

COVID-19 Data Analytics Dashboard

This project demonstrates how to analyze and visualize COVID-19 data using Power BI for business dashboards and Python (Streamlit + Pandas + Plotly) for analytics & interactive visualization.

data/WHO-COVID-Data.csv

- Download from WHO:



WHO-COVID-19-global-daily-data.csv

1 Columns in dataset:

- Date_reported → Date of report
- Country → Country name
- Country_code → 2-letter ISO code
- WHO_region → WHO region (e.g., Europe, Americas)
- New_cases → New daily cases
- Cumulative_cases → Total confirmed cases till date
- New_deaths → New daily deaths
- Cumulative_deaths → Total confirmed deaths till date

2 python/requirements.txt

- pandas
- matplotlib
- seaborn
- plotly
- streamlit

3 python/covid_dashboard.py

- This script creates an **interactive dashboard using Streamlit**

```
import streamlit as st
```

```
import pandas as pd
```

```
import plotly.express as px
```

```
# Load data

# The WHO file columns are: Date_reported, Country_code, Country, WHO_region,
# New_cases, Cumulative_cases, New_deaths, Cumulative_deaths
# Map them to simplified names used in the app.

data = pd.read_csv('WHO-COVID-19-global-daily-data.csv')

# Normalize column names to expected ones

column_mapping = {
    'Date_reported': 'date',
    'Country': 'location',
    'Cumulative_cases': 'total_cases',
}

# Ensure required columns exist

missing_required = [src for src in ['Date_reported', 'Country', 'Cumulative_cases'] if src not in data.columns]

if missing_required:
    st.error(f'Missing required columns in CSV: {missing_required}')
else:
    data = data.rename(columns=column_mapping)

    # Parse date and sort

    data['date'] = pd.to_datetime(data['date'], errors='coerce')

    data = data.dropna(subset=['date'])

    data = data.sort_values('date')

st.title("COVID-19 Dashboard")
```

```

# Country selector

countries = data["location"].dropna().unique()

country = st.selectbox("Select a Country", sorted(countries))

df_country = data[data["location"] == country]

# Line Chart for cumulative cases

if not df_country.empty:

    fig = px.line(df_country, x="date", y="total_cases", title=f"COVID-19 Total Cases in {country}")

    st.plotly_chart(fig, use_container_width=True)

else:

    st.info("No data available for the selected country.")

# Bar Chart (Top 10 Countries Latest Cases)

latest_date = data['date'].max()

latest = data[data["date"] == latest_date]

top10 = latest.sort_values("total_cases", ascending=False).head(10)

if not top10.empty:

    fig2 = px.bar(top10, x="location", y="total_cases", title=f"Top 10 Countries by Total Cases on {latest_date.date()}")

    st.plotly_chart(fig2, use_container_width=True)

else:

    st.info("No latest snapshot available to display top 10 chart.")

```

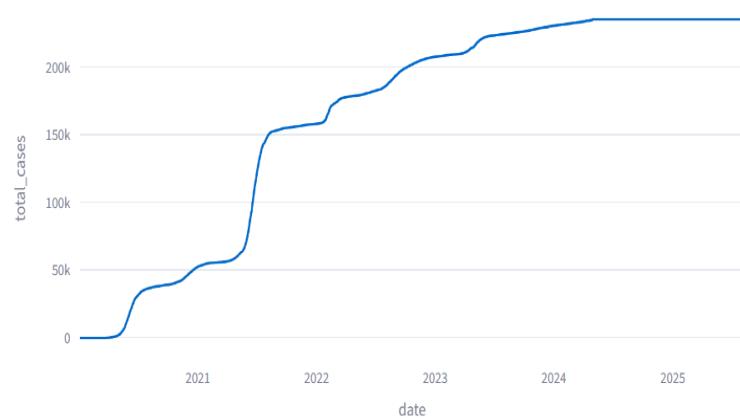
OUTPUT:

COVID-19 Dashboard

Select a Country

Afghanistan

COVID-19 Total Cases in Afghanistan



COVID-19 Dashboard

Select a Country

Afghanistan

Afghanistan

Albania

Algeria

American Samoa

Andorra

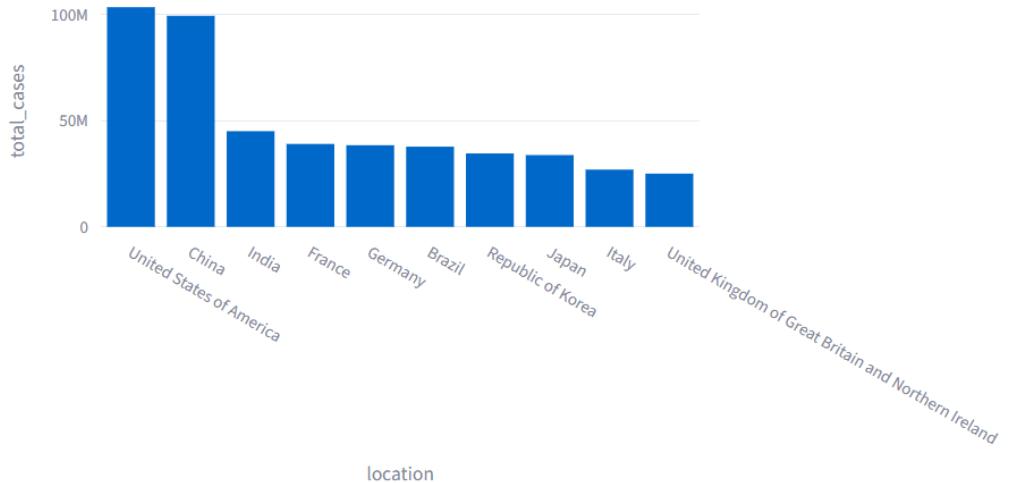
Angola

Anguilla

Antigua and Barbuda



Top 10 Countries by Total Cases on 2025-08-10



python/covid.py

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

data = pd.read_csv('WHO-COVID-19-global-daily-data.csv')

print(data.head())

# Map WHO columns to expected names

column_mapping = {

    'Date_reported': 'date',

    'Country': 'location',

    'Cumulative_cases': 'total_cases',

    'New_cases': 'new_cases',

    'Cumulative_deaths': 'total_deaths',}
```

```
'New_deaths': 'new_deaths',  
}  
  
missing_required = [src for src in column_mapping.keys() if src not in  
data.columns]  
  
if missing_required:  
    raise ValueError(f"Missing required columns in CSV:  
{missing_required}")  
  
# Rename and type-clean  
data = data.rename(columns=column_mapping)  
  
# Parse date and sort  
data['date'] = pd.to_datetime(data['date'], errors='coerce')  
  
# Keep only used columns (people_vaccinated not present in WHO file)  
used_columns = [  
    'date', 'location', 'total_cases', 'new_cases', 'total_deaths',  
    'new_deaths'  
]  
  
# Coerce numeric columns  
for col in ['total_cases', 'new_cases', 'total_deaths', 'new_deaths']:  
    data[col] = pd.to_numeric(data[col], errors='coerce')  
  
# Drop rows with invalid date or missing location  
data = data.dropna(subset=['date', 'location']).sort_values('date')  
  
# Fill remaining NaNs with 0 for plotting
```

```

data[used_columns] = data[used_columns].fillna(0)

india = data[data['location'] == 'India']

plt.figure(figsize=(10,6))

if not india.empty:

    plt.plot(india['date'], india['total_cases'], label='Total Cases',
color='blue')

    plt.plot(india['date'], india['total_deaths'], label='Total Deaths',
color='red')

    plt.xlabel('Date')

    plt.ylabel('Count')

    plt.title('COVID-19 Cases & Deaths in India')

    plt.legend()

    plt.tight_layout()

    plt.show()

else:

    print('No India data available to plot time series.')

latest = data[data['date'] == data['date'].max()]

top10 = latest.sort_values('total_cases', ascending=False).head(10)

plt.figure(figsize=(12,6))

if not top10.empty:

    sns.barplot(x='location', y='total_cases', data=top10,
palette='Blues_r')

    plt.xticks(rotation=45, ha='right')

    plt.title('Top 10 Countries by Total Cases')

    plt.tight_layout()

```

```

plt.show()

else:

    print('No latest snapshot available for Top 10 chart.')

corr_cols = ['total_cases', 'new_cases', 'total_deaths', 'new_deaths']

if not india.empty and india[corr_cols].dropna().shape[0] > 1:

    corr = india[corr_cols].corr()

    plt.figure(figsize=(8, 6))

    sns.heatmap(corr, annot=True, cmap='coolwarm')

    plt.title('Correlation Heatmap (India COVID Data)')

    plt.tight_layout()

    plt.show()

else:

    print('Not enough India data to compute correlation heatmap.')

