importing pandas

In [1]:

```
import pandas as pd
import numpy as np
```

reading the csv file

In [52]:

```
df = pd.read_csv('country_wise_latest.csv')
```

In [53]:

```
df
```

Out[53]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	
4	Angola	950	41	242	667	18	1	0	4.32	25.47	1
...	...	...	...	...	...	...	...	...	...	...	
182	West Bank and Gaza	10621	78	3752	6791	152	2	0	0.73	35.33	
183	Western Sahara	10	1	8	1	0	0	0	10.00	80.00	1
184	Yemen	1691	483	833	375	10	4	36	28.56	49.26	5
185	Zambia	4552	140	2815	1597	71	1	465	3.08	61.84	
186	Zimbabwe	2704	36	542	2126	192	2	24	1.33	20.04	

187 rows × 15 columns

In [ ]:

converting the zero values to null values in the data set

here in the data set we are having the zeros in place of null so im converting the zeros in the table to null values by using the below function so here it goes

In [5]:

```
df = df.applymap(lambda x: None if x == 0 else x)
```

In [6]:

```
df
```

Out[6]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
0	Afghanistan	36263	1269.0	25198.0	9796.0	106.0	10.0	18.0	3.50	69.49	
1	Albania	4880	144.0	2745.0	1991.0	117.0	6.0	63.0	2.95	56.25	
2	Algeria	27973	1163.0	18837.0	7973.0	616.0	8.0	749.0	4.16	67.34	
3	Andorra	907	52.0	803.0	52.0	10.0	NaN	NaN	5.73	88.53	
4	Angola	950	41.0	242.0	667.0	18.0	1.0	NaN	4.32	25.47	1
...	...	...	...	...	...	...	...	...	...	...	
182	West Bank and Gaza	10621	78.0	3752.0	6791.0	152.0	2.0	NaN	0.73	35.33	
183	Western Sahara	10	1.0	8.0	1.0	NaN	NaN	NaN	10.00	80.00	1
184	Yemen	1691	483.0	833.0	375.0	10.0	4.0	36.0	28.56	49.26	5
185	Zambia	4552	140.0	2815.0	1597.0	71.0	1.0	465.0	3.08	61.84	
186	Zimbabwe	2704	36.0	542.0	2126.0	192.0	2.0	24.0	1.33	20.04	

187 rows × 15 columns

the convertion of the zero values into null values is successfull

removing the rows containing null values

here i used the dropna() function to remove the rows that contain null values in the data set

In [7]:

```
df.dropna()
```

Out[7]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	D Rec
0	Afghanistan	36263	1269.0	25198.0	9796.0	106.0	10.0	18.0	3.50	69.49	
1	Albania	4880	144.0	2745.0	1991.0	117.0	6.0	63.0	2.95	56.25	
2	Algeria	27973	1163.0	18837.0	7973.0	616.0	8.0	749.0	4.16	67.34	
6	Argentina	167416	3059.0	72575.0	91782.0	4890.0	120.0	2057.0	1.83	43.35	
7	Armenia	37390	711.0	26665.0	10014.0	73.0	6.0	187.0	1.90	71.32	
...	...	...	...	...	...	...	...	...	...	...	
179	Uzbekistan	21209	121.0	11674.0	9414.0	678.0	5.0	569.0	0.57	55.04	
180	Venezuela	15988	146.0	9959.0	5883.0	525.0	4.0	213.0	0.91	62.29	
184	Yemen	1691	483.0	833.0	375.0	10.0	4.0	36.0	28.56	49.26	
185	Zambia	4552	140.0	2815.0	1597.0	71.0	1.0	465.0	3.08	61.84	
186	Zimbabwe	2704	36.0	542.0	2126.0	192.0	2.0	24.0	1.33	20.04	

84 rows × 15 columns

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

replacing the null values with 100

by removing the rows we lost the data more than 100 rows so the previouos technique is failed,so using another technique is recommended...

