DATA EXPLORATION USING PANDAS

In [3]:

```
import pandas as pd
```

For Data Exploration pandas library is imported as pd.

In [4]:

```
covid=pd.read_csv('C://Users//admin//Downloads//covid.csv')
```

Daily record of covid in the world(2019) .CSV reads dataset from the location.

In [5]:

```
covid
```

Out[5]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
0	22-01-2020	555	17	28	510	0	0	0	3.06	5.05	60.71	6
1	23-01-2020	654	18	30	606	99	1	2	2.75	4.59	60.00	8
2	24-01-2020	941	26	36	879	287	8	6	2.76	3.83	72.22	9
3	25-01-2020	1434	42	39	1353	493	16	3	2.93	2.72	107.69	11
4	26-01-2020	2118	56	52	2010	684	14	13	2.64	2.46	107.69	13
...	...	...	...	...	...	...	...	...	...	...	...	...
183	23-07-2020	15510481	633506	8710969	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	8939705	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	9158743	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	9293464	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	9468087	6358362	228693	5415	174623	3.97	57.45	6.91	187

188 rows × 12 columns

The dataset which is downloaded from kaggle.com

In [6]:

```
covid.head()
```

Out[6]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
0	22-01-2020	555	17	28	510	0	0	0	3.06	5.05	60.71	6
1	23-01-2020	654	18	30	606	99	1	2	2.75	4.59	60.00	8
2	24-01-2020	941	26	36	879	287	8	6	2.76	3.83	72.22	9
3	25-01-2020	1434	42	39	1353	493	16	3	2.93	2.72	107.69	11
4	26-01-2020	2118	56	52	2010	684	14	13	2.64	2.46	107.69	13

The head function shows first 5 observations from the dataset.

In [7]:

```
covid.tail()
```

Out[7]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
183	23-07-2020	15510481	633506	8710969	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	8939705	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	9158743	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	9293464	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	9468087	6358362	228693	5415	174623	3.97	57.45	6.91	187

The tail function shows last 5 observations from the dataset.

In [8]:

```
covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188 entries, 0 to 187
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Date                   188 non-null    object 
 1   Confirmed              188 non-null    int64  
 2   Deaths                 188 non-null    int64  
 3   Recovered              188 non-null    int64  
 4   Active                 188 non-null    int64  
 5   New cases              188 non-null    int64  
 6   New deaths             188 non-null    int64  
 7   New recovered          188 non-null    int64  
 8   Deaths / 100 Cases     188 non-null    float64
 9   Recovered / 100 Cases  188 non-null    float64
 10  Deaths / 100 Recovered  188 non-null    float64
 11  No. of countries       188 non-null    int64  
dtypes: float64(3), int64(8), object(1)
memory usage: 17.8+ KB
```

The info function shows the information about the dataframe, which includes data types,range index, memory usage etc.

In [9]:

```
covid.describe()
```

Out[9]:

	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths 10 Recovere
count	1.880000e+02	188.000000	1.880000e+02	1.880000e+02	188.000000	188.000000	188.000000	188.000000	188.000000	188.00000
mean	4.406960e+06	230770.760638	2.066001e+06	2.110188e+06	87771.021277	3478.824468	50362.015957	4.860638	34.343936	22.10452
std	4.757988e+06	217929.094183	2.627976e+06	1.969670e+06	75295.293255	2537.735652	56090.892479	1.579541	16.206159	22.56830
min	5.550000e+02	17.000000	2.800000e+01	5.100000e+02	0.000000	0.000000	0.000000	2.040000	1.710000	6.26000
25%	1.121910e+05	3935.000000	6.044125e+04	5.864175e+04	5568.500000	250.750000	2488.250000	3.510000	22.785000	9.65000
50%	2.848733e+06	204190.000000	7.847840e+05	1.859759e+06	81114.000000	4116.000000	30991.500000	4.850000	35.680000	15.38000
75%	7.422046e+06	418634.500000	3.416396e+06	3.587015e+06	131502.500000	5346.000000	79706.250000	6.297500	48.945000	25.34250
max	1.648048e+07	654036.000000	9.468087e+06	6.358362e+06	282756.000000	9966.000000	284394.000000	7.180000	57.450000	134.43000

The describe function shows statistical values like count,mean,standard deviation,minimum and maximum values from dataframe.

