# Facebook Users by Country Analysis with Python

Problem Statement The Facebook Users by Country Data (Cleaned) dataset is a collection of information on Facebook users from different countries. The dataset contains five columns of data, which are named as follows:

Names: This column contains the names of the countries for which the data is collected. Users: This column provides the number of Facebook users in millions for each respective country. Facebook_Users: This column shows the percentage of the total population of each country that uses Facebook. Date_of_Data: This column indicates the date on which the data was collected and compiled. Population: This column represents the total population of each country.

## Import Library

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

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWar
ning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of Sc
iPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

## Uploading Csv fle

In [3]:
```
df = pd.read_csv(r"C:\Users\Syed Arif\Desktop\data.csv")
```

## Data Preprocessing

## .head()

head is used show to the By default = 5 rows in the dataset

In [4]:
```
df.head()
```
Out[4]:

| | Name | Users | Facebook_Users% | Date_of_Data | Population |
|---|---|---|---|---|---|
| **0** | India | 416.6M | 29.16% | 2021-06 | 1,428,627,663 |
| **1** | United States | 240M | 70.59% | 2020-12 | 339,996,563 |
| **2** | Indonesia | 176.5M | 63.6% | 2021-06 | 277,534,122 |
| **3** | Brazil | 139M | 64.23% | 2020-12 | 216,422,446 |
| **4** | Philippines | 91M | 77.55% | 2021-06 | 117,337,368 |

## .tail()

tail is used to show rows by Descending order

In [5]:
```
df.tail()
```
Out[5]:

| | Name | Users | Facebook_Users% | Date_of_Data | Population |
|---|---|---|---|---|---|
| **221** | Cook Islands | 2.7K | 15.84% | 2020-12 | 17,044 |
| **222** | Montserrat | 2.6K | 59.28% | 2020-12 | 4,386 |
| **223** | Niue | 820 | 42.38% | 2020-12 | 1,935 |
| **224** | Vatican City | 799 | 154.25% | 2020-12 | 518 |
| **225** | Tokelau | 410 | 21.66% | 2020-12 | 1,893 |

## .shape

It show the total no of rows & Column in the dataset

In [6]:
```
df.shape
```
Out[6]:
```
(226, 5)
```

## .Columns

It show the no of each Column

In [7]:
```
df.columns
```
Out[7]:
```
Index(['Name', 'Users', 'Facebook_Users%', 'Date_of_Data', 'Population'], dty
pe='object')
```

## .dtypes

This Attribute show the data type of each column

In [8]:
```
df.dtypes
```

Out[8]:
```
Name                  object
Users                 object
Facebook_Users%       object
Date_of_Data          object
Population            object
dtype: object
```

