NAME = Aaraiz Ali

ID = 333564

ASSIGNMENT = DATA VISUALIZATION OF COVID 19 DATASET

In [1]:

```
import pandas as pd
```

Load and Display Data → Read the dataset and print the first 5 rows.

In [3]:

```
df = pd.read_csv("covid_19_country_wise_latest.csv")
```

In [5]:

```
df.head()
```

Out[5]:

	Unnamed: 0	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	re
0	0	Afghanistan	36263	1269	25198	9796	106.0	10.0	
1	1	Albania	4880	144	2745	1991	117.0	6.0	
2	2	Algeria	27973	1163	18837	7973	616.0	8.0	
3	3	Andorra	907	52	803	52	10.0	0.0	
4	4	Angola	950	41	242	667	18.0	1.0	

Set the Dataset index is Unnamed: 0

In [16]:

```
df.drop("Unnamed: 0", axis=1, inplace=True)
```

In [17]:

```
df
```

Out[17]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered
0	Afghanistan	36263	1269	25198	9796	106.0	10.0	18
1	Albania	4880	144	2745	1991	117.0	6.0	63
2	Algeria	27973	1163	18837	7973	616.0	8.0	749
3	Andorra	907	52	803	52	10.0	0.0	0
4	Angola	950	41	242	667	18.0	1.0	0
...	...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	78	3752	6791	152.0	2.0	0
183	Western Sahara	10	1	8	1	0.0	0.0	0
184	Yemen	1691	483	833	375	10.0	4.0	36
185	Zambia	4552	140	2815	1597	71.0	1.0	465
186	Zimbabwe	2704	non	542	2126	192.0	2.0	24

187 rows × 16 columns

Now replace the index column Unnamed: 0 name with index

In [25]:

```
df["Index"] = df.index
```

In [28]:

```
df.index.name = "Index"
```

In [29]:

```
df
```

Out[29]:

Index	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recoverec
0	Afghanistan	36263	1269	25198	9796	106.0	10.0	18
1	Albania	4880	144	2745	1991	117.0	6.0	63
2	Algeria	27973	1163	18837	7973	616.0	8.0	749
3	Andorra	907	52	803	52	10.0	0.0	0
4	Angola	950	41	242	667	18.0	1.0	0
...	...	...	...	...	...	...	...	..
182	West Bank and Gaza	10621	78	3752	6791	152.0	2.0	0
183	Western Sahara	10	1	8	1	0.0	0.0	0
184	Yemen	1691	483	833	375	10.0	4.0	36
185	Zambia	4552	140	2815	1597	71.0	1.0	465
186	Zimbabwe	2704	non	542	2126	192.0	2.0	24

187 rows × 16 columns

Create Function that take dataset columns and replace whitespace

```
    with Underscore and update dataset automatically and the
pass the
    dataset.
```

In [48]:

```
def update_string(df):
    df.columns = df.columns.str.replace(" ","_")
    return df
update_df = update_string(df)
```

In [46]:

```
update_df
```

Out[46]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths
Index							
0	Afghanistan	36263	1269	25198	9796	106.0	10.0
1	Albania	4880	144	2745	1991	117.0	6.0
2	Algeria	27973	1163	18837	7973	616.0	8.0
3	Andorra	907	52	803	52	10.0	0.0
4	Angola	950	41	242	667	18.0	1.0
...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	78	3752	6791	152.0	2.0
183	Western Sahara	10	1	8	1	0.0	0.0
184	Yemen	1691	483	833	375	10.0	4.0
185	Zambia	4552	140	2815	1597	71.0	1.0
186	Zimbabwe	2704	non	542	2126	192.0	2.0

187 rows × 16 columns

Check Basic Information → Find column names, data types, and

missing values.

In [49]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Country/Region          187 non-null    object 
 1   Confirmed               187 non-null    int64  
 2   Deaths                  187 non-null    object 
 3   Recovered               187 non-null    int64  
 4   Active                  187 non-null    int64  
 5   New_cases               177 non-null    float64
 6   New_deaths              177 non-null    float64
 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 
 15  Index                   187 non-null    int64  
dtypes: float64(6), int64(7), object(3)
memory usage: 23.5+ KB
```

```
In [50]: df.columns
```

```
Out[50]: 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', 'Index'],
               dtype='object')
```

```
In [57]: df.isna().sum()
```

```
Out[57]: 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
         Index                     0
         dtype: int64
```

fill the nan values with the mean of the column

In [62]:

```
df["New_cases"].fillna(df["New_cases"].mean())
df["New_deaths"].fillna(df["New_deaths"].mean())
```

Out[62]:

```
Index
0      10.0
1       6.0
2       8.0
3       0.0
4       1.0
       ...
182     2.0
183     0.0
184     4.0
185     1.0
186     2.0
Name: New_deaths, Length: 187, dtype: float64
```

In [64]:

```
df.isna().sum()
```

Out[64]:

```
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
Index                     0
dtype: int64
```

Count the number of unique countries in the dataset.

In [68]:

```
df["Country/Region"].nunique()
```

Out[68]:

```
187
```

Check if there are any duplicate country entries and remove them if

```
needed.
```

In [73]:

```
df["Country/Region"].duplicated().sum()
```

Out[73]:

```
0
```

Find the mean, median, and standard deviation of total cases.

