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

In [2]:
```
pd.__version__ #checking version of pandas
```

Out[2]:
```
'2.2.2'
```

In [3]:
```
df = pd.read_csv(r"C:\Users\Ramya\Downloads\data.csv") #df =object
df
```

Out[3]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

195 rows × 5 columns

In [4]:
```
id(df) #id of memory location
```

Out[4]:
```
2059209421888
```

In [5]:
```
len(df)
```

Out[5]:
```
195
```

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

Out[6]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [7]:
```
len(df.columns)
```

Out[7]:
```
5
```

In [8]:
```
df.isnull()                    #no there any false value       #checking missing val
```

Out[8]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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
...	...	...	...	...	...
190	False	False	False	False	False
191	False	False	False	False	False
192	False	False	False	False	False
193	False	False	False	False	False
194	False	False	False	False	False

195 rows × 5 columns

In [9]:
```
df.isna() #same code for checking wrong/missing
```

Out[9]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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
...	...	...	...	...	...
190	False	False	False	False	False
191	False	False	False	False	False
192	False	False	False	False	False
193	False	False	False	False	False
194	False	False	False	False	False

195 rows × 5 columns

In [10]:
```
df.isnull().sum() # i want count not false
```

Out[10]:
```
CountryName      0
CountryCode      0
BirthRate        0
InternetUsers    0
IncomeGroup      0
dtype: int64
```

In [11]:
```
df.head() #print top 5 rows by default it will print
```

Out[11]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income

In [12]:
```
df.tail() #bottom 5 rows
```

Out[12]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

In [13]:
```
df.info() #system,  has cateroical bydefault gives obj
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   CountryName    195 non-null    object 
 1   CountryCode    195 non-null    object 
 2   BirthRate      195 non-null    float64
 3   InternetUsers  195 non-null    float64
 4   IncomeGroup    195 non-null    object 
dtypes: float64(2), object(3)
memory usage: 7.7+ KB
```

In [14]:
```
df[:]
```

Out[14]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

195 rows × 5 columns

In [15]:
```
df[:1]
```

Out[15]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income

In [16]:
```
df[:-1] #silicing
```

Out[16]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
189	Samoa	WSM	26.172	15.3	Lower middle income
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income

194 rows × 5 columns

In [17]:
```
df[::-1] #reverse silicing
```

Out[17]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
194	Zimbabwe	ZWE	35.715	18.5	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
191	South Africa	ZAF	20.850	46.5	Upper middle income
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
...	...	...	...	...	...
4	United Arab Emirates	ARE	11.044	88.0	High income
3	Albania	ALB	12.877	57.2	Upper middle income
2	Angola	AGO	45.985	19.1	Upper middle income
1	Afghanistan	AFG	35.253	5.9	Low income
0	Aruba	ABW	10.244	78.9	High income

195 rows × 5 columns

In [18]:
```
df[1:100:10]
```

Out[18]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
1	Afghanistan	AFG	35.253	5.9000	Low income
11	Burundi	BDI	44.151	1.3000	Low income
21	Belize	BLZ	23.092	33.6000	Upper middle income
31	Switzerland	CHE	10.200	86.3400	High income
41	Cuba	CUB	10.400	27.9300	Upper middle income
51	Egypt, Arab Rep.	EGY	28.032	29.4000	Lower middle income
61	United Kingdom	GBR	12.200	89.8441	High income
71	Guatemala	GTM	27.465	19.7000	Lower middle income
81	Ireland	IRL	15.000	78.2477	High income
91	Kenya	KEN	35.194	39.0000	Lower middle income