In [10]:

```
covid.shape
```

Out[10]:

```
(188, 12)
```

The shape function calculates observations and variables in dataframe.

In [12]:

```
covid.columns
```

Out[12]:

```
Index(['Date', 'Confirmed', 'Deaths', 'Recovered', 'Active', 'New cases',
       'New deaths', 'New recovered', 'Deaths / 100 Cases',
       'Recovered / 100 Cases', 'Deaths / 100 Recovered', 'No. of countries'],
      dtype='object')
```

The column function shows all variables names in data frame

In [13]:

```
covid.index
```

Out[13]:

```
RangeIndex(start=0, stop=188, step=1)
```

The index function shows start , stop ,step of the observations in dataset.

In [16]:

```
covid.sort_values(['Deaths'],ascending=[True])
```

Out[16]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
0	22-01-2020	555	17	28	510	0	0	0	3.06	5.05	60.71	6
1	23-01-2020	654	18	30	606	99	1	2	2.75	4.59	60.00	8
2	24-01-2020	941	26	36	879	287	8	6	2.76	3.83	72.22	9
3	25-01-2020	1434	42	39	1353	493	16	3	2.93	2.72	107.69	11
4	26-01-2020	2118	56	52	2010	684	14	13	2.64	2.46	107.69	13
...	...	...	...	...	...	...	...	...	...	...	...	...
183	23-07-2020	15510481	633506	8710969	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	8939705	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	9158743	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	9293464	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	9468087	6358362	228693	5415	174623	3.97	57.45	6.91	187

188 rows × 12 columns

This sort function sorts the dataset in ascending by default. it also calculate decending order by using false in dataset

In [17]:

```
covid['Confirmed'].value_counts()
```

Out[17]:

```
555         1
6077978     1
5110064     1
5216964     1
5322253     1
           ..
538666      1
603066      1
670723      1
730300      1
16480485    1
Name: Confirmed, Length: 188, dtype: int64
```

The count function shows no of object appears in a covid confirmed list.

In [18]:

```
covid['Deaths'].value_counts()
```

Out[18]:

```
17        1
370718    1
334112    1
339396    1
343385    1
         ..
24800     1
28318     1
31997     1
35470     1
654036    1
Name: Deaths, Length: 188, dtype: int64
```

The count function shows no of object appears in a covid death list.

In [15]:

```
covid['Recovered'].value_counts()
```

Out[15]:

```
28         1
2509981    1
1900768    1
2008541    1
2062802    1
          ..
119804     1
128508     1
136800     1
146261     1
9468087    1
Name: Recovered, Length: 188, dtype: int64
```

The count function shows no of object appears in a covid recovered list.

In [16]:

```
covid['Active'].value_counts()
```

Out[16]:

```
510        1
3197279    1
2875184    1
2869027    1
2916066    1
          ..
394062     1
446240     1
501926     1
548569     1
6358362    1
Name: Active, Length: 188, dtype: int64
```

The count function shows no of object appears in a covid Active list.

In [19]:

```
covid.drop_duplicates(inplace=True)
```

In [20]:

```
covid
```

Out[20]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
0	22-01-2020	555	17	28	510	0	0	0	3.06	5.05	60.71	6
1	23-01-2020	654	18	30	606	99	1	2	2.75	4.59	60.00	8
2	24-01-2020	941	26	36	879	287	8	6	2.76	3.83	72.22	9
3	25-01-2020	1434	42	39	1353	493	16	3	2.93	2.72	107.69	11
4	26-01-2020	2118	56	52	2010	684	14	13	2.64	2.46	107.69	13
...	...	...	...	...	...	...	...	...	...	...	...	...
183	23-07-2020	15510481	633506	8710969	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	8939705	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	9158743	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	9293464	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	9468087	6358362	228693	5415	174623	3.97	57.45	6.91	187

188 rows × 12 columns

The drop dulicates function removes the dulicates from dataset.

