Getting Started with COVID Analysis - Dataset walkthrough. Get the output for the following questions also

Import Library

In [1]:
```
import pandas as pd
import numpy as np
```

Loading Dataset ¶

In [2]:
```
df = pd.read_csv('country_wise_latest.csv')

df.head()
```

Out[2]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	R
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	
1	Albania	4880	144	2745	1991	117	6	63	2.95	
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	
3	Andorra	907	52	803	52	10	0	0	5.73	
4	Angola	950	41	242	667	18	1	0	4.32	

In [3]:
```
df.shape
```

Out[3]:
```
(187, 15)
```

Walkthrough Dataset

Dataset information

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Country/Region          187 non-null    object 
 1   Confirmed               187 non-null    int64  
 2   Deaths                  187 non-null    int64  
 3   Recovered               187 non-null    int64  
 4   Active                  187 non-null    int64  
 5   New cases               187 non-null    int64  
 6   New deaths              187 non-null    int64  
 7   New recovered           187 non-null    int64  
 8   Deaths / 100 Cases      187 non-null    float64
 9   Recovered / 100 Cases   187 non-null    float64
 10  Deaths / 100 Recovered  187 non-null    float64
 11  Confirmed last week     187 non-null    int64  
 12  1 week change           187 non-null    int64  
 13  1 week % increase       187 non-null    float64
 14  WHO Region              187 non-null    object 
dtypes: float64(4), int64(9), object(2)
memory usage: 22.0+ KB
```

Describe Dataset - (mean, median, minimum , maximum)

In [5]:
```
df.describe()
```

Out[5]:

	Confirmed	Deaths	Recovered	Active	New cases	New deaths	
count	1.870000e+02	187.000000	1.870000e+02	1.870000e+02	187.000000	187.000000	1
mean	8.813094e+04	3497.518717	5.063148e+04	3.400194e+04	1222.957219	28.957219	9
std	3.833187e+05	14100.002482	1.901882e+05	2.133262e+05	5710.374790	120.037173	4
min	1.000000e+01	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	
25%	1.114000e+03	18.500000	6.265000e+02	1.415000e+02	4.000000	0.000000	
50%	5.059000e+03	108.000000	2.815000e+03	1.600000e+03	49.000000	1.000000	
75%	4.046050e+04	734.000000	2.260600e+04	9.149000e+03	419.500000	6.000000	2
max	4.290259e+06	148011.000000	1.846641e+06	2.816444e+06	56336.000000	1076.000000	33

Dataframe Column Names

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Country/Region', 'Confirmed', 'Deaths', 'Recovered', 'Active',
       'New cases', 'New deaths', 'New recovered', 'Deaths / 100 Cases',
       'Recovered / 100 Cases', 'Deaths / 100 Recovered',
       'Confirmed last week', '1 week change', '1 week % increase',
       'WHO Region'],
      dtype='object')
```

Checking Null Values

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Country/Region            0
Confirmed                 0
Deaths                    0
Recovered                 0
Active                    0
New cases                 0
New deaths                0
New recovered             0
Deaths / 100 Cases        0
Recovered / 100 Cases     0
Deaths / 100 Recovered    0
Confirmed last week       0
1 week change             0
1 week % increase         0
WHO Region                0
dtype: int64
```

About countries

In [8]:
```
countries = df['Country/Region'].unique()
countries
```

Out[8]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Andorra', 'Angola',
       'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
       'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh',
       'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bolivia', 'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'Brunei', 'Bulgaria', 'Burkina Faso', 'Burma', 'Burundi',
       'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada',
       'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia',
       'Comoros', 'Congo (Brazzaville)', 'Congo (Kinshasa)', 'Costa Rica',
       "Cote d'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czechia', 'Denmark',
       'Djibouti', 'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
       'Eswatini', 'Ethiopia', 'Fiji', 'Finland', 'France', 'Gabon',
       'Gambia', 'Georgia', 'Germany', 'Ghana', 'Greece', 'Greenland',
       'Grenada', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana',
       'Haiti', 'Holy See', 'Honduras', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kosovo',
       'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon', 'Lesotho',
       'Liberia', 'Libya', 'Liechtenstein', 'Lithuania', 'Luxembourg',
       'Madagascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta',
       'Mauritania', 'Mauritius', 'Mexico', 'Moldova', 'Monaco',
       'Mongolia', 'Montenegro', 'Morocco', 'Mozambique', 'Namibia',
       'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua', 'Niger',
       'Nigeria', 'North Macedonia', 'Norway', 'Oman', 'Pakistan',
       'Panama', 'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines',
       'Poland', 'Portugal', 'Qatar', 'Romania', 'Russia', 'Rwanda',
       'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Vincent and the Grenadines', 'San Marino',
       'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
       'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
       'Somalia', 'South Africa', 'South Korea', 'South Sudan', 'Spain',
       'Sri Lanka', 'Sudan', 'Suriname', 'Sweden', 'Switzerland', 'Syria',
       'Taiwan*', 'Tajikistan', 'Tanzania', 'Thailand', 'Timor-Leste',
       'Togo', 'Trinidad and Tobago', 'Tunisia', 'Turkey', 'US', 'Uganda',
       'Ukraine', 'United Arab Emirates', 'United Kingdom', 'Uruguay',
       'Uzbekistan', 'Venezuela', 'Vietnam', 'West Bank and Gaza',
       'Western Sahara', 'Yemen', 'Zambia', 'Zimbabwe'], dtype=object)
```

In [9]:
```
print('No of unique countries',len(countries))
```

```
No of unique countries 187
```

About WHO Region

In [10]:
```
region =  df['WHO Region'].unique()
region
```

Out[10]:
```
array(['Eastern Mediterranean', 'Europe', 'Africa', 'Americas',
       'Western Pacific', 'South-East Asia'], dtype=object)
```

In [11]:
```
print("No of unique WHO Region:",len(region))
```

```
No of unique WHO Region: 6
```

In [12]:
```
# return size of dataframe groupby WHO Region
data                    = df.groupby('WHO Region').size()

# Creating region dataframe
df_region               = pd.DataFrame()

# Adding Column 'WHO Region' and values of it
df_region['WHO Region'] = data.index

# Adding Column 'size' and values of it
df_region['size']       = data.values

#display region dataframe
df_region
```

Out[12]:

	WHO Region	size
0	Africa	48
1	Americas	35
2	Eastern Mediterranean	22
3	Europe	56
4	South-East Asia	10
5	Western Pacific	16