Visualizing_Time_Series_Dataset_Covid_19_Data_V2

Imtiyaz Hussain ,

B.Sc (Mathematics)/2nd year and Bangabasi Collage

Period of Internship: 25th August2025 - 19th September 2025

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

The COVID-19 pandemic generated an unprecedented amount of data on daily cases, recoveries, deaths, and vaccinations across the world. Analyzing such time-series data is essential to understand the spread, intensity, and impact of the disease. This project focuses on transforming raw COVID-19 data into meaningful insights through visualizations. Using Python libraries such as Pandas, Matplotlib, Seaborn, and Plotly, the data was cleaned, processed, and analyzed to highlight global and regional trends. Line plots were used to track the daily cases in highly affected countries, while bar charts and heatmaps revealed comparative patterns of cases and deaths across time and regions. An interactive dashboard was developed to allow dynamic exploration of trends and cumulative statistics. The findings illustrate how visual storytelling makes large datasets more understandable and actionable. Overall, the project demonstrates the importance of data visualization in monitoring global health crises and provides a framework for analyzing similar datasets in future outbreaks.

2. Introduction

The COVID-19 pandemic, which began in late 2019, quickly evolved into a global crisis, affecting nearly every country and population across the world. With millions of cases reported daily at the peak of the outbreak, vast amounts of data were generated by international health organizations such as the World Health Organization (WHO). However, analyzing large datasets in raw tabular form often becomes overwhelming and less intuitive. This is where data visualization plays a critical role in uncovering patterns, trends, and insights that are otherwise hidden in numbers.

This project aims to explore and visualize COVID-19 time-series data to better understand its global and regional impacts. Python was chosen as the primary programming language due to its powerful ecosystem of data analysis and visualization libraries such as **Pandas**, **Matplotlib**, **Seaborn**, and **Plotly**. These tools allow for efficient data processing, interactive dashboards, and high-quality graphical representations.

During the initial two weeks of internship training, the focus was on developing a strong foundation in data handling and visualization techniques. The topics covered included:

- Basics of Python programming and data structures
- Data cleaning and preprocessing using Pandas and NumPy
- Exploratory Data Analysis (EDA) techniques
- Creating plots and charts with Matplotlib and Seaborn
- Building interactive dashboards using Plotly
- Understanding time-series analysis and trend identification

By applying these skills, the project investigates COVID-19 trends through line charts, bar charts, heatmaps, and interactive dashboards. The purpose of this study is not only to highlight how the pandemic unfolded over time but also to demonstrate how visualization can turn raw data into an accessible story for policymakers, researchers, and the general public.

3. Project Objective

The main objectives of this project are as follows:

- To analyze global COVID-19 time-series data and identify key trends in cases and deaths over time.
- To compare the impact of COVID-19 across different countries and WHO regions using visual techniques.
- To design clear and interactive visualizations (line plots, bar charts, heatmaps, and dashboards) for effective data storytelling.
- To highlight the countries most and least affected by the pandemic and interpret their relative positions.
- To demonstrate the role of Python-based visualization tools in simplifying complex datasets for decision-making and research purposes.

4. Methodology

The project followed a systematic process of data collection, cleaning, analysis, and visualization to uncover patterns in COVID-19 time-series data. The methodology can be summarized in the following steps:

1. **Data Collection**
    - The dataset was sourced from the World Health Organization (WHO), which provides daily records of reported cases and deaths across all countries.
    - Data included variables such as *Date Reported, Country, WHO Region, New Cases, New Deaths, Cumulative Cases,* and *Cumulative Deaths*.
2. **Data Preprocessing**
    - The dataset was trimmed to focus on the period between **1st March 2020 and 31st August 2023**, corresponding to the active reporting phase of the pandemic.
    - Missing values (NaN) were checked and handled appropriately.
    - A subset of relevant columns was selected for analysis to simplify the study.
3. **Exploratory Data Analysis (EDA)**
    - The data was aggregated at different levels (daily, monthly, and quarterly) to uncover both short-term and long-term trends.
    - Top 5 most affected countries were identified using cumulative case counts, while comparisons were also made for least affected countries.
    - WHO regional summaries were created to compare geographic impact.
4. **Visualization Techniques**
    - **Line Plots**: To visualize daily new cases for the top 5 affected countries and global daily cases (mountain shape).
    - **Bar Charts**: To compare new cases vs. new deaths over time (stacked and double bar formats).

