```
In [6]: pip install --upgrade pandas pyarrow scipy matplotlib seaborn
```

```
Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-package
s (2.3.1)
Requirement already satisfied: pyarrow in c:\users\hp\anaconda3\lib\site-packag
es (20.0.0)
Requirement already satisfied: scipy in c:\users\hp\anaconda3\lib\site-packages
(1.16.0)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-pac
kages (3.10.3)
Requirement already satisfied: seaborn in c:\users\hp\anaconda3\lib\site-packag
es (0.13.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\hp\anaconda3\lib\site-
packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\anaconda3\
lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-p
ackages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\hp\anaconda3\lib\sit
e-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\anaconda3\lib\si
te-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-p
ackages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\anaconda3\lib\s
ite-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\hp\anaconda3\lib\s
ite-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\anaconda3\lib\sit
e-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in c:\users\hp\appdata\roaming\python\
python312\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hp\anaconda3\lib\si
te-packages (from matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\roaming\python\p
ython312\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
In [7]: import pandas as pd
```

```
In [8]: df=pd.read_csv(r"C:\Users\HP\Downloads\data.csv")
```

```
In [9]: df
```

Out[9]:

	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 [10]:
```
id(df)
```

Out[10]:
```
1784955200352
```

In [11]:
```
len(df)
```

Out[11]:
```
195
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [13]:
```
len(df.columns)
```

Out[13]:
```
5
```

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

Out[14]:

	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 [15]:

```
df.isna()   #isnull and isna both are same
```

Out[15]:

	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 [16]:

```
df.isnull().sum()  # to have the count for each attribute
```

Out[16]:

```
CountryName      0
CountryCode      0
BirthRate        0
InternetUsers    0
IncomeGroup      0
dtype: int64
```

In [17]:

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

Out[17]:

```
CountryName      0
CountryCode      0
BirthRate        0
InternetUsers    0
IncomeGroup      0
dtype: int64
```

In [18]:

```
df.head()
```

Out[18]:

	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 [19]:

```
df.tail()
```

Out[19]:

	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 [20]:

```
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 [21]:

```
df[::-1]
```

Out[21]:

	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 [22]:

```
df
```

Out[22]:

	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 [23]:

```
df[1:200:10]
```

Out[23]:

	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
101	St. Lucia	LCA	15.430	46.2000	Upper middle income
111	Madagascar	MDG	34.686	3.0000	Low income
121	Mauritania	MRT	33.801	6.2000	Lower middle income
131	Norway	NOR	11.600	95.0534	High income
141	Puerto Rico	PRI	10.800	73.9000	High income
151	Senegal	SEN	38.533	13.1000	Lower middle income
161	Slovak Republic	SVK	10.100	77.8826	High income
171	Turkmenistan	TKM	21.322	9.6000	Upper middle income
181	United States	USA	12.500	84.2000	High income
191	South Africa	ZAF	20.850	46.5000	Upper middle income

In [24]:

```
df.describe()  #by default gives us numerical data only
               # descriptive statstics always displays numerical records
```

Out[24]:

	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 [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','CountryName']]
```

Out[26]:

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

195 rows × 2 columns

In [27]:

```
df[['CountryCode','CountryName','IncomeGroup']]
```

Out[27]:

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

195 rows × 3 columns

In [28]:

```
df_cat=df[['CountryCode','CountryName','IncomeGroup']]
df_cat
```

Out[28]:

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

195 rows × 3 columns

In [29]:

```
len(df_cat)
```

Out[29]:

```
195
```

In [30]:

```
print(len(df.columns))
print(len(df_cat.columns))    #the thing which works behind the computer to all
```

```
5
3
```

In [31]:

```
df_cat.describe().T #descriptive stastics for categorical data
```

Out[31]:

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

In [32]:

```
df_num=df[['BirthRate'        ,'InternetUsers']]
df_num
```

Out[32]:

	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 [33]:

```
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 [34]:

```
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   CountryCode  195 non-null    object
 1   CountryName  195 non-null    object
 2   IncomeGroup  195 non-null    object
dtypes: object(3)
memory usage: 4.7+ KB
```

In [35]:

```
df_num.info()
```

```
<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 [36]:
```
df.describe().transpose()
```

Out[36]:

	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 [37]:
```
df.describe().T
```

Out[37]:

	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 [38]:
```
df.columns=['a','b','c','d','e']  #renaming the attributes/ columns
df
```

Out[38]:

	a	b	c	d	e
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 [39]:

```
df.columns
```

Out[39]:

```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

In [40]:

```
df.columns=['CountryName','CountryCode','BirthRate','InternetUsers','IncomeGrc
```

In [41]:

```
df
```

Out[41]:

	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 [42]:

```
df[['CountryName','CountryCode']][4:10] #specific subset from dataframe
```

Out[42]:

	CountryName	CountryCode
4	United Arab Emirates	ARE
5	Argentina	ARG
6	Armenia	ARM
7	Antigua and Barbuda	ATG
8	Australia	AUS
9	Austria	AUT

In [43]:
```
df[4:10][['CountryName','CountryCode']]  #for subset another way
```

Out[43]:

	CountryName	CountryCode
4	United Arab Emirates	ARE
5	Argentina	ARG
6	Armenia	ARM
7	Antigua and Barbuda	ATG
8	Australia	AUS
9	Austria	AUT

In [44]:
```
df.BirthRate*df.InternetUsers # i want to add this data in new column in an ex
```

Out[44]:
```
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 [45]:
```
df['new column']=df.BirthRate*df.InternetUsers
```

In [46]:
```
df.head()
```

Out[46]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	new column
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 [47]:
```
len(df.columns) # we learnt to add new coolumn in an existing dataframe
```

Out[47]:
```
6
```

In [48]:
```
df
```

Out[48]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	n colu
0	Aruba	ABW	10.244	78.9	High income	808.2
1	Afghanistan	AFG	35.253	5.9	Low income	207.9
2	Angola	AGO	45.985	19.1	Upper middle income	878.3
3	Albania	ALB	12.877	57.2	Upper middle income	736.5
4	United Arab Emirates	ARE	11.044	88.0	High income	971.8
...	...	...	...	...	...	
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income	658.9
191	South Africa	ZAF	20.850	46.5	Upper middle income	969.5
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	93.2
193	Zambia	ZMB	40.471	15.4	Lower middle income	623.2
194	Zimbabwe	ZWE	35.715	18.5	Low income	660.7

195 rows × 6 columns

In [49]:
```
df=df.drop('new column',axis=1)   #axis=1 becoz we are deleting a column and w
```

In [50]:
```
df.head()
```

Out[50]:

	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 [173…
```
df[df.InternetUsers<2]
```

Out[173…

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

In [183…

```
df[df.IncomeGroup == 'High income']
```

Out[183…

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

67 rows × 5 columns

In [51]:

```
#the country  with internet users are less than 2

filter1=df.InternetUsers<2
```

In [52]:
```
filter2=df.BirthRate>40
filter2                    # countries with higher birth rate
```

Out[52]:
```
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 [53]:
```
df[filter1 & filter2]   # when we combine low internet users and high birth ra
```

Out[53]:

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

In [54]:
```
df.IncomeGroup.unique()   # the specific values that contains by each attribut
```

Out[54]:
```
array(['High income', 'Low income', 'Upper middle income',
       'Lower middle income'], dtype=object)
```

In [55]:
```
import matplotlib.pyplot as plt #visualization
import seaborn as sns           #advaced visualization

%matplotlib inline        # plot the graph in the line
plt.rcParams['figure.figsize'] = 6,2 # rcParams comes from plt lib where fig s

import warnings
warnings.filterwarnings('ignore')   #whenever os will update.ignore the os wil
```

