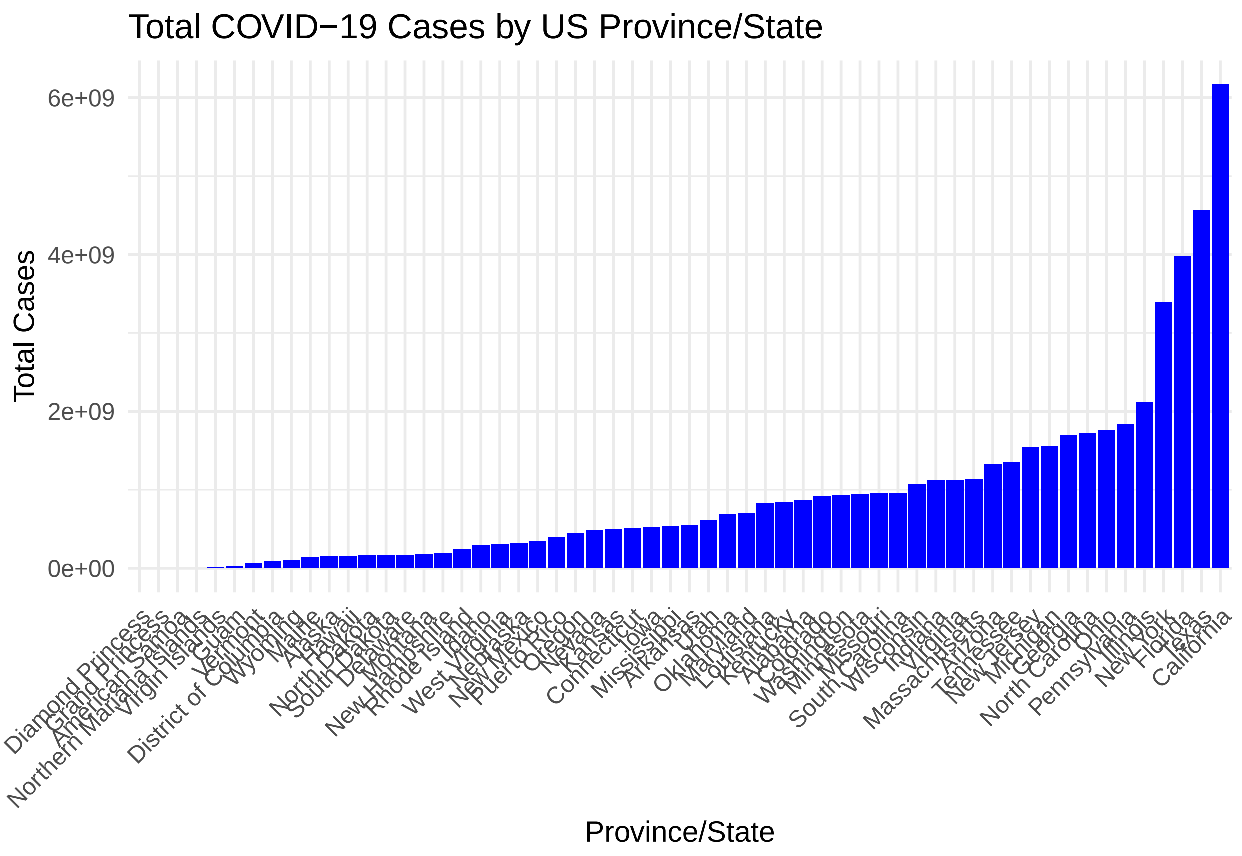
1. Total COVID-19 cases by US States/Territories