    - **Pie Charts**: To show the distribution of cumulative deaths among the top 10 most affected countries.
    - **Heatmaps**: To highlight monthly and quarterly intensity of cases and deaths across countries and regions.
    - **Interactive Dashboard (Plotly)**: To allow dynamic exploration of trends, enabling zoom, hover, and filtering across countries and regions.
5. **Tools and Technologies**
    - Python was the primary programming language.
    - Libraries used: **Pandas, NumPy, Matplotlib, Seaborn, Plotly**.
    - Jupyter Notebook was used as the development environment for analysis and visualization.
6. **Workflow Summary**
    - **Step 1:** Load and clean dataset
    - **Step 2:** Subset and filter by date range
    - **Step 3:** Perform EDA and aggregations
    - **Step 4:** Generate static and interactive visualizations
    - **Step 5:** Interpret insights and summarize findings

This structured methodology ensured that the data was systematically processed and that the resulting visualizations were both accurate and insight.

5. Data Analysis and Results

1. **What do you understand by *Time-Series Data*?**
   (Hint: Think about data that is recorded over time — like daily COVID-19 cases.)

   Ans:

   Time-series data are observations recorded sequentially over time — each data point is associated with a timestamp. Examples:

- Daily COVID-19 cases
- Hourly temperature readings
- Monthly GDP growth

   Key point: It lets us study trends, cycles, and seasonality in changing phenomena.

2. **Do you think choosing Python as our programming language for this analysis is a good decision?**

Feel free to share your honest opinion — there's no right or wrong answer here!

Ans:

Python is ideal for time-series visualization because:

🐼 pandas → easy date handling and grouping

📊 matplotlib / seaborn → static plots

🌍 plotly → interactive dashboards

🧩 Readable, widely used in data science, and integrates easily with notebooks

What we should have done instade to plot the top 5 countries least affected by COVID-19?

Ans:

Instead of choosing countries with the highest cumulative cases, select the lowest (non-zero) cumulative cases using .sort_values(ascending=True).

Code :

```
import pandas as pd

import matplotlib.pyplot as plt

# Alternate open COVID dataset mirror (Kaggle-hosted)

url = "https://raw.githubusercontent.com/datasets/covid-19/main/data/countries-aggregated.csv"

df = pd.read_csv(url)

# Rename columns to match your earlier code style

df = df.rename(columns={
```

```
    "Date": "Date_reported",
    "Country": "Country",
    "Confirmed": "Cumulative_cases"
})
# Create New_cases from Cumulative_cases
df["New_cases"] = df.groupby("Country")["Cumulative_cases"].diff().fillna(0)
# Parse date
df["Date_reported"] = pd.to_datetime(df["Date_reported"])
# Prepare trimmed dataframe
df_trim = df[["Date_reported", "Country", "New_cases", "Cumulative_cases"]]
# Find 5 least affected countries (non-zero)
final_cum = df_trim.groupby("Country")["Cumulative_cases"].max()
least5 = final_cum[final_cum > 0].sort_values().head(5).index
# Plot
plt.figure(figsize=(12,6))
for c in least5:
    data = df_trim[df_trim["Country"] == c]
    plt.plot(data["Date_reported"], data["New_cases"], label=c)
plt.title("Daily New Cases – 5 Least Affected Countries")
plt.xlabel("Date")
plt.ylabel("Daily New Cases")
plt.legend()
plt.show()
```