and im gonna replace all the null values with 100

In [8]:

```
df
```

Out[8]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
0	Afghanistan	36263	1269.0	25198.0	9796.0	106.0	10.0	18.0	3.50	69.49	
1	Albania	4880	144.0	2745.0	1991.0	117.0	6.0	63.0	2.95	56.25	
2	Algeria	27973	1163.0	18837.0	7973.0	616.0	8.0	749.0	4.16	67.34	
3	Andorra	907	52.0	803.0	52.0	10.0	NaN	NaN	5.73	88.53	
4	Angola	950	41.0	242.0	667.0	18.0	1.0	NaN	4.32	25.47	1
...	...	...	...	...	...	...	...	...	...	...	
182	West Bank and Gaza	10621	78.0	3752.0	6791.0	152.0	2.0	NaN	0.73	35.33	
183	Western Sahara	10	1.0	8.0	1.0	NaN	NaN	NaN	10.00	80.00	1
184	Yemen	1691	483.0	833.0	375.0	10.0	4.0	36.0	28.56	49.26	5
185	Zambia	4552	140.0	2815.0	1597.0	71.0	1.0	465.0	3.08	61.84	
186	Zimbabwe	2704	36.0	542.0	2126.0	192.0	2.0	24.0	1.33	20.04	

187 rows × 15 columns

In [62]:

```
df['Confirmed']                =df['Confirmed'].fillna(100)
df['Deaths']                   =df['Deaths'].fillna(100)
df['Recovered']                =df['Recovered'].fillna(100)
df['Active']                   =df['Active'].fillna(100)
df['New cases']                =df['New cases'].fillna(100)
df['New deaths']               =df['New deaths'].fillna(100)
df['New recovered']            =df['New recovered'].fillna(100)
df['Deaths / 100 Cases']       =df['Deaths / 100 Cases'].fillna(100)
df['Recovered / 100 Cases']    =df['Recovered / 100 Cases'].fillna(100)
df['Deaths / 100 Recovered']   =df['Deaths / 100 Recovered'].fillna(100)
df['Confirmed last week']      =df['Confirmed last week'].fillna(100)
df['1 week change']            =df['1 week change'].fillna(100)
df['1 week % increase']        =df['1 week % increase'].fillna(100)
```

In [10]:

```
df
```

Out[10]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
0	Afghanistan	36263	1269.0	25198.0	9796.0	106.0	10.0	18.0	3.50	69.49	
				2745.0	1991.0	117.0	6.0	63.0	2.95	56.25	

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
2	Algeria	27973	1163.0	18837.0	7973.0	616.0	8.0	749.0	4.16	67.34	
3	Andorra	907	52.0	803.0	52.0	10.0	100.0	100.0	5.73	88.53	
4	Angola	950	41.0	242.0	667.0	18.0	1.0	100.0	4.32	25.47	1
...	...	...	...	...	...	...	...	...	...	...	
182	West Bank and Gaza	10621	78.0	3752.0	6791.0	152.0	2.0	100.0	0.73	35.33	
183	Western Sahara	10	1.0	8.0	1.0	100.0	100.0	100.0	10.00	80.00	1
184	Yemen	1691	483.0	833.0	375.0	10.0	4.0	36.0	28.56	49.26	5
185	Zambia	4552	140.0	2815.0	1597.0	71.0	1.0	465.0	3.08	61.84	
186	Zimbabwe	2704	36.0	542.0	2126.0	192.0	2.0	24.0	1.33	20.04	

187 rows × 15 columns

using imputers

In [16]:

```
!pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\menak\anaconda3\lib\site-packages (0.0.
post7)
```

In [22]:

```
from sklearn.impute import SimpleImputer
```

In [43]:

```
imputer
```

Out[43]:

```
SimpleImputer(strategy='most_frequent')
```

In [24]:

```
df = pd.read_csv('country_wise_latest.csv')
df = df.applymap(lambda x: None if x == 0 else x)
```