In this section, we are going to generate a plot displaying the State/Terriotry-wise COVID-19 cases in the United States. In this case, we are going to use a bar graph to represent the number of total case for each US State or territory.

```r
# Cast column date in us_all and global to type Data
us_all$date <- as.Date(us_all$date, format="%m/%d/%Y")
global$date <- as.Date(global$date, format="%m/%d/%Y")

# Group cases in US by Province_State and count total cases per group
# Additionally sort in descending order by cases
cases_by_state <- us_all %>%
  group_by(Province_State) %>%
  summarise(total_cases = sum(cases, na.rm = TRUE)) %>%
  arrange(desc(total_cases))

# Plotting the bar graph for cases by Province_State
ggplot(cases_by_state, aes(x = reorder(Province_State, total_cases), y = total_cases)) +
  geom_bar(stat = "identity", fill = "blue") +  # Bar graph with blue bars
  labs(title = "Total COVID-19 Cases by US Province/State",
       x = "Province/State",
       y = "Total Cases") +
  theme_minimal() +  # Clean theme
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



From the above visualization, we see that the most number of COVID-19 cases were observed in the state of California. Whereas the least number of COVID-19 cases were observed in the US cruise ship Diamond Princess.

2. Trend of Total COVID-19 Confirmed Cases and Deaths Worldwide Over Time

In this section, we are going to generate plots highlighting the trend of total deaths and total cases worldwide due to COVID-19. The number daily cases are very larger than the daily deaths. So, we are going to plot two line graphs side by side to see how the trends look like.

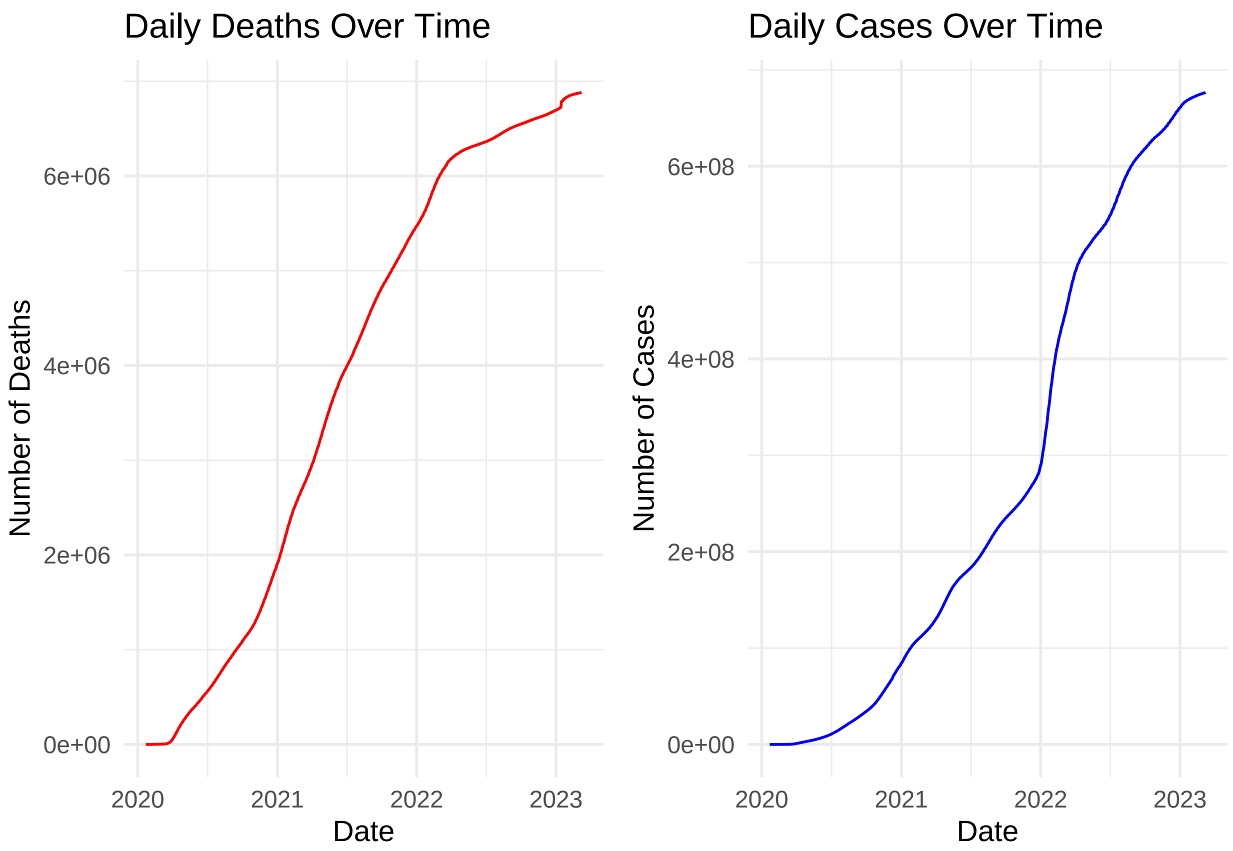
```r
# Aggregate data to sum up deaths and cases per day
deaths_per_day <- global %>%
  group_by(date) %>%
  summarise(daily_deaths = sum(deaths, na.rm = TRUE))

cases_per_day <- global %>%
  group_by(date) %>%
  summarise(daily_cases = sum(cases, na.rm = TRUE))