## .unique()

In a column, It show the unique value of specific column.

```
In [9]: df["Name"].unique()
```

```
Out[9]: array(['India', 'United States', 'Indonesia', 'Brazil', 'Philippines',
               'Mexico', 'Vietnam', 'Thailand', 'Japan', 'Pakistan', 'Egypt',
               'Bangladesh', 'Turkey', 'United Kingdom', 'Iran', 'France',
               'Germany', 'Italy', 'Nigeria', 'Argentina', 'Colombia', 'Malaysia',
               'Spain', 'Saudi Arabia', 'South Korea', 'Iraq', 'Algeria',
               'South Africa', 'Canada', 'Morocco', 'Taiwan', 'Myanmar', 'Peru',
               'Poland', 'Australia', 'Russia', 'Nepal', 'Venezuela', 'Chile',
               'Cambodia', 'Kazakhstan', 'Netherlands', 'Romania', 'Kenya',
               'United Arab Emirates', 'Ecuador', 'Ukraine', 'Sweden', 'Syria',
               'Tunisia', 'Sri Lanka', 'Ghana', 'Portugal', 'Guatemala',
               'Ethiopia', 'Belgium', 'Jordan', 'Israel', 'Hong Kong', 'Bolivia',
               'Hungary', 'Greece', 'Ivory Coast', 'Libya', 'Tanzania',
               'Dominican Republic', 'Austria', 'Singapore', 'Czech Republic',
               'Uzbekistan', 'Azerbaijan', 'Cameroon', 'Kuwait', 'Lebanon',
               'Denmark', 'China', 'Bulgaria', 'Belarus', 'Laos', 'Senegal',
               'Dr Congo', 'Switzerland', 'New Zealand', 'Honduras', 'Serbia',
               'El Salvador', 'Norway', 'Yemen', 'Uganda', 'Paraguay', 'Georgia',
               'Ireland', 'Finland', 'Costa Rica', 'Palestine', 'Oman', 'Qatar',
               'Madagascar', 'Angola', 'Mozambique', 'Mongolia', 'Slovakia',
               'Zambia', 'Nicaragua', 'Uruguay', 'Puerto Rico', 'Somalia',
               'Croatia', 'Mali', 'Panama', 'Burkina Faso', 'Guinea', 'Armenia',
               'Haiti', 'Albania', 'Benin', 'Bahrain', 'Lithuania', 'Zimbabwe',
               'Sudan', 'Botswana', 'Tajikistan', 'Jamaica', 'Slovenia',
               'North Macedonia', 'Cyprus', 'Mauritania', 'Mauritius', 'Togo',
               'Sierra Leone', 'Republic Of The Congo', 'Gabon', 'Rwanda',
               'Moldova', 'Namibia', 'Estonia', 'Latvia', 'Trinidad And Tobago',
               'Burundi', 'Liberia', 'Malawi', 'Fiji', 'Reunion', 'Niger',
               'Bhutan', 'Lesotho', 'Brunei', 'Chad', 'Timor Leste',
               'South Sudan', 'Macau', 'Gambia', 'Maldives', 'Malta',
               'Luxembourg', 'Guyana', 'Eswatini', 'Montenegro', 'Suriname',
               'Djibouti', 'Iceland', 'Turkmenistan', 'Bahamas', 'Guadeloupe',
               'French Polynesia', 'Belize', 'Comoros', 'New Caledonia',
               'Martinique', 'Barbados', 'Guam', 'Guinea Bissau',
               'Central African Republic', 'Samoa', 'Solomon Islands',
               'Equatorial Guinea', 'Curacao', 'Vanuatu', 'French Guiana',
               'Mayotte', 'Saint Lucia', 'Aruba', 'Tonga', 'Seychelles', 'Jersey',
               'Grenada', 'Saint Vincent And The Grenadines', 'Andorra',
               'Kiribati', 'Isle Of Man', 'Cayman Islands', 'Micronesia',
               'Dominica', 'Bermuda', 'Greenland', 'Saint Kitts And Nevis',
               'Guernsey', 'Northern Mariana Islands', 'Faroe Islands',
               'American Samoa', 'Sint Maarten', 'Western Sahara',
               'Turks And Caicos Islands', 'Marshall Islands', 'Gibraltar',
               'United States Virgin Islands', 'Saint Martin', 'North Korea',
               'Liechtenstein', 'Palau', 'Monaco', 'British Virgin Islands',
               'Anguilla', 'San Marino', 'Tuvalu', 'Saint Barthelemy', 'Eritrea',
               'Wallis And Futuna', 'Saint Pierre And Miquelon',
               'Falkland Islands', 'Nauru', 'Cook Islands', 'Montserrat', 'Niue',
               'Vatican City', 'Tokelau'], dtype=object)
```

## .nuique()

It will show the total no of unqe value from whole data frame

```
In [10]: df.nunique()
```

```
Out[10]: Name                226
         Users               180
         Facebook_Users%     224
         Date_of_Data          3
         Population          226
         dtype: int64
```

## .describe()

It show the Count, mean , median etc

```
In [11]: df.describe()
```

Out[11]:

| | Name | Users | Facebook_Users% | Date_of_Data | Population |
|---|---|---|---|---|---|
| **count** | 226 | 226 | 226 | 226 | 226 |
| **unique** | 226 | 180 | 224 | 3 | 226 |
| **top** | India | 3.4M | 23.28% | 2020-12 | 1,428,627,663 |
| **freq** | 1 | 4 | 2 | 179 | 1 |