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

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

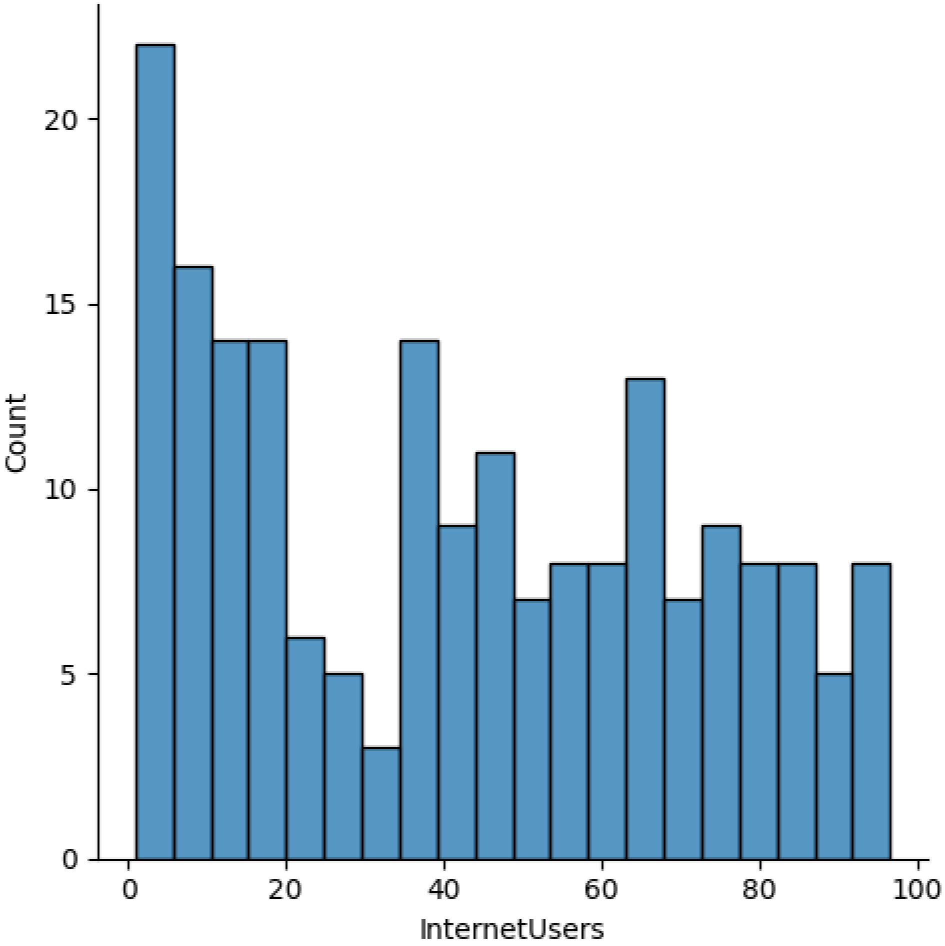
Internet Users Distribution

the below graph shows how the usage of internet varies globally

Many countries have either very high (>70%) or very low (<20%) internet usage.

In [132…

```
vis1=sns.displot(df['InternetUsers'],bins=20)   # uni varient analaysis --> pl
plt.show(vis1)                                  # bins ===>indepth analysis wi
```



In [126…

```
vis2=sns.distplot(df['InternetUsers'])   # uni varient analaysis --> plot the
plt.show(vis2)
```

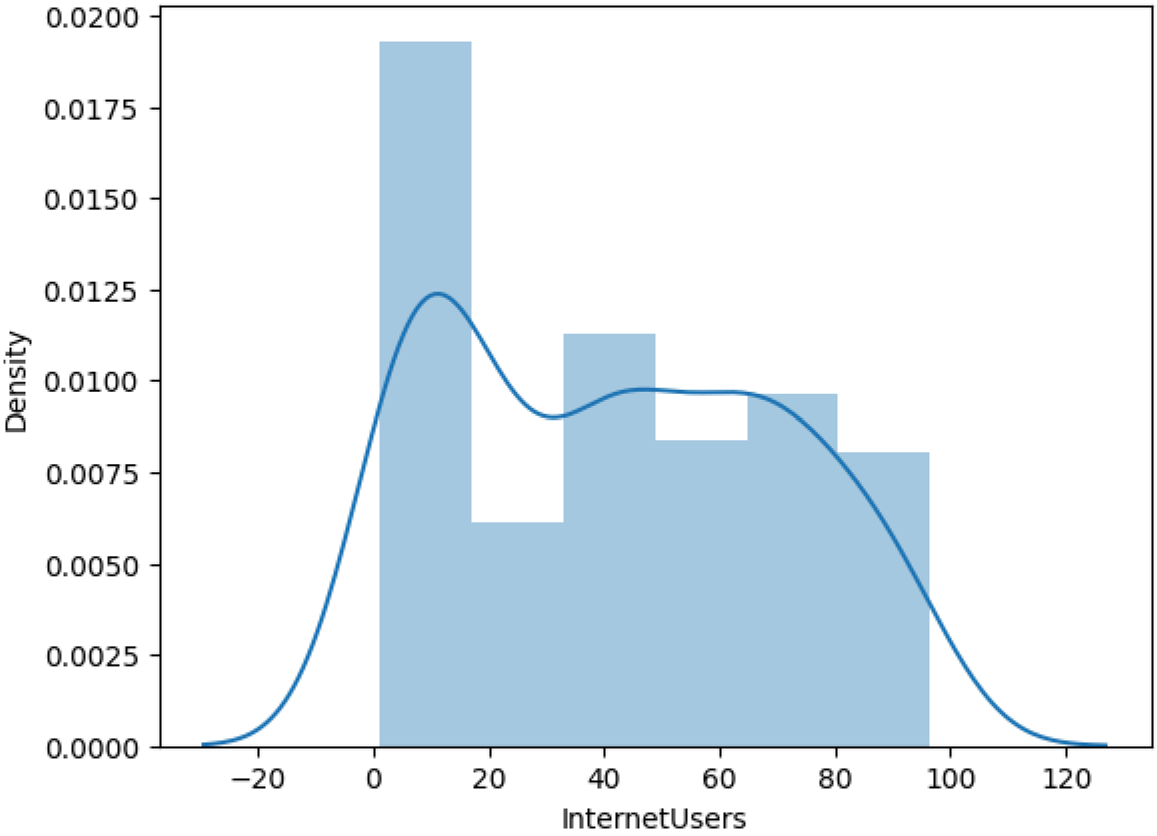
```
C:\Users\HP\AppData\Local\Temp\ipykernel_23396\1449659713.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  vis2=sns.distplot(df['InternetUsers'])   # uni varient analaysis --> plot the
graph using one attribute
```