# Create the line plot for deaths per day
plot_deaths <- ggplot(deaths_per_day, aes(x = date, y = daily_deaths)) +
  geom_line(color = "red") +
  labs(title = "Daily Deaths Over Time",
       x = "Date",
       y = "Number of Deaths") +
  theme_minimal()
```

```r
# Create the line plot for cases per day
plot_cases <- ggplot(cases_per_day, aes(x = date, y = daily_cases)) +
  geom_line(color = "blue") +
  labs(title = "Daily Cases Over Time",
       x = "Date",
       y = "Number of Cases") +
  theme_minimal()

# Arrange the two plots side by side
grid.arrange(plot_deaths, plot_cases, ncol = 2)
```



The visualization above shows COVID-19 cumulative daily total deaths versus cumulative daily total cases worldwide over the years. If we look at the graphs, the trend in both plots is mostly similar except the number of cases reduced a bit in the second half of the year 2021.

Model Fitting: Predicting Deaths in US Based on Confirmed Cases and Population

In this section, we will try to fit a linear regression model to predict the number of deaths per Province/State in the United States based on the number of confirmed cases and the population of the Province/State. To do this, we will sample some data and create a 70:30 train and test split. Then we will fit a Linear Regression model on the train split and try to predict the number of deaths on the test split.

Training

```r
# Filter and group the data based on Province_State
sampled_data <- us_all %>%
  group_by(Province_State) %>%
  dplyr::filter(date == max(date)) %>%  # Get the latest date's data
  select(Province_State, cases, deaths, Population)  # Select relevant columns

# Remove any rows with missing values
sampled_data <- sampled_data %>%
  dplyr::filter(!is.na(cases) & !is.na(deaths) & !is.na(Population))

set.seed(123)

# We split the dataset into two parts: train and test with a ratio of 0.7 to 0.3.
train_index <- createDataPartition(sampled_data$deaths, p = 0.7, list = FALSE)
train_data <- sampled_data[train_index, ]
test_data <- sampled_data[-train_index, ]

# Fit a linear regression model: Predict deaths based on cases and population
model <- lm(deaths ~ cases + Population, data = train_data)