In [19]:
```
df[10:21]
```

Out[19]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
10	Azerbaijan	AZE	18.300	58.70000	Upper middle income
11	Burundi	BDI	44.151	1.30000	Low income
12	Belgium	BEL	11.200	82.17020	High income
13	Benin	BEN	36.440	4.90000	Low income
14	Burkina Faso	BFA	40.551	9.10000	Low income
15	Bangladesh	BGD	20.142	6.63000	Lower middle income
16	Bulgaria	BGR	9.200	53.06150	Upper middle income
17	Bahrain	BHR	15.040	90.00004	High income
18	Bahamas, The	BHS	15.339	72.00000	High income
19	Bosnia and Herzegovina	BIH	9.062	57.79000	Upper middle income
20	Belarus	BLR	12.500	54.17000	Upper middle income

10th

In [20]:
```
df
```

Out[20]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

195 rows × 5 columns

In [21]:
```
df.head(2)
```

Out[21]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income

In [22]:
```
df.describe() #descriptive statstics(mean,median,mode)of numerical records
```

Out[22]:

	BirthRate	InternetUsers
count	195.000000	195.000000
mean	21.469928	42.076471
std	10.605467	29.030788
min	7.900000	0.900000
25%	12.120500	14.520000
50%	19.680000	41.000000
75%	29.759500	66.225000
max	49.661000	96.546800

In [23]:
```
df.head(1)
```

Out[23]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income

In [24]:
```
df.head(1) #cateriocial data to numerical data
```

Out[24]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income

In [25]:
```
df['CountryName']
```

Out[25]:
```
0                    Aruba
1              Afghanistan
2                   Angola
3                  Albania
4      United Arab Emirates
               ...
190            Yemen, Rep.
191           South Africa
192       Congo, Dem. Rep.
193                 Zambia
194               Zimbabwe
Name: CountryName, Length: 195, dtype: object
```

In [26]:
```
df['CountryCode']
```

Out[26]:
```
0      ABW
1      AFG
2      AGO
3      ALB
4      ARE
      ...
190    YEM
191    ZAF
192    COD
193    ZMB
194    ZWE
Name: CountryCode, Length: 195, dtype: object
```

In [27]:
```
df[['CountryName','CountryCode']]
```

Out[27]:

	CountryName	CountryCode
0	Aruba	ABW
1	Afghanistan	AFG
2	Angola	AGO
3	Albania	ALB
4	United Arab Emirates	ARE
...	...	...
190	Yemen, Rep.	YEM
191	South Africa	ZAF
192	Congo, Dem. Rep.	COD
193	Zambia	ZMB
194	Zimbabwe	ZWE

195 rows × 2 columns

In [28]:
```
df[['CountryName','CountryCode','IncomeGroup']]
```

Out[28]:

	CountryName	CountryCode	IncomeGroup
0	Aruba	ABW	High income
1	Afghanistan	AFG	Low income
2	Angola	AGO	Upper middle income
3	Albania	ALB	Upper middle income
4	United Arab Emirates	ARE	High income
...	...	...	...
190	Yemen, Rep.	YEM	Lower middle income
191	South Africa	ZAF	Upper middle income
192	Congo, Dem. Rep.	COD	Low income
193	Zambia	ZMB	Lower middle income
194	Zimbabwe	ZWE	Low income

195 rows × 3 columns

In [29]:
```
df_cat=df[['CountryName','CountryCode','IncomeGroup']]
df_cat
```

Out[29]:

	CountryName	CountryCode	IncomeGroup
0	Aruba	ABW	High income
1	Afghanistan	AFG	Low income
2	Angola	AGO	Upper middle income
3	Albania	ALB	Upper middle income
4	United Arab Emirates	ARE	High income
...	...	...	...
190	Yemen, Rep.	YEM	Lower middle income
191	South Africa	ZAF	Upper middle income
192	Congo, Dem. Rep.	COD	Low income
193	Zambia	ZMB	Lower middle income
194	Zimbabwe	ZWE	Low income

195 rows × 3 columns

In [30]:
```
print(len(df.columns))
print(len(df_cat.columns)) #cateroical data
```

```
5
3
```

In [31]:
```
print((df.columns))
```

```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [32]:
```
print((df_cat.columns))
```