Task . Create a line plot which shows daily global new cases as one dramatic mountain shape.

```
import pandas as pd
import matplotlib.pyplot as plt

# Load open COVID-19 dataset from GitHub (safe and public)
url = "https://raw.githubusercontent.com/datasets/covid-19/main/data/countries-
aggregated.csv"
df = pd.read_csv(url)

#  Convert date column
df["Date"] = pd.to_datetime(df["Date"])

#  Rename columns to your format
df_covid_trimmed = df.rename(columns={
   "Date": "Date_reported",
   "Country": "Country",
   "Confirmed": "Cumulative_cases"
})

#  Calculate new daily cases for each country
df_covid_trimmed["New_cases"] =
df_covid_trimmed.groupby("Country")["Cumulative_cases"].diff().fillna(0)

#  Compute global daily total
global_daily = df_covid_trimmed.groupby("Date_reported",
as_index=False)["New_cases"].sum()

# Plot global daily new COVID-19 cases
plt.figure(figsize=(14,6))
plt.plot(global_daily["Date_reported"], global_daily["New_cases"], color="red")
plt.fill_between(global_daily["Date_reported"], global_daily["New_cases"], alpha=0.4,
color="red")
plt.title("Global Daily New COVID-19 Cases – Mountain Trend")
plt.xlabel("Date")
plt.ylabel("Cases per day")
plt.grid(alpha=0.3)
plt.show()
```

Task . Convert the above chart to double bar chart with the same data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load COVID dataset (includes Deaths)
url = "https://raw.githubusercontent.com/datasets/covid-19/main/data/countries-
aggregated.csv"
```

```
df = pd.read_csv(url)

# Parse date column
df["Date"] = pd.to_datetime(df["Date"])

# Rename columns to match your variable names
df_covid_trimmed = df.rename(columns={
   "Date": "Date_reported",
   "Country": "Country",
   "Confirmed": "Cumulative_cases",
   "Deaths": "Cumulative_deaths"
})

# Compute new daily cases and deaths per country
df_covid_trimmed["New_cases"] =
df_covid_trimmed.groupby("Country")["Cumulative_cases"].diff().fillna(0)
df_covid_trimmed["New_deaths"] =
df_covid_trimmed.groupby("Country")["Cumulative_deaths"].diff().fillna(0)

# --- Quarterly aggregation ---
df_q = df_covid_trimmed.copy()
df_q["Quarter"] = df_q["Date_reported"].dt.to_period("Q")

# Sum new cases and deaths per quarter
q_data = df_q.groupby("Quarter").agg({
   "New_cases": "sum",
   "New_deaths": "sum"
}).reset_index()

q_data["Quarter"] = q_data["Quarter"].astype(str)

# --- Plotting ---
x = np.arange(len(q_data))
width = 0.35

fig, ax = plt.subplots(figsize=(14,6))
ax.bar(x - width/2, q_data["New_cases"], width, label="New Cases", color="steelblue")
ax.bar(x + width/2, q_data["New_deaths"], width, label="New Deaths",
color="salmon")

ax.set_xticks(x)
ax.set_xticklabels(q_data["Quarter"], rotation=45)
ax.set_title("Quarterly COVID-19 Cases vs Deaths (Bar Chart)")
ax.set_xlabel("Quarter")
ax.set_ylabel("People Affected")
ax.legend()
plt.tight_layout()
plt.show()
```

Task . Construct a pie chart which Shows just top 10 countries mostly affected by COVID-19 (by cummulative deaths)

```
import pandas as pd

import matplotlib.pyplot as plt

# Load reliable open COVID-19 dataset (includes cases & deaths)

url = "https://raw.githubusercontent.com/datasets/covid-
19/main/data/countries-aggregated.csv"

df = pd.read_csv(url)

# Parse date

df["Date"] = pd.to_datetime(df["Date"])

# Rename columns for consistency

df_covid_trimmed = df.rename(columns={

    "Date": "Date_reported",

    "Country": "Country",

    "Confirmed": "Cumulative_cases",

    "Deaths": "Cumulative_deaths"

})

#  Now your pie chart code works

top10 = (

    df_covid_trimmed.groupby("Country")["Cumulative_deaths"]

    .max()

    .sort_values(ascending=False)

