**COVID 19 USING COGNOS ANALYTICS**

**1. Data Source:**

Dataset from “kaggle datasets download -d chakradharmattapalli/covid-19-cases”

**2. Data Preparation:**

**Data Cleaning:** Raw data from these sources might have inconsistencies. So we use tools like Python (Pandas library) or Excel to clean the data.

**Data Integration:** Combining data from different sources into a single dataset if necessary.

**Data Transformation:** Converting data into a format suitable for analysis. Ensure consistent date formats, handle missing values, etc.

**3. Setting up Cognos Analytics:**

**Connect to Data Source:** In Cognos Analytics, connecting to our prepared dataset. Cognos supports various data sources including databases, Excel files, and web services. Choosing the appropriate connection type.

**Data Import:** Importing our cleaned and prepared data into Cognos Analytics.

**4. Data Modeling:**

**Create Data Modules:** Organizing our data into logical modules. Defining relationships between tables if our data is relational.

**Create Calculations:** If needed,we can create calculated fields that can aid in our analysis.

**5. Report Creation:**

**Create Reports and Dashboards:** Using the drag-and-drop interface of Cognos to create reports and dashboards. we can visualize COVID-19 trends, infection rates, recovery rates, etc.

**Interactive Dashboards:** Creating interactive dashboards allowing users to filter data based on different parameters like date, region, etc.

**6. Data Analysis:**

**Perform Analysis:** Utilizing Cognos' features to analyze the data. This might include trend analysis, geographical analysis, or demographic analysis.

**Apply Advanced Analytics:** Depending on our Cognos version, there might be features for predictive analytics.

**7. Sharing and Collaboration:**

**Schedule and Share Reports:** Schedule reports to be automatically generated and shared with stakeholders.

**Collaboration:** Using Cognos features for collaboration, such as commenting on reports, to foster teamwork and insights.

**8. Security:**

**Set Security Policies:** Ensure that sensitive data is accessible only to authorized personne

SOURCE CODE:

import numpy as np

import pandas as pd

import plotly.express as px

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

/kaggle/input/latest-covid19-data-of-european-countries/Latest Covid-19 Data of European Countries.csv

df=pd.read\_csv("../input/latest-covid19-data-of-european-countries/Latest Covid-19 Data of European Countries.csv")

df.head()

OUTPUT:

| Country | Total Cases | Total Deaths | Total Recovered | Active Cases | Total Cases/1 mil population | Deaths/1 mil population | Total Tests | Tests/1 mil population | Population |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Albania | 188557 | 2950 | 177713 | 7894 | 65618 | 1027 | 1322841 | 460349 | 2873562 |
| 1 | Andorra | 15618 | 130 | 15289 | 199 | 201700 | 1679 | 193595 | 2500194 | 77432 |
| 2 | Austria | 875333 | 11482 | 795835 | 68016 | 96449 | 1265 | 99857207 | 11002797 | 9075620 |
| 3 | Belarus | 612015 | 4730 | 587208 | 20077 | 64797 | 501 | 9814116 | 1039066 | 9445134 |
| 4 | Belgium | 1414463 | 26131 | 1230086 | 158246 | 121334 | 2242 | 22225317 | 1906513 | 11657574 |
| 5 | Italy | 1456889 | 26789 | 1345678 | 973678 | 355687 | 9874 | 08766788 | 1389000 | 2133674 |

df.info()

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

RangeIndex: 43 entries, 0 to 42

Data columns (total 10 columns

# Column Non-Null Count Dtype

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

0 Country 43 non-null object

1 Total Cases 43 non-null int64

2 Total Deaths 43 non-null int64

3 Total Recovered 43 non-null int64

4 Active Cases 43 non-null int64

5 Total Cases/1 mil population 43 non-null int64

6 Deaths/1 mil population 43 non-null int64

7 Total Tests 43 non-null int64

8 Tests/1 mil population 43 non-null int64

9 Population 43 non-null int64

dtypes: int64(9), object(1)

memory usage: 3.5+ KB

df.isnull().sum()

Country 0

Total Cases 0

Total Deaths 0

Total Recovered 0

Active Cases 0

Total Cases/1 mil population 0

Deaths/1 mil population 0

Total Tests 0

Tests/1 mil population 0

Population 0

dtype: int64

df = df.rename(columns = {"Country/Other":"Country"})

df.columns

Index(['Country', 'Total Cases', 'Total Deaths', 'Total Recovered',

'Active Cases', 'Total Cases/1 mil population',

'Deaths/1 mil population ', 'Total Tests', 'Tests/1 mil population',

'Population'],

dtype='object')

df1 = df.sort\_values(by = ['Total Cases'],ascending = False).reset\_index().head(10)