```
Index(['CountryName', 'CountryCode', 'IncomeGroup'], dtype='object')
```

In [33]:
```
df_cat.describe()
```

Out[33]:

	CountryName	CountryCode	IncomeGroup
count	195	195	195
unique	195	195	4
top	Aruba	ABW	High income
freq	1	1	67

In [34]:
```
df_num=df[['BirthRate','InternetUsers']]
df_num
```

Out[34]:

	BirthRate	InternetUsers
0	10.244	78.9
1	35.253	5.9
2	45.985	19.1
3	12.877	57.2
4	11.044	88.0
...	...	...
190	32.947	20.0
191	20.850	46.5
192	42.394	2.2
193	40.471	15.4
194	35.715	18.5

195 rows × 2 columns

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   CountryName    195 non-null    object 
 1   CountryCode    195 non-null    object 
 2   BirthRate      195 non-null    float64
 3   InternetUsers  195 non-null    float64
 4   IncomeGroup    195 non-null    object 
dtypes: float64(2), object(3)
memory usage: 7.7+ KB
```

In [36]:
```
df_cat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   CountryName  195 non-null    object
 1   CountryCode  195 non-null    object
 2   IncomeGroup  195 non-null    object
dtypes: object(3)
memory usage: 4.7+ KB
```

In [37]:
```
df_num.info() #constructor is building
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   BirthRate      195 non-null    float64
 1   InternetUsers  195 non-null    float64
dtypes: float64(2)
memory usage: 3.2 KB
```

In [38]:
```
df.describe()
```

Out[38]:

	BirthRate	InternetUsers
count	195.000000	195.000000
mean	21.469928	42.076471
std	10.605467	29.030788
min	7.900000	0.900000
25%	12.120500	14.520000
50%	19.680000	41.000000
75%	29.759500	66.225000
max	49.661000	96.546800

In [39]:
```
df.describe().transpose() #transpose the result – flipping rows to columns and c
```

Out[39]:

	count	mean	std	min	25%	50%	75%	max
BirthRate	195.0	21.469928	10.605467	7.9	12.1205	19.68	29.7595	49.6610
InternetUsers	195.0	42.076471	29.030788	0.9	14.5200	41.00	66.2250	96.5468

In [40]:
```
df.describe().T #transpose or T # same result
```

Out[40]:

	count	mean	std	min	25%	50%	75%	max
BirthRate	195.0	21.469928	10.605467	7.9	12.1205	19.68	29.7595	49.6610
InternetUsers	195.0	42.076471	29.030788	0.9	14.5200	41.00	66.2250	96.5468

In [41]:
```
df.columns #countryname to 'A',br to ;B'
```

Out[41]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [42]:
```
df.columns=['a','b','c','d','e']
```

In [43]:
```
df.head(1)
```

Out[43]:

	a	b	c	d	e
0	Aruba	ABW	10.244	78.9	High income

In [44]:
```
df.columns=['CountryName','CountryCode','BirthRate','InternetUsers','IncomeGroup
df.head(1)
```

Out[44]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income

In [45]:
```
df[['CountryCode','BirthRate','InternetUsers']][4:8]#subset of dataset
```

Out[45]:

	CountryCode	BirthRate	InternetUsers
4	ARE	11.044	88.0
5	ARG	17.716	59.9
6	ARM	13.308	41.9
7	ATG	16.447	63.4

In [46]:
```
df[4:8][['CountryCode','BirthRate','InternetUsers']]
```

Out[46]:

	CountryCode	BirthRate	InternetUsers
4	ARE	11.044	88.0
5	ARG	17.716	59.9
6	ARM	13.308	41.9
7	ATG	16.447	63.4

In [47]:
```
df.columns
```

Out[47]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [48]:
```
df.BirthRate * df.InternetUsers
```