## .value_counts

It Shows all the unique values with their count

```
In [12]: df["Name"].value_counts()
```

```
Out[12]: India          1
         Barbados       1
         Niger          1
         Bhutan         1
         Lesotho        1
                       ..
         Senegal        1
         Dr Congo       1
         Switzerland    1
         New Zealand    1
         Tokelau        1
         Name: Name, Length: 226, dtype: int64
```

## .isnull()

It shows the how many null values

In [13]:
```
df.isnull()
```

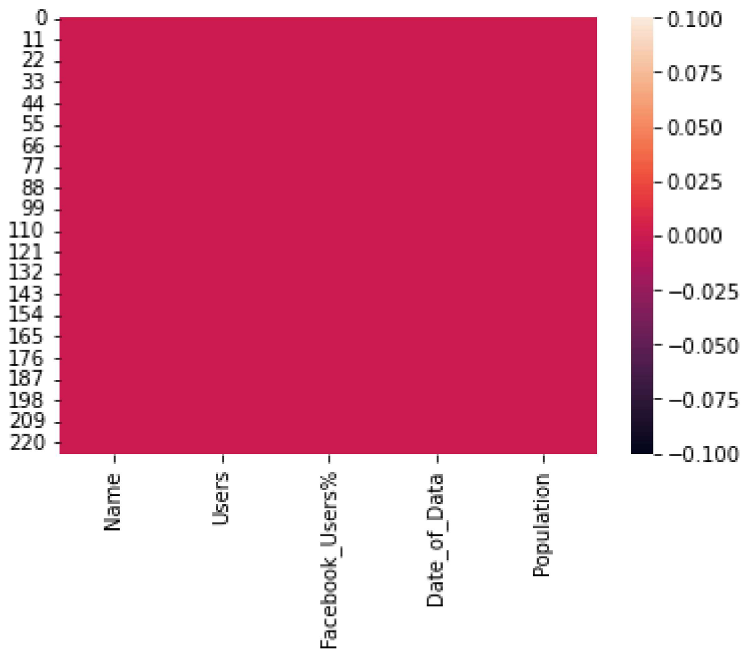
Out[13]:

| | Name | Users | Facebook_Users% | Date_of_Data | Population |
|---|---|---|---|---|---|
| **0** | False | False | False | False | False |
| **1** | False | False | False | False | False |
| **2** | False | False | False | False | False |
| **3** | False | False | False | False | False |
| **4** | False | False | False | False | False |
| **...** | ... | ... | ... | ... | ... |
| **221** | False | False | False | False | False |
| **222** | False | False | False | False | False |
| **223** | False | False | False | False | False |
| **224** | False | False | False | False | False |
| **225** | False | False | False | False | False |

