DATA ANALYTICS WITH COGNOS

COVID-19 CASES ANALYSIS

PHASE 3

INTRODUCTION

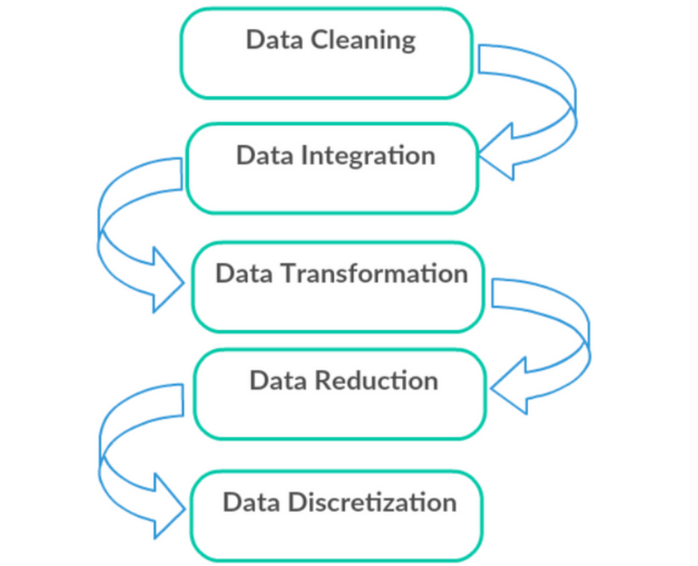
The project involves analyzing COVID-19 cases and deaths data using IBM Cognos. The objective is to compare and contrast the mean values and standard deviations of cases and associated deaths per day and by country in the EU/EEA. This project encompasses defining analysis objectives, collecting COVID-19 data, designing relevant visualizations in IBM Cognos, and deriving insights from the data.

DATASET:

https://www.kaggle.com/datasets/chakradharmattapalli/covid-19-cases

DATA PREPROCESSING AND DATA COLLECTION

Data preprocessing is a crucial step in any data analysis project. Here are the general steps you might follow for the COVID-19 cases analysis:



Load the Data:

First, you need to load the data into a suitable data structure. If you're using Python, pandas DataFrame is a good option for this.

CODE:

```
import pandas as pd
data=pd.read_csv('covid19_data.csv')
```

Inspect the Data:

Take a look at the data to understand its structure and contents. Check the number of rows and columns, column names, data types, etc.

CODE:

```
data.info()
```

Handle Missing Values:

Check for missing or null values in the data. You might fill them with appropriate values (like mean or median), or drop the rows/columns containing them, based on the situation.

CODE:

```
data.isnull().sum()
```

MANIPULATION OF DATA

Data manipulation refers to the process of adjusting data to make it organised and easier to read. Data manipulation language, or DML, is a programming language that adjusts data by inserting, deleting and modifying data in a database such as to cleanse or map the data

CODING

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
covid=pd.read_csv("Covid_19_cases4.csv")
covid

covid.describe()

covid.info()

x=covid.drop("deaths",axis=1)
y=covid['deaths']
```

CODING

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
5,random_state=0)
x_train

x_train.shape

x_test.shape

y_train.shape

y_test.shape

y_test

x_test

x

display(covid.drop_duplicates())

plt.plot(covid.cases,covid.deaths)
plt.xlabel('covid.cases')
plt.ylabel('covid.deaths')
plt.title('covid graph')
plt.show()
```

OUTPUT

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
covid=pd.read_csv("Covid_19_cases4.csv")
covid
```

	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
...	...	...	...	...	...	...	...
2725	06-03-2021	6	3	2021	3455	17	Sweden
2726	05-03-2021	5	3	2021	4069	12	Sweden
2727	04-03-2021	4	3	2021	4884	14	Sweden
2728	03-03-2021	3	3	2021	4876	19	Sweden
2729	02-03-2021	2	3	2021	6191	19	Sweden

2730 rows × 7 columns

```
covid.info()
```

```
<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
```

OUTPUT

```
covid.describe()
```

	day	month	year	cases	deaths
count	2730.000000	2730.000000	2730.0	2730.000000	2730.000000
mean	16.000000	4.010989	2021.0	3661.010989	65.291941
std	8.765919	0.818813	0.0	6490.510073	113.956634
min	1.000000	3.000000	2021.0	-2001.000000	-3.000000
25%	8.000000	3.000000	2021.0	361.250000	2.000000
50%	16.000000	4.000000	2021.0	926.500000	14.500000
75%	24.000000	5.000000	2021.0	3916.250000	72.000000
max	31.000000	5.000000	2021.0	53843.000000	956.000000

```
x=covid.drop("deaths",axis=1)
y=covid['deaths']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5,randc
x_train

x_train.shape

x_test.shape

y_train.shape

y_test.shape

y_test

x_test
```

	dateRep	day	month	year	cases	countriesAndTerritories
578	29-04-2021	29	4	2021	835	Denmark
2242	03-04-2021	3	4	2021	548	Portugal
2690	10-04-2021	10	4	2021	5628	Sweden
70	22-03-2021	22	3	2021	2918	Austria
758	01-05-2021	1	5	2021	195	Finland
...	...	...	...	...	...	...
1549	29-05-2021	29	5	2021	2	Liechtenstein
308	26-04-2021	26	4	2021	1254	Croatia
1001	31-05-2021	31	5	2021	906	Greece
1651	18-05-2021	18	5	2021	527	Lithuania
143	09-04-2021	9	4	2021	4480	Belgium

1365 rows × 6 columns

CONCLUSION

COVID-19 Cases Analysis insights aid decision-makers in understanding current scenarios, predicting future trends, and making informed choices. These insights guide healthcare professionals in allocating resources and implementing them.