**COVID-19 CASE ANALYSIS**

**Data Collection:**

Obtain the COVID-19 case data from reliable sources, such as government health departments, the World Health Organization (WHO), or datasets provided by reputable organizations.

**Data Loading:**

Load the data into a data analysis tool like Python with Pandas or R.

**Data Inspection:**

Use Pandas to inspect the data, checking for the structure, data types, and the first few rows.

**Handling Missing Values:**

Identify and handle missing values, which may include removing rows with missing data or imputing values.

**Data Type Conversion:**

Ensure that data types are appropriate for analysis. Convert data types as needed.

**Data Cleaning:**

Clean the data by addressing inconsistencies, outliers, and errors. This might involve removing duplicates, correcting data entries, or filtering the dataset.

**Feature Engineering:**

Create new features or modify existing ones to extract valuable information for analysis.

**Data Aggregation:**

Aggregate data when necessary to summarize information over time, by location, or by other relevant factors.

**Data Transformation:**

Apply transformations, such as scaling, normalization, or log transformations, if required for the analysis.

**Data Visualization:**

Create data visualizations to explore the data and detect patterns or anomalies.

**Data Export:**

Save the pre processed data for analysis in a new file if needed.

**SOURCE CODE**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load your COVID-19 data into a pandas DataFrame

data = pd.read\_csv("/content/Covid\_19\_cases4.csv") # Replace with the path to your data file

# Convert the "dateRep" column to datetime

data['dateRep'] = pd.to\_datetime(data['dateRep'], format='%d-%m-%Y')

# Create a new column for daily cases

data['daily\_cases'] = data['cases'].diff()

# Create a new column for the 7-day rolling average of daily cases

data['rolling\_avg\_cases'] = data['daily\_cases'].rolling(window=7).mean()

# Set the date as the index for time series plots

data.set\_index('dateRep', inplace=True)

# Plot 2: Bar Plot of Total Cases and Deaths by Month

monthly\_data = data.resample('M').sum()

plt.figure(figsize=(10, 5))

monthly\_data[['cases', 'deaths']].plot(kind='bar', stacked=True)

plt.xlabel('Month')

plt.ylabel('Count')

plt.title('Total COVID-19 Cases and Deaths by Month in Austria')

plt.xticks(rotation=0)

plt.show()

# Plot 4: Box Plot of Daily Cases by Weekday

data['weekday'] = data.index.weekday

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

sns.boxplot(data=data, x='weekday', y='daily\_cases')

plt.xlabel('Weekday')

plt.ylabel('Daily Cases')

plt.title('Distribution of Daily COVID-19 Cases by Weekday')

plt.xticks(range(7), ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])

plt.show()

**OUTPUT**