```
In [77]: df[["Deaths_/_100_Cases", "Recovered_/_100_Cases"]].mean()
```

```
Out[77]: Deaths_/_100_Cases        3.019519
         Recovered_/_100_Cases    64.820535
         dtype: float64
```

```
In [78]: df[["Deaths_/_100_Cases", "Recovered_/_100_Cases"]].median()
```

```
Out[78]: Deaths_/_100_Cases        2.15
         Recovered_/_100_Cases    71.32
         dtype: float64
```

```
In [79]: df[["Deaths_/_100_Cases", "Recovered_/_100_Cases"]].std()
```

```
Out[79]: Deaths_/_100_Cases        3.454302
         Recovered_/_100_Cases    26.287694
         dtype: float64
```

Find out the string in Deaths Column and replace it with the

mean of Deaths column.

```
In [80]: df
```

Out[80]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths
Index							
0	Afghanistan	36263	1269	25198	9796	106.0	10.0
1	Albania	4880	144	2745	1991	117.0	6.0
2	Algeria	27973	1163	18837	7973	616.0	8.0
3	Andorra	907	52	803	52	10.0	0.0
4	Angola	950	41	242	667	18.0	1.0
...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	78	3752	6791	152.0	2.0
183	Western Sahara	10	1	8	1	0.0	0.0
184	Yemen	1691	483	833	375	10.0	4.0
185	Zambia	4552	140	2815	1597	71.0	1.0
186	Zimbabwe	2704	non	542	2126	192.0	2.0

187 rows × 16 columns

```
In [ ]:
```

```
In [92]: df["Deaths"] = pd.to_numeric(df["Deaths"], errors = "coerce")
```

```
In [104… df["Deaths"] = df["Deaths"].fillna(df["Deaths"].mean())
```

Change the datatype of column Deaths

```
In [109… df["Deaths"].dtype
```

```
Out[109… dtype('float64')
```

Calculate total number of Death and Recovery all over the

world

```
In [113… df["New_deaths"].sum()
```

```
Out[113… 18208.4011299435
```

```
In [114… df["New_recovered"].sum()
```

```
Out[114… 174623
```

Drop columns name WHO Region and Confirmed

```
In [123… df.drop("WHO_Region", axis=1)
```

Out [123…

	Country/Region	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths
Index							
0	Afghanistan	36263	NaN	25198	9796	106.0	10.0
1	Albania	4880	NaN	2745	1991	117.0	6.0
2	Algeria	27973	NaN	18837	7973	616.0	8.0
3	Andorra	907	NaN	803	52	10.0	0.0
4	Angola	950	NaN	242	667	18.0	1.0
...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	NaN	3752	6791	152.0	2.0
183	Western Sahara	10	NaN	8	1	0.0	0.0
184	Yemen	1691	NaN	833	375	10.0	4.0
185	Zambia	4552	NaN	2815	1597	71.0	1.0
186	Zimbabwe	2704	NaN	542	2126	192.0	2.0

187 rows × 15 columns

In [124…

```
df.drop("Confirmed", axis=1)
```

Out [124…

	Country/Region	Deaths	Recovered	Active	New_cases	New_deaths	New_recov
Index							
0	Afghanistan	NaN	25198	9796	106.0	10.0	
1	Albania	NaN	2745	1991	117.0	6.0	
2	Algeria	NaN	18837	7973	616.0	8.0	
3	Andorra	NaN	803	52	10.0	0.0	
4	Angola	NaN	242	667	18.0	1.0	
...	...	...	...	...	...	...	
182	West Bank and Gaza	NaN	3752	6791	152.0	2.0	
183	Western Sahara	NaN	8	1	0.0	0.0	
184	Yemen	NaN	833	375	10.0	4.0	
185	Zambia	NaN	2815	1597	71.0	1.0	
186	Zimbabwe	NaN	542	2126	192.0	2.0	

187 rows × 15 columns

In [128…

```
df["Deaths"] = df["Deaths"].fillna(0)
```

In [130…

```

```

Out [130…

	Country/Region	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths
Index							
0	Afghanistan	36263	0.0	25198	9796	106.0	10.0
1	Albania	4880	0.0	2745	1991	117.0	6.0
2	Algeria	27973	0.0	18837	7973	616.0	8.0
3	Andorra	907	0.0	803	52	10.0	0.0
4	Angola	950	0.0	242	667	18.0	1.0
...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	0.0	3752	6791	152.0	2.0
183	Western Sahara	10	0.0	8	1	0.0	0.0
184	Yemen	1691	0.0	833	375	10.0	4.0
185	Zambia	4552	0.0	2815	1597	71.0	1.0
186	Zimbabwe	2704	0.0	542	2126	192.0	2.0

187 rows × 16 columns

In [137…