if not latest.empty:

    fig = px.choropleth(

        latest,

        locations='location',

        locationmode='country names',

        color='total_cases',

        hover_name='location',

        color_continuous_scale='Reds',

        title='Worldwide COVID-19 Cases'

    )

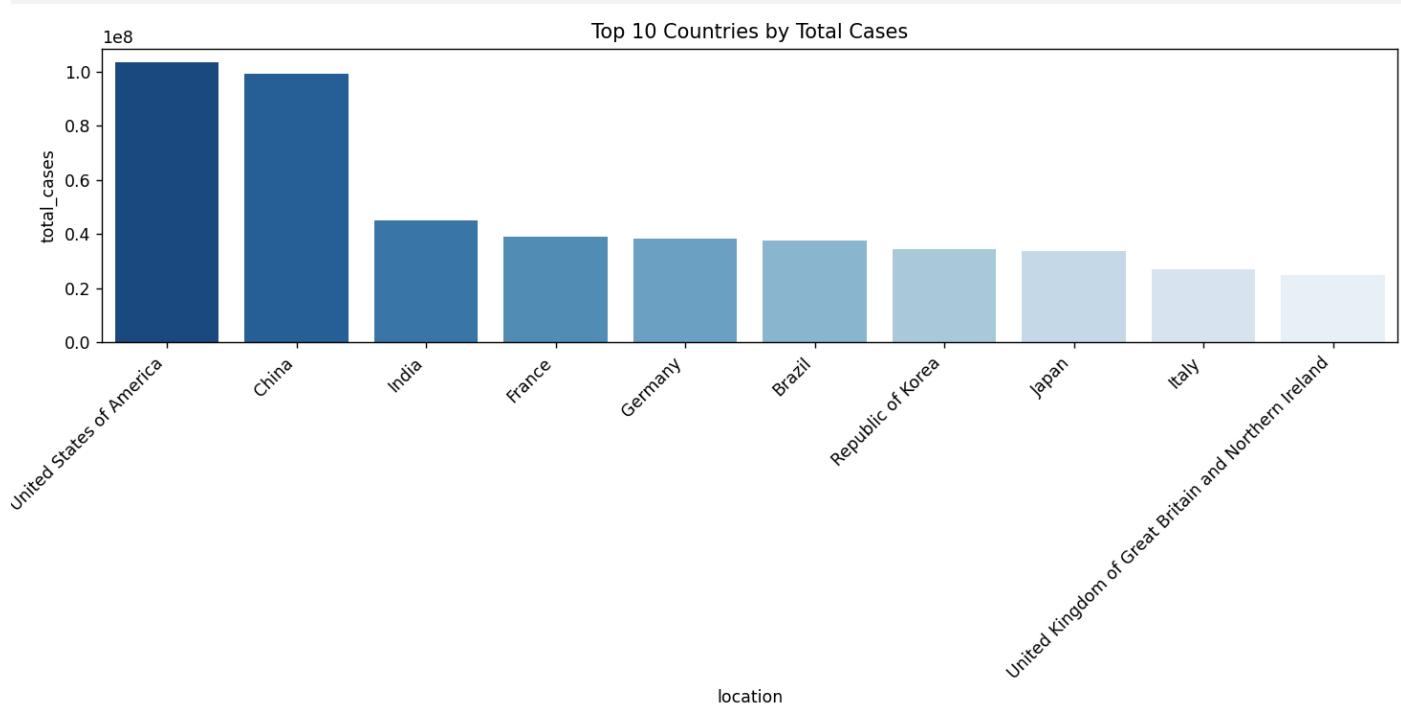
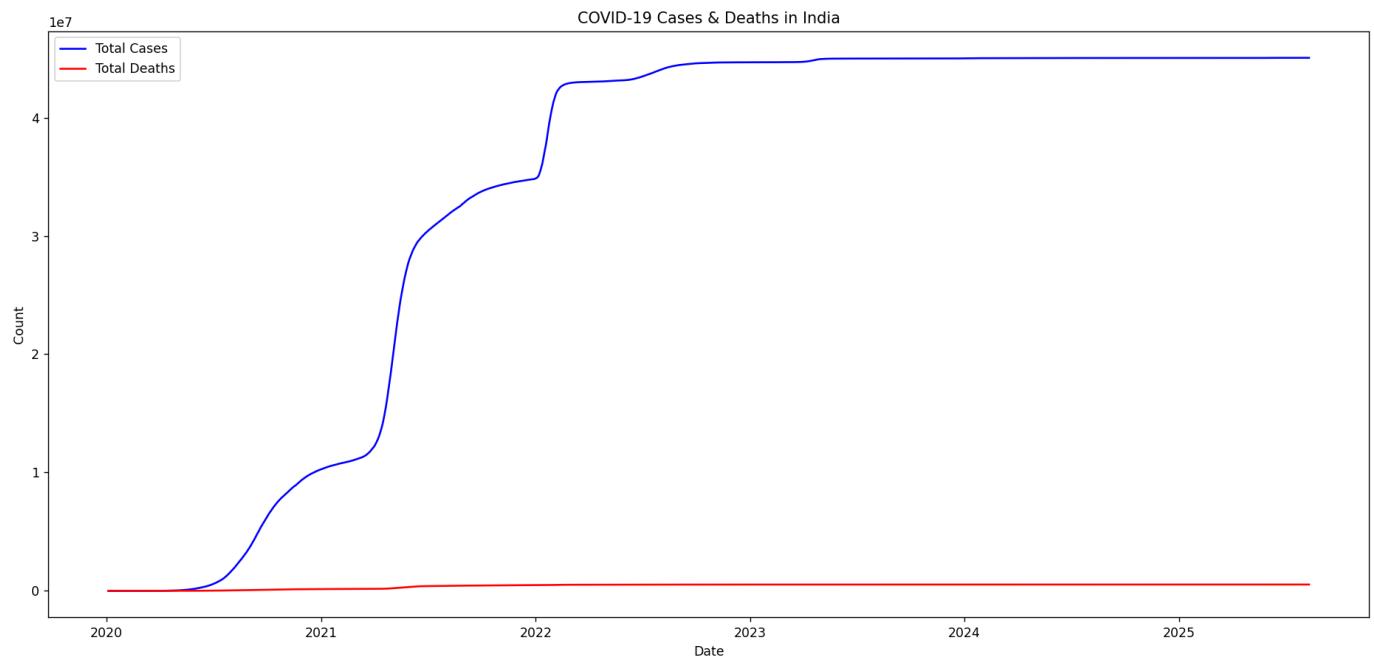
    fig.show()