226 rows × 5 columns

In [14]:
```
sns.heatmap(df.isnull())
```

Out[14]:
```
<AxesSubplot:>
```



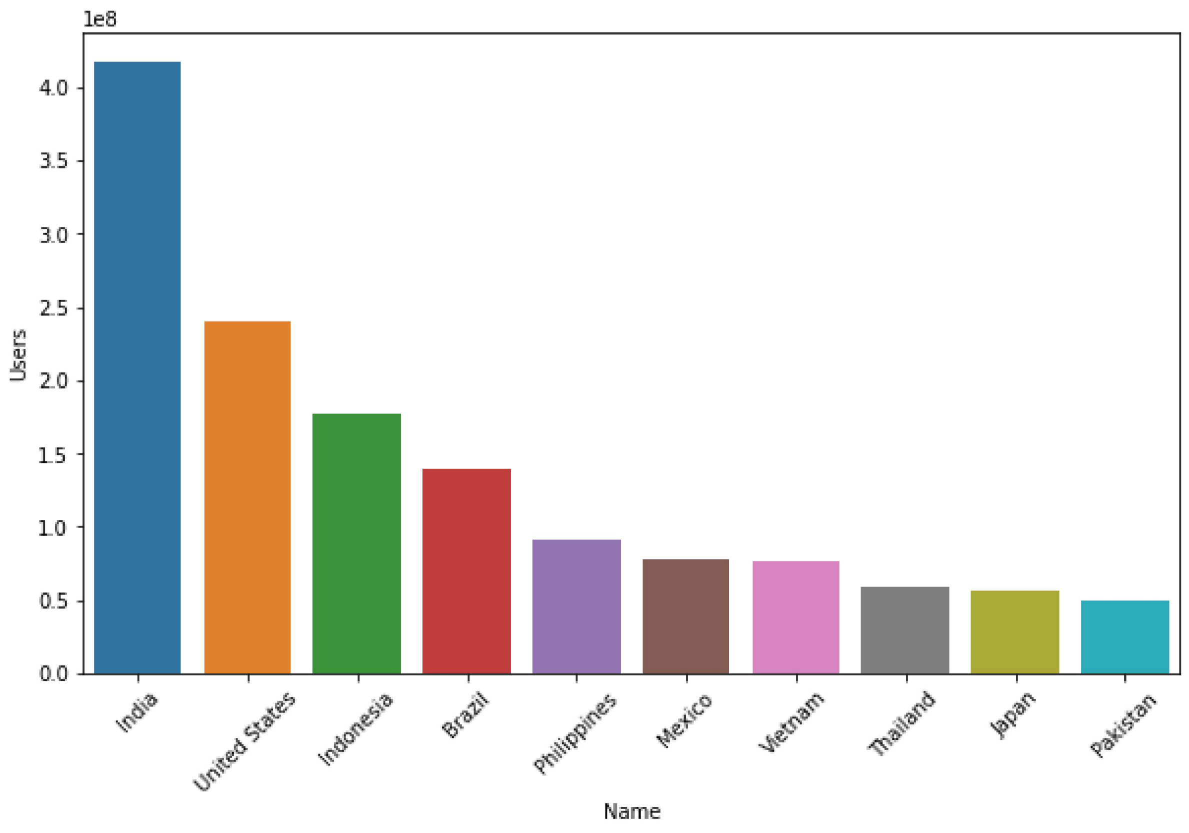
In [24]:
```
def remove_alphs(data):
    if 'M' in data:
        x= data.replace('M','')
        x= pd.to_numeric(x)
        return float(x*1000000)
    elif 'K' in data:
        x = data.replace('K','')
        x= pd.to_numeric(x)
        return float(x*1000)
```

In [25]:
```
df['Users'] = df['Users'].apply(lambda x: remove_alphs(x))
```

In [31]:
```
# Top Country by Facebook_User%
Top =df.nlargest(10,'Users')
```

In [32]:
```
plt.figure(figsize=(10,6))
sns.barplot(x= 'Name',y= 'Users',data= Top)
plt.xticks(rotation=45)
```