```
df_3 = pd.read_csv("covid_19_country_wise_latest.csv")
df_3
```

Out [137…

	Unnamed: 0	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths
0	0	Afghanistan	36263	1269	25198	9796	106.0	10.0
1	1	Albania	4880	144	2745	1991	117.0	6.0
2	2	Algeria	27973	1163	18837	7973	616.0	8.0
3	3	Andorra	907	52	803	52	10.0	0.0
4	4	Angola	950	41	242	667	18.0	1.0
...	...	...	...	...	...	...	...	...
182	182	West Bank and Gaza	10621	78	3752	6791	152.0	2.0
183	183	Western Sahara	10	1	8	1	0.0	0.0
184	184	Yemen	1691	483	833	375	10.0	4.0
185	185	Zambia	4552	140	2815	1597	71.0	1.0
186	186	Zimbabwe	2704	non	542	2126	192.0	2.0

187 rows × 16 columns

In [142…

```
#df_3["Deaths"] = pd.to_numeric[df_3["Deaths"], errors='coerce']
df_3["Deaths"] = pd.to_numeric(df_3["Deaths"], errors='coerce')
```

In [146…

```
df_3["Deaths"] = df_3["Deaths"].fillna(df_3["Deaths"].mean())
```

In [147…

```
df_3
```

Out [147…

	Unnamed: 0	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	N dea
0	0	Afghanistan	36263	1269.000000	25198	9796	106.0	1
1	1	Albania	4880	144.000000	2745	1991	117.0	
2	2	Algeria	27973	1163.000000	18837	7973	616.0	
3	3	Andorra	907	52.000000	803	52	10.0	
4	4	Angola	950	41.000000	242	667	18.0	
...	...	...	...	...	...	...	...	
182	182	West Bank and Gaza	10621	78.000000	3752	6791	152.0	
183	183	Western Sahara	10	1.000000	8	1	0.0	
184	184	Yemen	1691	483.000000	833	375	10.0	
185	185	Zambia	4552	140.000000	2815	1597	71.0	
186	186	Zimbabwe	2704	3516.129032	542	2126	192.0	

187 rows × 16 columns

Find the Country with the Max Deaths

In [155…

```
death_max = df_3["Deaths"].max()
```

In [158…

```
max_death = df_3[df_3["Deaths"] == death_max]
max_death
```

Out [158…

	Unnamed: 0	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	N dea
173	173	US	4290259	148011.0	1325804	2816444	56336.0	107

Sort Countries by Deaths (Descending Order)

In [164…

```
df_3_sort = df_3.sort_values(by="Country/Region", ascending=True)
df_3_sort
```

Out [164…

	Unnamed: 0	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	N dea
0	0	Afghanistan	36263	1269.000000	25198	9796	106.0	
1	1	Albania	4880	144.000000	2745	1991	117.0	
2	2	Algeria	27973	1163.000000	18837	7973	616.0	
3	3	Andorra	907	52.000000	803	52	10.0	
4	4	Angola	950	41.000000	242	667	18.0	
...	...	...	...	...	...	...	...	
182	182	West Bank and Gaza	10621	78.000000	3752	6791	152.0	
183	183	Western Sahara	10	1.000000	8	1	0.0	
184	184	Yemen	1691	483.000000	833	375	10.0	
185	185	Zambia	4552	140.000000	2815	1597	71.0	
186	186	Zimbabwe	2704	3516.129032	542	2126	192.0	

187 rows × 16 columns

Make new column name Total_cases that have a sum of Deaths, Recovered , Active

In [172…

```
sum_up_all = df_3["Deaths"] + df_3["Recovered"] + df["Active"]
sum_up_all
```

Out[172…

```
0       36263.000000
1        4880.000000
2       27973.000000
3         907.000000
4         950.000000
            ...
182     10621.000000
183        10.000000
184      1691.000000
185      4552.000000
186      6184.129032
Length: 187, dtype: float64
```

In [173…

```
df_3[" Total_cases"] = sum_up_all
```

In [174…

```
df_3
```

Out[174…

	Unnamed: 0	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	N dea
0	0	Afghanistan	36263	1269.000000	25198	9796	106.0	
1	1	Albania	4880	144.000000	2745	1991	117.0	
2	2	Algeria	27973	1163.000000	18837	7973	616.0	
3	3	Andorra	907	52.000000	803	52	10.0	
4	4	Angola	950	41.000000	242	667	18.0	
...	...	...	...	...	...	...	...	
182	182	West Bank and Gaza	10621	78.000000	3752	6791	152.0	
183	183	Western Sahara	10	1.000000	8	1	0.0	
184	184	Yemen	1691	483.000000	833	375	10.0	
185	185	Zambia	4552	140.000000	2815	1597	71.0	
186	186	Zimbabwe	2704	3516.129032	542	2126	192.0	

187 rows × 17 columns

```
In [ ]:
```

```
In [ ]:
```