# Summary of the model to check coefficients, significance, etc.
summary(model)
```

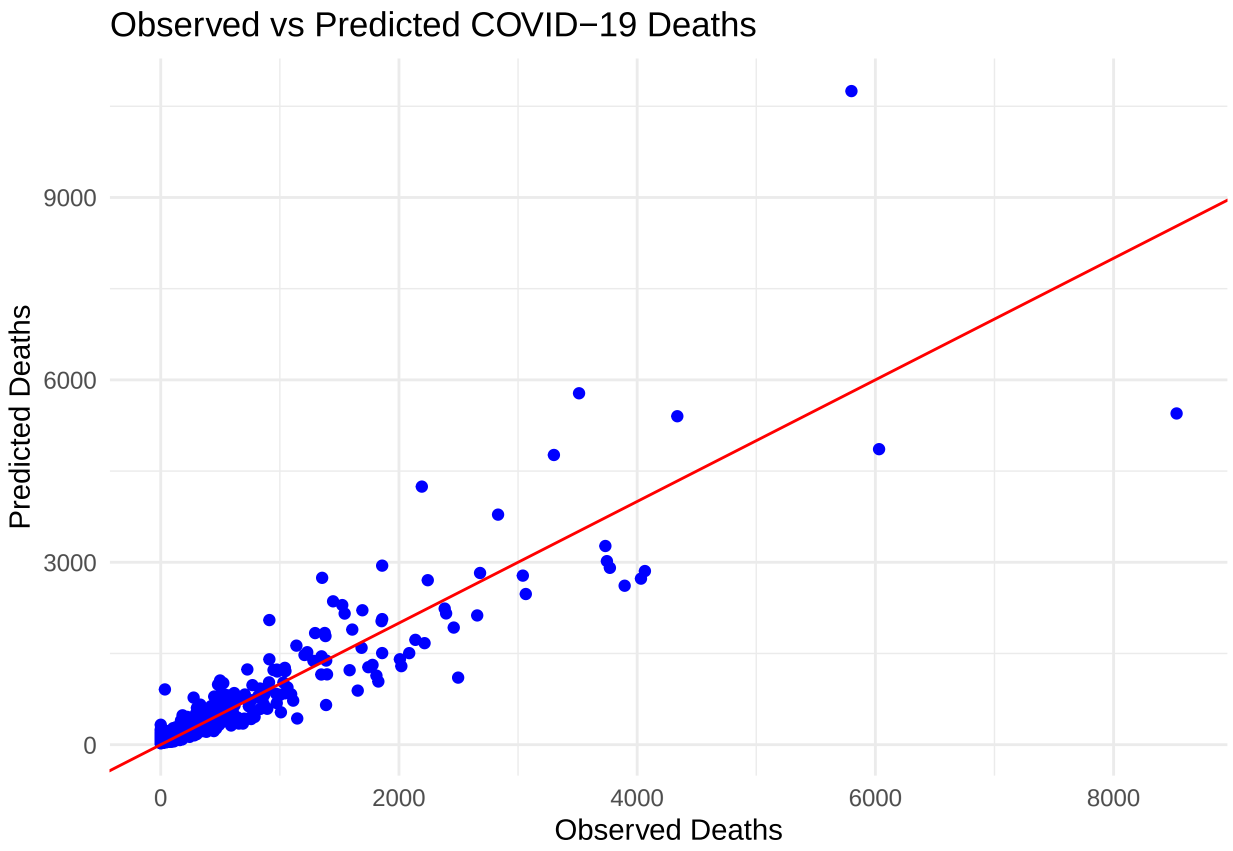
```
##
## Call:
## lm(formula = deaths ~ cases + Population, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2600.0   -28.1   -10.3    23.8  5573.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.650e+01  6.910e+00   3.835 0.000129 ***
## cases       8.694e-03  2.899e-04  29.992  < 2e-16 ***
## Population  4.317e-04  9.811e-05   4.400 1.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 320.1 on 2338 degrees of freedom
## Multiple R-squared:  0.9348, Adjusted R-squared:  0.9348
## F-statistic: 1.676e+04 on 2 and 2338 DF,  p-value: < 2.2e-16
```

Predicting

```r
# Make predictions using the model
test_data$predicted_deaths <- predict(model, newdata = test_data)

# Plot observed vs predicted deaths
```

```
ggplot(test_data, aes(x = deaths, y = predicted_deaths)) +
  geom_point(color = "blue") +  # Scatter plot for observed vs predicted deaths
  # Add a 45-degree line for perfect prediction
  geom_abline(intercept = 0, slope = 1, color = "red") +
  labs(title = "Observed vs Predicted COVID-19 Deaths",
       x = "Observed Deaths",
       y = "Predicted Deaths") +
  theme_minimal()
```



```
# Model performance: Compute R-squared and residuals
model_r_squared <- summary(model)$r.squared
model_r_squared
```

```
## [1] 0.9348067
```

Our Linear Regression model did quite good on predicting the deaths with a R-squared error of 0.9348.

Bias and Limitations of the Data

It's important to acknowledge potential biases and limitations in the data:

Testing Bias: Some countries may have better access to testing, leading to higher reported confirmed cases. Regions with limited testing resources may have fewer confirmed cases reported.

Underreporting: Deaths may be underreported, especially in countries with limited healthcare infrastructure or inconsistent reporting practices.

Data Completeness: Some regions may not report COVID-19 data consistently, and data may be missing or delayed.

Population Differences: Larger countries or those with dense populations may show higher absolute numbers of cases and deaths, but this may not reflect the actual per capita impact.

This analysis provides a useful overview of the trends in COVID-19 cases and deaths, as well as insight into the relationship between confirmed cases and deaths in the U.S. This type of analysis can be used for further modeling or policy planning.