Out[48]:
```
0      808.2516
1      207.9927
2      878.3135
3      736.5644
4      971.8720
         ...
190    658.9400
191    969.5250
192     93.2668
193    623.2534
194    660.7275
Length: 195, dtype: float64
```

In [49]:
```
df.head(2)
```

Out[49]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income

In [50]:
```
df['newcloumn']=df.BirthRate * df.InternetUsers
```

In [51]:
```
df.head(5)
```

Out[51]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
0	Aruba	ABW	10.244	78.9	High income	808.2516
1	Afghanistan	AFG	35.253	5.9	Low income	207.9927
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
3	Albania	ALB	12.877	57.2	Upper middle income	736.5644
4	United Arab Emirates	ARE	11.044	88.0	High income	971.8720

In [52]:
```
len(df.columns)
```

Out[52]: 6

In [53]:
```
df = df.drop('newcolumn', axis=1, errors='ignore')
```

In [54]:
```
df.head(1)
```

Out[54]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
0	Aruba	ABW	10.244	78.9	High income	808.2516

In [55]:
```
df
```

Out[55]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
0	Aruba	ABW	10.244	78.9	High income	808.2516
1	Afghanistan	AFG	35.253	5.9	Low income	207.9927
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
3	Albania	ALB	12.877	57.2	Upper middle income	736.5644
4	United Arab Emirates	ARE	11.044	88.0	High income	971.8720
...	...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income	658.9400
191	South Africa	ZAF	20.850	46.5	Upper middle income	969.5250
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	93.2668
193	Zambia	ZMB	40.471	15.4	Lower middle income	623.2534
194	Zimbabwe	ZWE	35.715	18.5	Low income	660.7275

195 rows × 6 columns

In [78]:
```
df.InternetUsers<2
```

Out[78]:
```
0      False
1      False
2      False
3      False
4      False
       ...
190    False
191    False
192    False
193    False
194    False
Name: InternetUsers, Length: 195, dtype: bool
```

In [58]:
```
df[df.InternetUsers<2]
```

Out[58]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
11	Burundi	BDI	44.151	1.3	Low income	57.3963
52	Eritrea	ERI	34.800	0.9	Low income	31.3200
55	Ethiopia	ETH	32.925	1.9	Low income	62.5575
64	Guinea	GIN	37.337	1.6	Low income	59.7392
117	Myanmar	MMR	18.119	1.6	Lower middle income	28.9904
127	Niger	NER	49.661	1.7	Low income	84.4237
154	Sierra Leone	SLE	36.729	1.7	Low income	62.4393
156	Somalia	SOM	43.891	1.5	Low income	65.8365
172	Timor-Leste	TLS	35.755	1.1	Lower middle income	39.3305

In [59]:
```
len(df[df.InternetUsers<2])
```

Out[59]:
```
9
```

In [60]:
```
df.BirthRate>40
```

Out[60]:
```
0      False
1      False
2       True
3      False
4      False
       ...
190    False
191    False
192     True
193     True
194    False
Name: BirthRate, Length: 195, dtype: bool
```

In [61]:
```
df[df.BirthRate>40]
```

Out[61]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
11	Burundi	BDI	44.151	1.3	Low income	57.3963
14	Burkina Faso	BFA	40.551	9.1	Low income	369.0141
65	Gambia, The	GMB	42.525	14.0	Low income	595.3500
115	Mali	MLI	44.138	3.5	Low income	154.4830
127	Niger	NER	49.661	1.7	Low income	84.4237
128	Nigeria	NGA	40.045	38.0	Lower middle income	1521.7100
156	Somalia	SOM	43.891	1.5	Low income	65.8365
167	Chad	TCD	45.745	2.3	Low income	105.2135
178	Uganda	UGA	43.474	16.2	Low income	704.2788
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	93.2668
193	Zambia	ZMB	40.471	15.4	Lower middle income	623.2534