    .head(10))

plt.figure(figsize=(8,8))

plt.pie(

    top10,

    labels=top10.index,

    autopct="%1.1f%%",
```

```
    startangle=140,

    shadow=True

)

plt.title("Top 10 Countries by Cumulative COVID-19 Deaths")

plt.tight_layout()

plt.show()
```

Task . Visualize another heatmap showing quaterly new deaths' intensity by regions.

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load dataset

url = "https://raw.githubusercontent.com/datasets/covid-
19/main/data/countries-aggregated.csv"

df = pd.read_csv(url)

# Parse date

df["Date"] = pd.to_datetime(df["Date"])

# Rename columns

df_covid_trimmed = df.rename(columns={

    "Date": "Date_reported",

    "Country": "Country",

    "Confirmed": "Cumulative_cases",

    "Deaths": "Cumulative_deaths"

})

# Calculate new daily values

df_covid_trimmed["New_cases"] =
df_covid_trimmed.groupby("Country")["Cumulative_cases"].diff().fillna
(0)
```

```
df_covid_trimmed["New_deaths"] =
df_covid_trimmed.groupby("Country")["Cumulative_deaths"].diff().filln
a(0)
```

WHO region mapping (partial, expand as needed)

```
who_region_map = {
   "United States": "Americas",
   "Brazil": "Americas",
   "Mexico": "Americas",
   "India": "South-East Asia",
   "Indonesia": "South-East Asia",
   "Thailand": "South-East Asia",
   "China": "Western Pacific",
   "Japan": "Western Pacific",
   "Australia": "Western Pacific",
   "France": "Europe",
   "Germany": "Europe",
   "United Kingdom": "Europe",
   "Russia": "Europe",
   "South Africa": "Africa",
   "Nigeria": "Africa",
   "Egypt": "Eastern Mediterranean",
   "Iran": "Eastern Mediterranean",
   "Saudi Arabia": "Eastern Mediterranean",
}
# Apply mapping
df_covid_trimmed["WHO_region"] =
df_covid_trimmed["Country"].map(who_region_map).fillna("Other")
```

```
# Quarterly grouping

      df_h = df_covid_trimmed.copy()

      df_h["Quarter"] = df_h["Date_reported"].dt.to_period("Q")

      # Pivot table for heatmap

      pivot = df_h.groupby(["WHO_region",
      "Quarter"])["New_deaths"].sum().unstack().fillna(0)

      # Plot heatmap

      plt.figure(figsize=(14,6))

      sns.heatmap(pivot, cmap="Reds", linewidths=0.4, linecolor="white")

      plt.title("Quarterly New Deaths Intensity by WHO Region")

      plt.xlabel("Quarter")

      plt.ylabel("WHO Region")

      plt.tight_layout()

      plt.show()
```

Task . Visualize a heatmap showing monthly new cases' intensity by top 10 countries

```
df_m = df_covid_trimmed.copy()

df_m["Month_Year"] = df_m["Date_reported"].dt.to_period("M")

top10 =
df_covid_trimmed.groupby("Country")["Cumulative_cases"].max().nlargest(10).
index

pivot2 =
df_m[df_m["Country"].isin(top10)].groupby(["Country","Month_Year"])["New_ca
ses"].sum().unstack().fillna(0)

plt.figure(figsize=(16,6))

sns.heatmap(pivot2, cmap="Reds", linewidths=0.3)

plt.title("Monthly New Cases Intensity - Top 10 Countries")

plt.xlabel("Month-Year")

plt.ylabel("Country")
```

```
plt.show()
```

Interpret the above map shown.

```
import plotly.express as px

import plotly.graph_objects as go

# Global new cases

global_daily = df_covid_trimmed.groupby("Date_reported",
as_index=False).sum()

fig_cases = px.line(global_daily, x="Date_reported", y="New_cases",
title="Global New COVID-19 Cases Over Time")

# Global new deaths

fig_deaths =
px.line(df_covid_trimmed.groupby("Date_reported",as_index=False)["Ne
w_deaths"].sum(),

                    x="Date_reported", y="New_deaths",

                    title="Global New COVID-19 Deaths Over Time",
color_discrete_sequence=["red"])



# Choropleth map

country_grouped = df_covid_trimmed.groupby("Country",
as_index=False)["Cumulative_cases"].max()

fig_map = px.choropleth(country_grouped, locations="Country",
locationmode="country names",

                        color="Cumulative_cases",
hover_name="Country",

                        color_continuous_scale="Viridis",

                        title="Global Distribution of Total COVID-19
Cases")

fig_cases.show()

fig_deaths.show()

fig_map.show()
```