Out[32]:
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'India'),
  Text(1, 0, 'United States'),
  Text(2, 0, 'Indonesia'),
  Text(3, 0, 'Brazil'),
  Text(4, 0, 'Philippines'),
  Text(5, 0, 'Mexico'),
  Text(6, 0, 'Vietnam'),
  Text(7, 0, 'Thailand'),
  Text(8, 0, 'Japan'),
  Text(9, 0, 'Pakistan')])
```



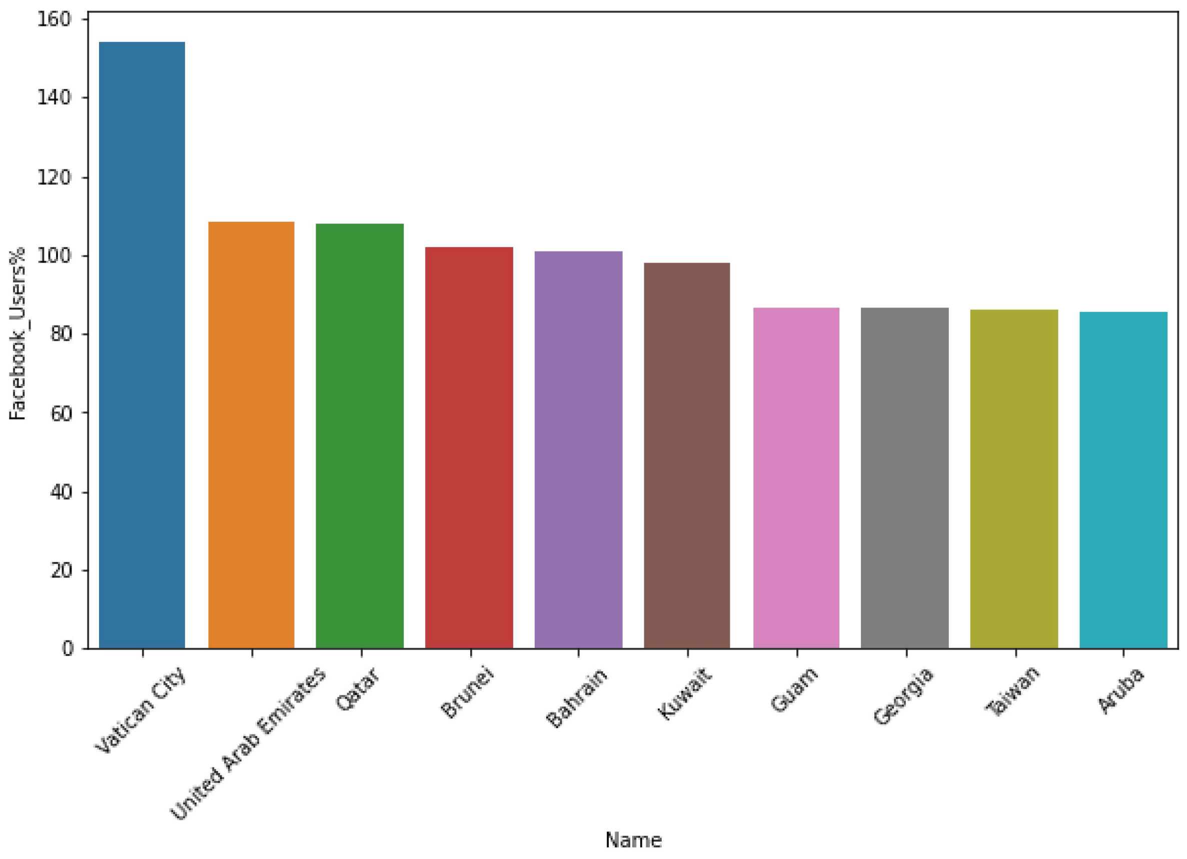
In [33]:
```
def remove_perc(data):
    if '%' in data:
        data = data.replace('%','')
        data = pd.to_numeric(data)
        return float(data)
```

In [34]:
```
df['Facebook_Users%'] = df['Facebook_Users%'].apply(lambda x: remove_perc(x))
```

In [35]:
```
per_users = df.nlargest(10,'Facebook_Users%')
```

In [36]:
```
plt.figure(figsize=(10,6))
sns.barplot(x= 'Name',y= 'Facebook_Users%',data=per_users)
plt.xticks(rotation=45)
```

Out[36]:
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Vatican City'),
  Text(1, 0, 'United Arab Emirates'),
  Text(2, 0, 'Qatar'),
  Text(3, 0, 'Brunei'),
  Text(4, 0, 'Bahrain'),
  Text(5, 0, 'Kuwait'),
  Text(6, 0, 'Guam'),
  Text(7, 0, 'Georgia'),
  Text(8, 0, 'Taiwan'),
  Text(9, 0, 'Aruba')])
```



In [37]:
```
df['Actual_Users'] = df['Users']*df['Facebook_Users%']
df.head()
```

Out[37]:

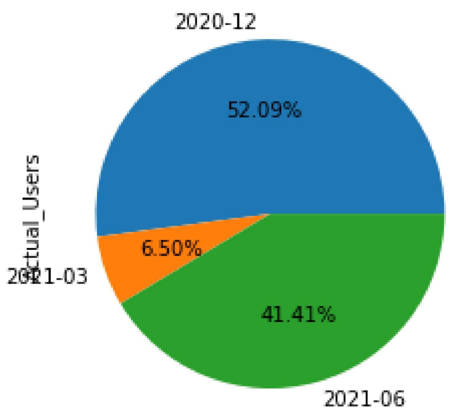
| | Name | Users | Facebook_Users% | Date_of_Data | Population | Actual_Users |
|---|---|---|---|---|---|---|
| **0** | India | 416600000.0 | 29.16 | 2021-06 | 1,428,627,663 | 1.214806e+10 |
| **1** | United States | 240000000.0 | 70.59 | 2020-12 | 339,996,563 | 1.694160e+10 |
| **2** | Indonesia | 176500000.0 | 63.60 | 2021-06 | 277,534,122 | 1.122540e+10 |
| **3** | Brazil | 139000000.0 | 64.23 | 2020-12 | 216,422,446 | 8.927970e+09 |
| **4** | Philippines | 91000000.0 | 77.55 | 2021-06 | 117,337,368 | 7.057050e+09 |

In [38]:
```
df = df.drop('Facebook_Users%',axis=1)
df.head()
```

Out[38]:

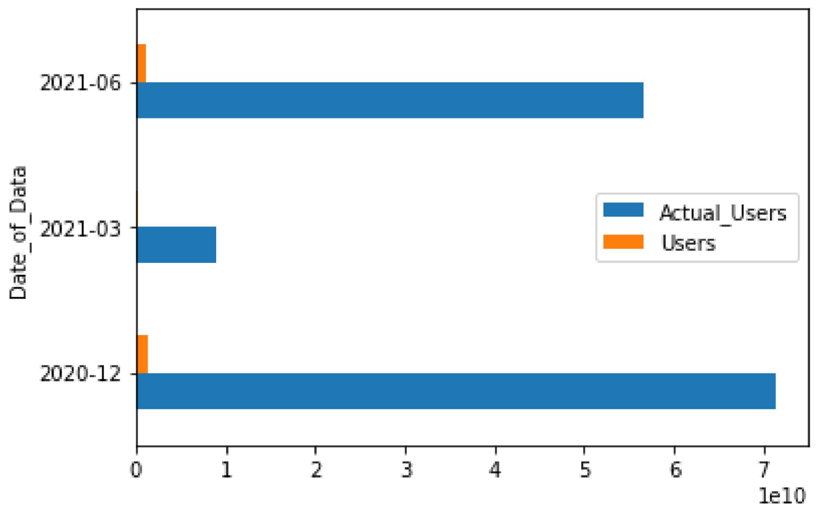
| | Name | Users | Date_of_Data | Population | Actual_Users |
|---|---|---|---|---|---|
| **0** | India | 416600000.0 | 2021-06 | 1,428,627,663 | 1.214806e+10 |
| **1** | United States | 240000000.0 | 2020-12 | 339,996,563 | 1.694160e+10 |
| **2** | Indonesia | 176500000.0 | 2021-06 | 277,534,122 | 1.122540e+10 |
| **3** | Brazil | 139000000.0 | 2020-12 | 216,422,446 | 8.927970e+09 |
| **4** | Philippines | 91000000.0 | 2021-06 | 117,337,368 | 7.057050e+09 |

In [39]:
```
df.groupby('Date_of_Data')['Actual_Users'].sum().plot(kind='pie',autopct='%0.2
plt.show()
```



In [41]:
```
df.groupby('Date_of_Data')['Actual_Users','Users'].sum().plot(kind='barh')
plt.show()
```

```
C:\Users\Syed Arif\AppData\Local\Temp\ipykernel_7584\1683901388.py:1: FutureW
arning: Indexing with multiple keys (implicitly converted to a tuple of keys)
will be deprecated, use a list instead.
  df.groupby('Date_of_Data')['Actual_Users','Users'].sum().plot(kind='barh')
```



```
In [ ]:
```