In [62]:
```
Filter = df.InternetUsers < 2
```

In [63]:
```
Filter2 = df.BirthRate >40
```

In [82]:
```
df[Filter & Filter2]
```

Out[82]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
11	Burundi	BDI	44.151	1.3	Low income	57.3963
127	Niger	NER	49.661	1.7	Low income	84.4237
156	Somalia	SOM	43.891	1.5	Low income	65.8365

11th

In [81]:
```
df
```

Out[81]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
0	Aruba	ABW	10.244	78.9	High income	808.2516
1	Afghanistan	AFG	35.253	5.9	Low income	207.9927
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
3	Albania	ALB	12.877	57.2	Upper middle income	736.5644
4	United Arab Emirates	ARE	11.044	88.0	High income	971.8720
...	...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income	658.9400
191	South Africa	ZAF	20.850	46.5	Upper middle income	969.5250
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	93.2668
193	Zambia	ZMB	40.471	15.4	Lower middle income	623.2534
194	Zimbabwe	ZWE	35.715	18.5	Low income	660.7275

195 rows × 6 columns

In [66]:
```
df_num
```

Out[66]:

	BirthRate	InternetUsers
0	10.244	78.9
1	35.253	5.9
2	45.985	19.1
3	12.877	57.2
4	11.044	88.0
...	...	...
190	32.947	20.0
191	20.850	46.5
192	42.394	2.2
193	40.471	15.4
194	35.715	18.5

195 rows × 2 columns

In [67]:
```
df_cat
```

Out[67]:

	CountryName	CountryCode	IncomeGroup
0	Aruba	ABW	High income
1	Afghanistan	AFG	Low income
2	Angola	AGO	Upper middle income
3	Albania	ALB	Upper middle income
4	United Arab Emirates	ARE	High income
...	...	...	...
190	Yemen, Rep.	YEM	Lower middle income
191	South Africa	ZAF	Upper middle income
192	Congo, Dem. Rep.	COD	Low income
193	Zambia	ZMB	Lower middle income
194	Zimbabwe	ZWE	Low income

195 rows × 3 columns

In [68]:
```
df[df.IncomeGroup == 'High income']
```

Out[68]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
0	Aruba	ABW	10.244	78.90	High income	808.25160
4	United Arab Emirates	ARE	11.044	88.00	High income	971.87200
5	Argentina	ARG	17.716	59.90	High income	1061.18840
7	Antigua and Barbuda	ATG	16.447	63.40	High income	1042.73980
8	Australia	AUS	13.200	83.00	High income	1095.60000
...	...	...	...	...	...	...
174	Trinidad and Tobago	TTO	14.590	63.80	High income	930.84200
180	Uruguay	URY	14.374	57.69	High income	829.23606
181	United States	USA	12.500	84.20	High income	1052.50000
184	Venezuela, RB	VEN	19.842	54.90	High income	1089.32580
185	Virgin Islands (U.S.)	VIR	10.700	45.30	High income	484.71000