In [21]:

```
covid1_subset=covid[['Confirmed','Deaths','Recovered','Active']]
```

In [22]:

```
covid1_subset
```

Out[22]:

	Confirmed	Deaths	Recovered	Active
0	555	17	28	510
1	654	18	30	606
2	941	26	36	879
3	1434	42	39	1353
4	2118	56	52	2010
...	...	...	...	...
183	15510481	633506	8710969	6166006
184	15791645	639650	8939705	6212290
185	16047190	644517	9158743	6243930
186	16251796	648621	9293464	6309711
187	16480485	654036	9468087	6358362

188 rows × 4 columns

The subset function enables us to form a subset of a dataset according to a specific observations or variables or both in dataset.

In [23]:

```
covid_row=covid[covid['Deaths']>600000]
```

In [36]:

```
covid_row
```

Out[36]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
178	18-07-2020	14292198	602130	7944550	5745518	237635	5627	150790	4.21	55.59	7.58	187
179	19-07-2020	14506845	606159	8032235	5868451	214647	4029	87685	4.18	55.37	7.55	187
180	20-07-2020	14713623	610319	8190777	5912527	206778	4160	158542	4.15	55.67	7.45	187
181	21-07-2020	14947078	616557	8364986	5965535	233565	6238	174209	4.12	55.96	7.37	187
182	22-07-2020	15227725	623540	8541255	6062930	280647	6983	176269	4.09	56.09	7.30	187
183	23-07-2020	15510481	633506	8710969	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	8939705	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	9158743	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	9293464	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	9468087	6358362	228693	5415	174623	3.97	57.45	6.91	187

The row function helps to filter the data from dataset.

In [37]:

```
covid['Recovered']= covid['Confirmed']-covid['Deaths']
```

In [38]:

```
covid
```

Out[38]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
0	22-01-2020	555	17	538	510	0	0	0	3.06	5.05	60.71	6
1	23-01-2020	654	18	636	606	99	1	2	2.75	4.59	60.00	8
2	24-01-2020	941	26	915	879	287	8	6	2.76	3.83	72.22	9
3	25-01-2020	1434	42	1392	1353	493	16	3	2.93	2.72	107.69	11
4	26-01-2020	2118	56	2062	2010	684	14	13	2.64	2.46	107.69	13
...	...	...	...	...	...	...	...	...	...	...	...	...
183	23-07-2020	15510481	633506	14876975	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	15151995	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	15402673	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	15603175	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	15826449	6358362	228693	5415	174623	3.97	57.45	6.91	187

188 rows × 12 columns

The covid recovered data set is formed by minusing covid deaths from covid confirmed

In [27]:

```
covid_index=covid.set_index('No. of countries')
```

In [28]:

```
covid_index
```

Out[28]:

No. of countries	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered
6	22-01-2020	555	17	538	510	0	0	0	3.06	5.05	60.71
8	23-01-2020	654	18	636	606	99	1	2	2.75	4.59	60.00
9	24-01-2020	941	26	915	879	287	8	6	2.76	3.83	72.22
11	25-01-2020	1434	42	1392	1353	493	16	3	2.93	2.72	107.69
13	26-01-2020	2118	56	2062	2010	684	14	13	2.64	2.46	107.69
...	...	...	...	...	...	...	...	...	...	...	...
187	23-07-2020	15510481	633506	14876975	6166006	282756	9966	169714	4.08	56.16	7.27
187	24-07-2020	15791645	639650	15151995	6212290	281164	6144	228736	4.05	56.61	7.16
187	25-07-2020	16047190	644517	15402673	6243930	255545	4867	219038	4.02	57.07	7.04
187	26-07-2020	16251796	648621	15603175	6309711	204606	4104	134721	3.99	57.18	6.98
187	27-07-2020	16480485	654036	15826449	6358362	228693	5415	174623	3.97	57.45	6.91