fig = px.bar(df1, x = 'Country', y = 'Total Cases', color = 'Total Cases',

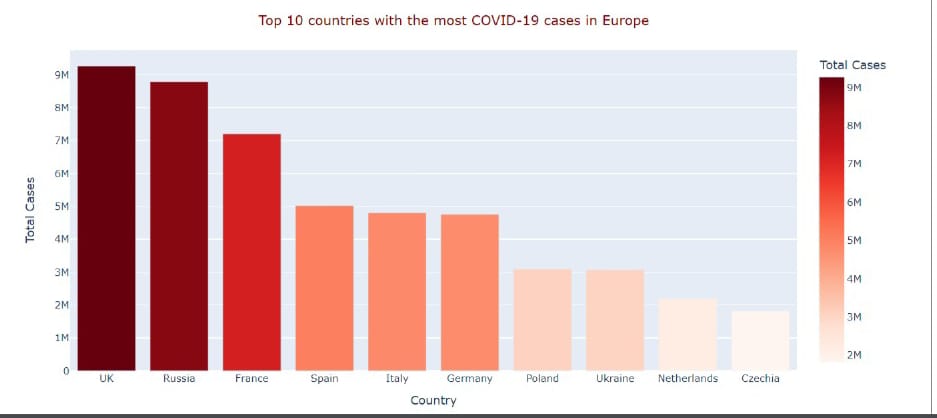
color\_continuous\_scale = 'reds')

fig.update\_layout(title = 'Top 10 countries with the most COVID-19 cases in Europe',

title\_x = 0.5,

title\_font = dict(size = 16, color = 'DarkRed'))

fig.show()



df1 = df.sort\_values(by = ['Total Deaths'],ascending = False).reset\_index().head(10)

fig = px.bar(df1, x= 'Country',

y ='Total Deaths',

color = 'Total Deaths',

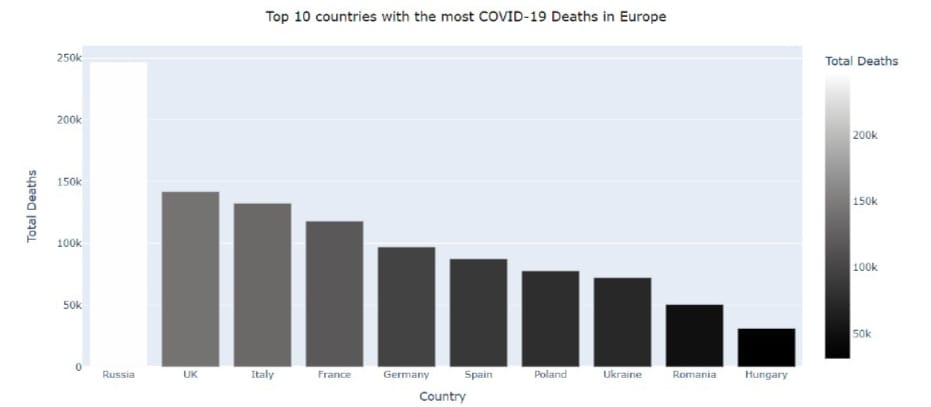
color\_continuous\_scale = 'gray')

fig.update\_layout(title = 'Top 10 countries with the most COVID-19 Deaths in Europe',

title\_x = 0.5,

title\_font = dict(size = 16, color = 'Black'))

fig.show()



df1 = df.sort\_values(by = ['Active Cases'],ascending = False).reset\_index().head(10)

fig = px.bar(df1,

x = 'Country',

y = 'Active Cases',

color = 'Active Cases',

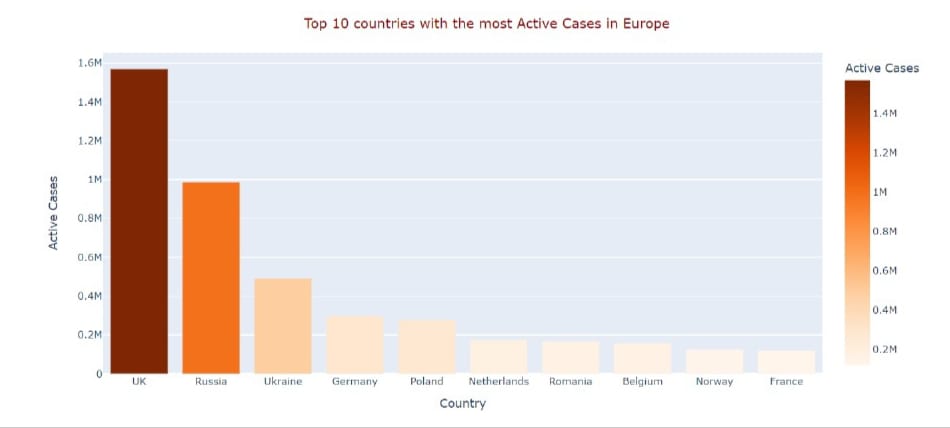
color\_continuous\_scale = 'oranges')

fig.update\_layout(title = 'Top 10 countries with the most Active Cases in Europe',

title\_x = 0.5,

title\_font = dict(size = 16,color = 'Darkred'))

fig.show()



df1 = df.sort\_values(by = ['Total Cases/1 mil population'],ascending = False).reset\_index().head(20)

fig = px.scatter(df1,

x = 'Total Cases/1 mil population',

y = 'Deaths/1 mil population ',

color ='Country',

color\_discrete\_sequence = px.colors.qualitative.Alphabet,

labels = {"Deaths/1 mil population ":"Total Deaths/per 1 mil population"})

fig.update\_traces(marker = dict(size = 30,

line = dict(color = 'lightgray',width = 1.5)),

mode = 'markers')

fig.update\_layout(title = 'Top 20 countries: Total Cases v.s Total Deaths/per 1 mil population',

title\_x = 0.5,

title\_font = dict(size = 16, color = 'Black'),

yaxis = dict(title = 'Total Deaths/per 1 mil population'))

fig.show()