67 rows × 6 columns

In [69]:
```
df[df.IncomeGroup == 'Low income']
```

Out[69]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newcloumn
1	Afghanistan	AFG	35.253	5.90	Low income	207.99270
11	Burundi	BDI	44.151	1.30	Low income	57.39630
13	Benin	BEN	36.440	4.90	Low income	178.55600
14	Burkina Faso	BFA	40.551	9.10	Low income	369.01410
29	Central African Republic	CAF	34.076	3.50	Low income	119.26600
38	Comoros	COM	34.326	6.50	Low income	223.11900
52	Eritrea	ERI	34.800	0.90	Low income	31.32000
55	Ethiopia	ETH	32.925	1.90	Low income	62.55750
64	Guinea	GIN	37.337	1.60	Low income	59.73920
65	Gambia, The	GMB	42.525	14.00	Low income	595.35000
66	Guinea-Bissau	GNB	37.503	3.10	Low income	116.25930
77	Haiti	HTI	25.345	10.60	Low income	268.65700
93	Cambodia	KHM	24.462	6.80	Low income	166.34160
99	Liberia	LBR	35.521	3.20	Low income	113.66720
111	Madagascar	MDG	34.686	3.00	Low income	104.05800
115	Mali	MLI	44.138	3.50	Low income	154.48300
120	Mozambique	MOZ	39.705	5.40	Low income	214.40700
123	Malawi	MWI	39.459	5.05	Low income	199.26795
127	Niger	NER	49.661	1.70	Low income	84.42370
132	Nepal	NPL	20.923	13.30	Low income	278.27590
148	Rwanda	RWA	32.689	9.00	Low income	294.20100
154	Sierra Leone	SLE	36.729	1.70	Low income	62.43930
156	Somalia	SOM	43.891	1.50	Low income	65.83650
158	South Sudan	SSD	37.126	14.10	Low income	523.47660
167	Chad	TCD	45.745	2.30	Low income	105.21350
168	Togo	TGO	36.080	4.50	Low income	162.36000
177	Tanzania	TZA	39.518	4.40	Low income	173.87920
178	Uganda	UGA	43.474	16.20	Low income	704.27880
192	Congo, Dem. Rep.	COD	42.394	2.20	Low income	93.26680
194	Zimbabwe	ZWE	35.715	18.50	Low income	660.72750

In [70]:
```
df.IncomeGroup.unique() # income
```

Out[70]:
```
array(['High income', 'Low income', 'Upper middle income',
       'Lower middle income'], dtype=object)
```

In [71]:
```
df.IncomeGroup.nunique() # no.of unique group
```

Out[71]: 4

we analysis python dataset

In [72]:
```
import matplotlib.pyplot as plt #visualization
import seaborn as sns #stata visualization,advances visualzation

%matplotlib inline #plot the graph in the line
plt.rcParams['figure.figsize'] = 6,2 #(rc parameter) a parameter comes from plt

import warnings
warnings.filterwarnings('ignore') # when os updated ,to ignore the error msg
```

```
UsageError: unrecognized arguments: #plot the graph in the line
```

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

In [ ]:
```
df['InternetUsers']
```

vis1 = plt.distplot(df["InternetUsers"])

In [ ]:
```
vis1 = sns.distplot(df["InternetUsers"]) #adding seaborn
```

In [ ]:
```
vis1 = sns.distplot(df["InternetUsers"]) #or this code # univariate analysis
plt.show(vis1)
```

In [ ]:
```
vis2 = sns.displot(df["InternetUsers"])# removing t distribution line gone
plt.show(vis2)
```

In [ ]:
```
vis3 = sns.distplot(df["InternetUsers"],bins=15) #or this code # univariate anal
plt.show(vis3)
```

In [ ]:
```
vis3 = sns.distplot(df["InternetUsers"],bins=20) #or this code # univariate anal
plt.show(vis3)
```

In [ ]:
```
plt.rcParams['figure.figsize'] = 6,3
```

In [ ]:
```
vis4 = sns.boxplot(data = df,x='IncomeGroup',y='BirthRate') #bivariable
plt.show(vis4)
```

In [ ]:
```
vis5 = sns.lmplot(data = df,x='InternetUsers',y='BirthRate') #linear
plt.show(vis5)
```

In [ ]:
```
vis6 = sns.lmplot(data = df,x='InternetUsers',y='BirthRate',fit_reg=False) #no l
plt.show(vis6)
```

In [ ]:
```
vis7 = sns.lmplot(data = df,x='InternetUsers',y='BirthRate',fit_reg=True) #reges
plt.show(vis7)
```

In [ ]:
```
vis8 = sns.lmplot(data = df,x='InternetUsers',y='BirthRate',fit_reg=False,hue='I
plt.show(vis8)
```

In [ ]:
```
vis9 = sns.lmplot(data = df,x='InternetUsers',y='BirthRate',fit_reg=True,hue='In
plt.show(vis9)
```

In [ ]:
```

```