188 rows × 11 columns

The set variable helps to fix the variables as a index in a dataset

In [44]:

```
covid_index=covid_index.reset_index()
```

In [45]:

```
covid_index
```

Out[45]:

	index	No. of countries	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered
0	0	6	22-01-2020	555	17	538	510	0	0	0	3.06	5.05	60.71
1	1	8	23-01-2020	654	18	636	606	99	1	2	2.75	4.59	60.00
2	2	9	24-01-2020	941	26	915	879	287	8	6	2.76	3.83	72.22
3	3	11	25-01-2020	1434	42	1392	1353	493	16	3	2.93	2.72	107.69
4	4	13	26-01-2020	2118	56	2062	2010	684	14	13	2.64	2.46	107.69
...	...	...	...	...	...	...	...	...	...	...	...	...	...
183	183	187	23-07-2020	15510481	633506	14876975	6166006	282756	9966	169714	4.08	56.16	7.27
184	184	187	24-07-2020	15791645	639650	15151995	6212290	281164	6144	228736	4.05	56.61	7.16
185	185	187	25-07-2020	16047190	644517	15402673	6243930	255545	4867	219038	4.02	57.07	7.04
186	186	187	26-07-2020	16251796	648621	15603175	6309711	204606	4104	134721	3.99	57.18	6.98
187	187	187	27-07-2020	16480485	654036	15826449	6358362	228693	5415	174623	3.97	57.45	6.91

188 rows × 13 columns

The reset index function helps to remove the variables as a index in dataset.

In [31]:

```
covid_loc=covid.loc[(covid.Deaths>=600000)]
```

In [32]:

```
covid_loc
```

Out[32]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	No. of countries
178	18-07-2020	14292198	602130	13690068	5745518	237635	5627	150790	4.21	55.59	7.58	187
179	19-07-2020	14506845	606159	13900686	5868451	214647	4029	87685	4.18	55.37	7.55	187
180	20-07-2020	14713623	610319	14103304	5912527	206778	4160	158542	4.15	55.67	7.45	187
181	21-07-2020	14947078	616557	14330521	5965535	233565	6238	174209	4.12	55.96	7.37	187
182	22-07-2020	15227725	623540	14604185	6062930	280647	6983	176269	4.09	56.09	7.30	187
183	23-07-2020	15510481	633506	14876975	6166006	282756	9966	169714	4.08	56.16	7.27	187
184	24-07-2020	15791645	639650	15151995	6212290	281164	6144	228736	4.05	56.61	7.16	187
185	25-07-2020	16047190	644517	15402673	6243930	255545	4867	219038	4.02	57.07	7.04	187
186	26-07-2020	16251796	648621	15603175	6309711	204606	4104	134721	3.99	57.18	6.98	187
187	27-07-2020	16480485	654036	15826449	6358362	228693	5415	174623	3.97	57.45	6.91	187

The location function helps in finding particular observations and variables by using particular condition in dataset.

In [33]:

```
covid_iloc=covid.iloc[0:5,1:4]
```

In [39]:

```
covid_iloc
```

Out[39]:

	Confirmed	Deaths	Recovered
0	555	17	538
1	654	18	636
2	941	26	915
3	1434	42	1392
4	2118	56	2062

The ilocation function helps to get obeservations and variables at interger location in dataset

In [40]:

```
covid=covid.groupby()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_12520\591203190.py in <module>
----> 1 covid=covid.groupby()

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in groupby(self, by, axis, level, as_index, sort, g
roup_keys, squeeze, observed, dropna)
   7713
   7714         if level is None and by is None:
-> 7715             raise TypeError("You have to supply one of 'by' and 'level'")
   7716         axis = self._get_axis_number(axis)
   7717

TypeError: You have to supply one of 'by' and 'level'
```

The groupby command is not applicable in this dataset.

In [42]:

```
covid['No. of countries']=covid['No. of countries'].astype('object')
```

In [43]:

```
covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 188 entries, 0 to 187
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Date                   188 non-null    object
 1   Confirmed              188 non-null    int64
 2   Deaths                 188 non-null    int64
 3   Recovered              188 non-null    int64
 4   Active                 188 non-null    int64
 5   New cases              188 non-null    int64
 6   New deaths             188 non-null    int64
 7   New recovered          188 non-null    int64
 8   Deaths / 100 Cases     188 non-null    float64
 9   Recovered / 100 Cases  188 non-null    float64
 10  Deaths / 100 Recovered  188 non-null    float64
 11  No. of countries       188 non-null    object
dtypes: float64(3), int64(7), object(2)
memory usage: 19.1+ KB
```

The astype function helps to change the data type in dataset.