PROJECT TITLE

Analyzing COVID-19 Cases and Deaths Data using IBM Cognos

TEAM MEMBERS

YUVARAJ T	**2021115124**
ABINAYA S	**2021115004**
AARTHI R	**2021115001**
TAMIZHARASAN D	**2021115325**
HARISH M	**2021115326**

PHASE 3

Data Preprocessing Report

INTRODUCTION:

This report outlines the data preprocessing steps taken to prepare and analyze COVID-19 data using Jupyter Notebook(python). The analysis was conducted with the objective of understanding and visualizing daily COVID-19 cases. The dataset was loaded from a CSV file named 'covidcases.csv' and processed using Python libraries, primarily Pandas and Matplotlib.

DATA PRE-PROCESSING OBJECTIVES:

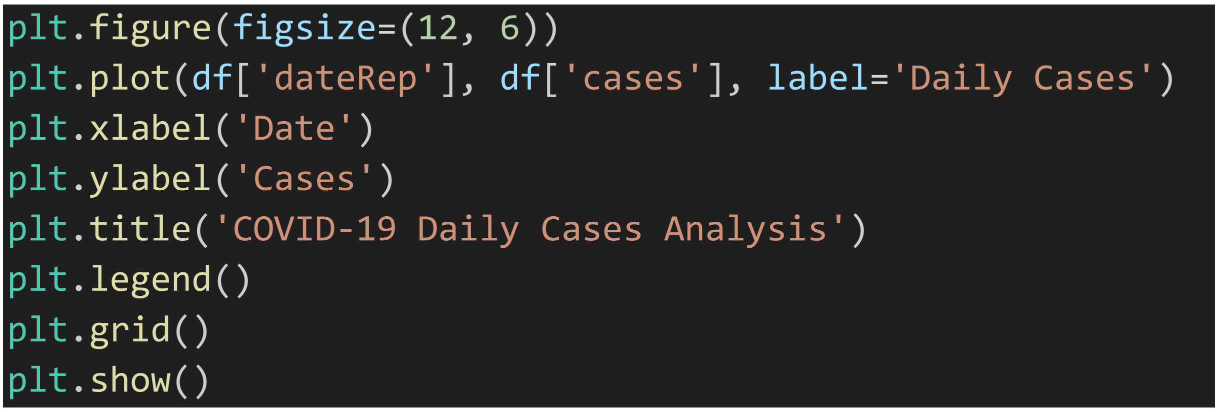
The main objectives of data preprocessing for this analysis were as follows:

1. **Data Loading**: The dataset was loaded from the 'covidcases.csv' file using the Pandas library. The dataset includes columns such as 'dateRep,' 'day,' 'month,' 'year,' 'cases,' 'deaths,' and 'countriesAndTerritories.'

2. **Data Cleaning**: Missing data were handled by filling in missing values with zeros, as we assumed that there were no missing values in the sample data. Additionally, we sorted the data by date to ensure that it was in chronological order.
3. **Data Conversion**: The 'dateRep' column was converted to a datetime object with the correct format ('%d-%m-%Y') for effective time series analysis.
4. **Data Visualization**: The pre-processed data was visualized using Matplotlib to achieve the objective of understanding and visualizing daily COVID-19 cases.

CODE FOR CLEANSING AND VISUALIZATION:

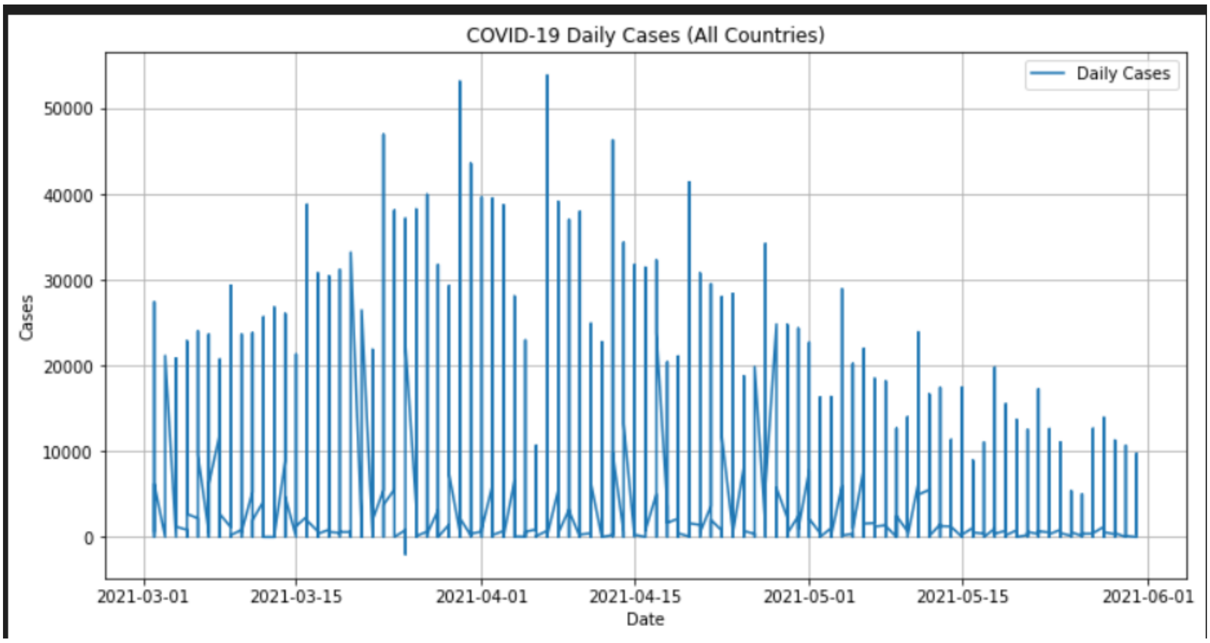
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data from the CSV file
df = pd.read_csv('covidcases.csv')

# Objective of the Analysis:
# Let's assume the objective is to analyze and visualize the daily COVID-19 cases.

# 1. Convert 'dateRep' to a datetime object
df['dateRep'] = pd.to_datetime(df['dateRep'], format='%d-%m-%Y')

# 2. Sort the data by date
df = df.sort_values(by='dateRep')

# 3. Handle missing data (if needed)
# For this example, we'll assume there is no missing data in the sample.

# 4. Data Visualization
```



```
plt.figure(figsize=(12, 6))
plt.plot(df['dateRep'], df['cases'], label='Daily Cases')
plt.xlabel('Date')
plt.ylabel('Cases')
plt.title('COVID-19 Daily Cases Analysis')
plt.legend()
plt.grid()
plt.show()
```

OUTPUT:

Visualized graph



DATA VISUALIZATION:

As part of this data preprocessing, we generated a line plot to visualize the daily COVID-19 cases over time. The plot displayed the number of daily cases on the y-axis and the corresponding dates on the x-axis. This visualization provided a clear overview of how the number of COVID-19 cases evolved over time.

CONCLUSION AND FUTURE STEPS:

The data preprocessing steps performed in this analysis successfully prepared the dataset for further analysis and visualization. However, to make the analysis more comprehensive we might consider adding the following in future steps:

1. **Feature Engineering**: Create additional features or metrics that could be valuable for deeper analysis. For example, you could calculate rolling averages, growth rates, or 7-day moving averages to identify trends and patterns.
2. **Geospatial Analysis**: Looking forward in incorporating geospatial analysis to visualize COVID-19 cases on maps, analyze regional variations, and identify hotspots.
3. **Statistical Analysis**: Perform statistical tests and analysis to identify significant trends, correlations, or anomalies in the data.
4. **Machine Learning**: Consider utilizing machine learning models for predictive analysis, time series forecasting, or clustering to gain more insights from the data.