PHASE 3:

Define Analysis Objectives: Clearly define the goals and objectives of your analysis. What specific insights or questions are you trying to address through this COVID-19 cases analysis? Understanding your objectives is crucial for structuring your analysis effectively.

Obtain COVID-19 Data: You'll need to acquire the COVID-19 cases and deaths data file. This data can often be obtained from reliable sources such as government health agencies, international organizations, or research institutions. Ensure you have the necessary permissions and rights to use this data for your analysis.

Data Processing and Cleaning: Preprocessing is a critical step in any data analysis project. Here's what it typically involves:

Data Loading: Import the COVID-19 data file into your analysis tool (in this case, IBM Cognos).

Data Cleaning: Inspect the data for missing values, outliers, and inconsistencies. Clean and transform the data to ensure its accuracy and reliability.

Data Validation: Check the data for errors and confirm that it aligns with your analysis objectives.

Data Integration: If necessary, combine the COVID-19 data with other relevant datasets for a more comprehensive analysis.

Data Aggregation: Summarize the data at the appropriate level (daily, weekly, by region, etc.) based on your analysis objectives.

After these initial steps, you can proceed to analyze and visualize the COVID-19 data using IBM Cognos, keeping your defined analysis objectives in mind. Visualization tools like IBM Cognos can help you create informative charts, graphs, and dashboards to communicate your findings effectively.

SAMPLE PROGRAM:

INPUT:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

data=pd.read\_csv(r"C:\Users\keccs\OneDrive\Documents\Covid\_19\_cases4.csv")

data.head()

OUTPUT:

dateRep day month year cases deaths countriesAndTerritories

0 31-05-2021 31 5 2021 366 5 Austria

1 30-05-2021 30 5 2021 570 6 Austria

2 29-05-2021 29 5 2021 538 11 Austria

3 28-05-2021 28 5 2021 639 4 Austria

4 27-05-2021 27 5 2021 405 19 Austria

INPUT:

data.info()

OUTPUT:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2730 entries, 0 to 2729

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 dateRep 2730 non-null object

1 day 2730 non-null int64

2 month 2730 non-null int64

3 year 2730 non-null int64

4 cases 2730 non-null int64

5 deaths 2730 non-null int64

6 countriesAndTerritories 2730 non-null object

dtypes: int64(5), object(2)

memory usage: 149.4+ KB

INPUT:

data.drop\_duplicates(inplace=True)

print(data.head())

OUTPUT:

dateRep day month year cases deaths countriesAndTerritories

0 31-05-2021 31 5 2021 366 5 Austria

1 30-05-2021 30 5 2021 570 6 Austria

2 29-05-2021 29 5 2021 538 11 Austria

3 28-05-2021 28 5 2021 639 4 Austria

4 27-05-2021 27 5 2021 405 19 Austria

INPUT:

data.describe()

OUTPUT:

day month year cases deaths

count 2730.0 2730.000000 2730.0 2730.000000 2730.000000

mean 16.0 4.010989 2021.0 3661.010989 65.291941

std 8.765919 0.818813 0.0 6490.510073 113.956634

min 1.0 3.000000 2021.0 -2001.000000 -3.000000

25% 8.0 3.000000 2021.0 361.250000 2.000000

50% 16.0 4.000000 2021.0 926.500000 14.500000

75% 24.0 5.000000 2021.0 3916.250000 72.000000

max 31.0 5.000000 2021.0 53843.000000 956.000000

INPUT:

countriesAndTerritories = data['countriesAndTerritories']

print(countriesAndTerritories)

OUTPUT:

0 Austria

1 Austria

2 Austria

3 Austria

4 Austria

...

2725 Sweden

2726 Sweden

2727 Sweden

2728 Sweden

2729 Sweden

Name: countriesAndTerritories, Length: 2730, dtype: object

Done by

ABINASH.S

ABINASH.P

DHANUSH.M

DINESH.K

​