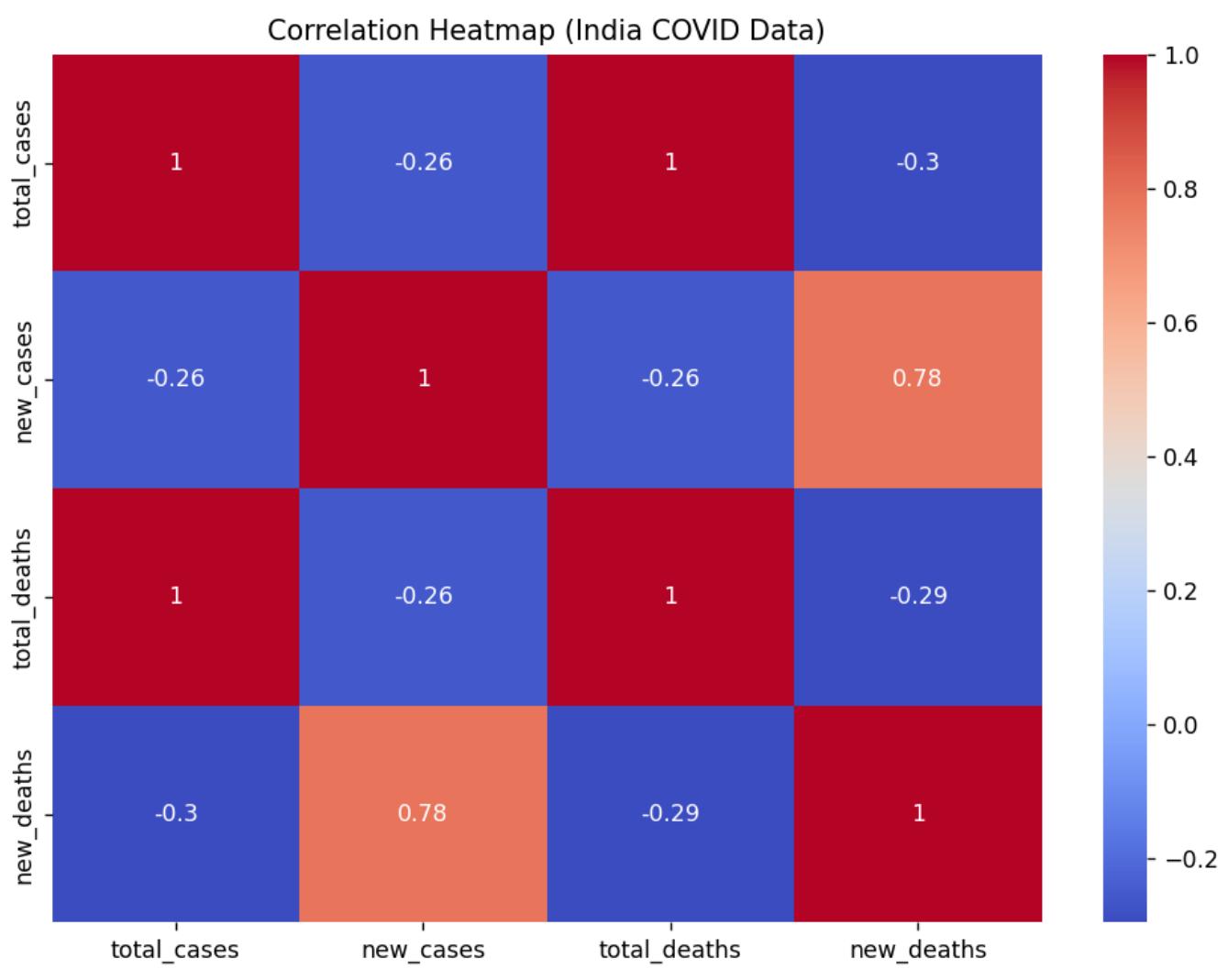
else:

    print('No latest data for choropleth.')

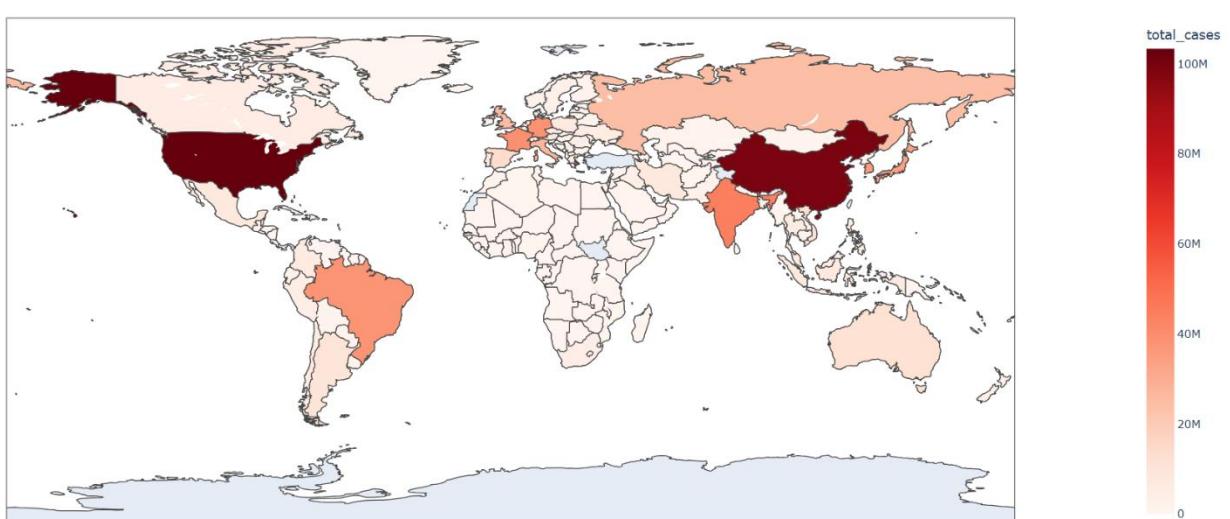
```

OUTPUTS:





Worldwide COVID-19 Cases



4 powerbi/covid_dashboard.pbix

This is the **Power BI dashboard file**.

Steps to Build in Power BI:

1. **Get Data** → **CSV** → Load WHO-COVID-Data.csv.
2. **Transform Data** (Power Query Editor):
 - o Convert Date_reported to Date.
 - o Remove unnecessary columns if needed.
3. **Build Visuals:**
 - o Line chart.
 - o Bar chart.
 - o Map.
 - o pie