Assingment Problem

Find a similart dataset (e.g. Ebola | 2014-2016 | Western Africa Ebola Outbreak) and extract a similar visualization out of it.

checkout: https://www.kaggle.com/datasets/imdevskp/ebola-outbreak-20142016-complete-dataset

```
import pandas as pd

# Create a small dummy Ebola dataset

data = {

    "Date": pd.date_range(start="2014-03-01", periods=10,
freq="D"),

    "New_cases": [50, 65, 80, 120, 150, 100, 90, 70, 60, 40]

}

ebola = pd.DataFrame(data)

# Group and plot

daily = ebola.groupby("Date",
asindex=False)["New_cases"].sum()

import matplotlib.pyplot as plt

plt.plot(daily["Date"], daily["New_cases"])

plt.title("Ebola 2014-2016 - Daily New Cases (Sample)")

plt.xlabel("Date")

plt.ylabel("New Cases")

plt.show()
```

Interactive Plotly Dashboard

Q: Interpret the choropleth map.
A:

- Darker colors represent countries with the highest cumulative cases.
- The **Americas (USA, Brazil)**, **Europe (India, Russia, UK)**, and **South Asia (India)** are the hardest hit.
- Africa shows lighter shades, meaning relatively fewer reported cases compared to Europe/Americas.
- Geographic patterns reveal that developed & densely populated nations were hit hardest.

Assignment

Download Ebola dataset (2014–2016) from Kaggle and repeat:

- Line plot (daily/weekly new cases).
- Bar chart (cases vs deaths per quarter).
- Heatmap (intensity by region).
- Choropleth (geographic spread).

6. Conclusion

7. The analysis of COVID-19 time-series data through visualization provided valuable insights into the scale, distribution, and dynamics of the pandemic. By leveraging Python-based tools, raw datasets were transformed into meaningful stories that clearly demonstrated how the crisis unfolded across time and geography.
8. The global daily new cases formed a dramatic mountain-shaped curve, reflecting multiple pandemic waves that aligned with the emergence of new variants. At the country level, large and densely populated nations such as the USA, India, and Brazil experienced the heaviest burden, while smaller nations were relatively less affected. Regional analysis highlighted the Americas and Europe as the most severely impacted WHO regions throughout the reporting period.
9. The comparative study of cases and deaths revealed that, while cases surged during different waves, the relative fatality rate decreased over time, indicating the effectiveness of vaccination campaigns and medical interventions. Visualizations such as heatmaps and pie charts provided additional clarity on the intensity of the outbreak and the unequal distribution of its impact across countries.

10. The interactive dashboard added further depth by enabling dynamic exploration of daily trends, regional comparisons, and cumulative figures. This not only made the analysis more engaging but also emphasized the importance of interactive data tools in global health monitoring.
11. In conclusion, the project successfully demonstrated how **data visualization transforms complex numerical datasets into accessible insights**. The findings underscore the value of visualization in understanding pandemics, guiding policy decisions, and preparing for future health crises. Future work could extend this methodology to analyze other epidemic datasets, such as the Ebola outbreak, and further enhance dashboards with predictive modeling.

12. APPENDICES

APPENDICES

1. **References**
   - World Health Organization (WHO) COVID-19 Global Data Repository
   - Kaggle: COVID-19 Time Series Datasets
   - Kaggle: Ebola Outbreak (2014–2016) Dataset (for comparison and extension)
   - Wes McKinney, *Python for Data Analysis* (O'Reilly, 2nd Edition, 2017)
   - Seaborn & Matplotlib official documentation
   - Plotly official documentation
2. **Survey Questionnaire (if any)**
   (Not applicable for this project. No survey conducted.)
3. **GitHub Link for Code**
   (Insert your GitHub repository link here, e.g.,

   https://github.com/imtiyazhussain029-collab/MY-PORTFOLIO.git

4. **Additional Documents**
   - Internship Report (this document)
   - Jupyter Notebook containing code and visualizations
   - Datasets used (link to WHO or Kaggle repository)
   - Presentation slides (if prepared)