In [ ]:

```

```

In [57]:

```
imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
```

In [25]:

```
df
```

Out[25]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
0	Afghanistan	36263	1269.0	25198.0	9796.0	106.0	10.0	18.0	3.50	69.49	
1	Albania	4880	144.0	2745.0	1991.0	117.0	6.0	63.0	2.95	56.25	
2	Algeria	27973	1163.0	18837.0	7973.0	616.0	8.0	749.0	4.16	67.34	
				803.0	52.0	10.0	NaN	NaN	5.73	88.53	

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
4	Angola	950	41.0	242.0	667.0	18.0	1.0	NaN	4.32	25.47	1
...	...	...	...	...	...	...	...	...	...	...	
182	West Bank and Gaza	10621	78.0	3752.0	6791.0	152.0	2.0	NaN	0.73	35.33	
183	Western Sahara	10	1.0	8.0	1.0	NaN	NaN	NaN	10.00	80.00	1
184	Yemen	1691	483.0	833.0	375.0	10.0	4.0	36.0	28.56	49.26	5
185	Zambia	4552	140.0	2815.0	1597.0	71.0	1.0	465.0	3.08	61.84	
186	Zimbabwe	2704	36.0	542.0	2126.0	192.0	2.0	24.0	1.33	20.04	

187 rows × 15 columns

In [34]:

```
Confirmedlastweek = np.reshape(df['Confirmed last week'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Confirmedlastweek)
df['Confirmed last week']=imputer.transform(Confirmedlastweek)

Deaths = np.reshape(df['Deaths'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Deaths)
df['Deaths']=imputer.transform(Deaths)

Recovered = np.reshape(df['Recovered'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Recovered)
df['Recovered']=imputer.transform(Recovered)

Active = np.reshape(df['Active'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Active)
df['Active']=imputer.transform(Active)

Newcases = np.reshape(df['New cases'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Newcases)
df['New cases']=imputer.transform(Newcases)

Newdeaths = np.reshape(df['New deaths'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Newdeaths)
df['New deaths']=imputer.transform(Newdeaths)


Newrecovered = np.reshape(df['New recovered'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Newrecovered)
df['New recovered']=imputer.transform(Newrecovered)

Deaths_per_100_Cases = np.reshape(df['Deaths / 100 Cases'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
imputer.fit(Deaths_per_100_Cases)
df['Deaths / 100 Cases']=imputer.transform(Deaths_per_100_Cases)

Recovered_per_100_Cases = np.reshape(df['Recovered / 100 Cases'].values,(187,1))
imputer = SimpleImputer(missing_values = np.nan,strategy = 'most_frequent')
```

```
imputer.fit(Recovered_per_100_Cases)
df['Recovered / 100 Cases']=imputer.transform(Recovered_per_100_Cases)
```

In [46]:

```
df
```

Out[46]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Dea Recov
0	Afghanistan	36263	1269.0	25198.0	9796.0	106.0	10.0	18.0	3.50	69.49	
1	Albania	4880	144.0	2745.0	1991.0	117.0	6.0	63.0	2.95	56.25	
2	Algeria	27973	1163.0	18837.0	7973.0	616.0	8.0	749.0	4.16	67.34	
3	Andorra	907	52.0	803.0	52.0	10.0	1.0	2.0	5.73	88.53	
4	Angola	950	41.0	242.0	667.0	18.0	1.0	2.0	4.32	25.47	1
...	...	...	...	...	...	...	...	...	...	...	
182	West Bank and Gaza	10621	78.0	3752.0	6791.0	152.0	2.0	2.0	0.73	35.33	
183	Western Sahara	10	1.0	8.0	1.0	1.0	1.0	2.0	10.00	80.00	1
184	Yemen	1691	483.0	833.0	375.0	10.0	4.0	36.0	28.56	49.26	5
185	Zambia	4552	140.0	2815.0	1597.0	71.0	1.0	465.0	3.08	61.84	
186	Zimbabwe	2704	36.0	542.0	2126.0	192.0	2.0	24.0	1.33	20.04	

187 rows × 15 columns

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js