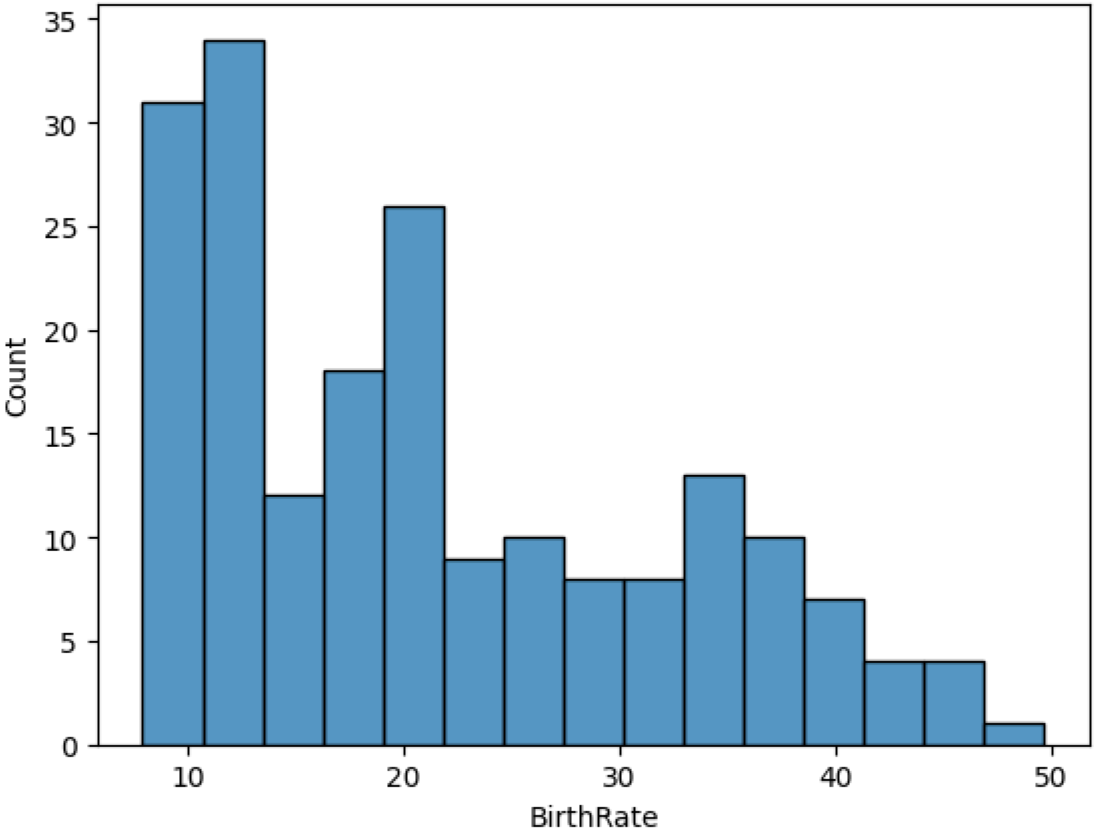
Birth Rate Distribution

most countries have birth rate between 10 and 40 AND

only few countries having birth rate >=45

In [136…

```
vis3=sns.histplot(df['BirthRate'],bins=15)
plt.show(vis3)
```



In [ ]:

```
plt.rcParams['figure.figsize'] = 6,3    #we can change the size of figure heig
```

Income Group VS BirthRate

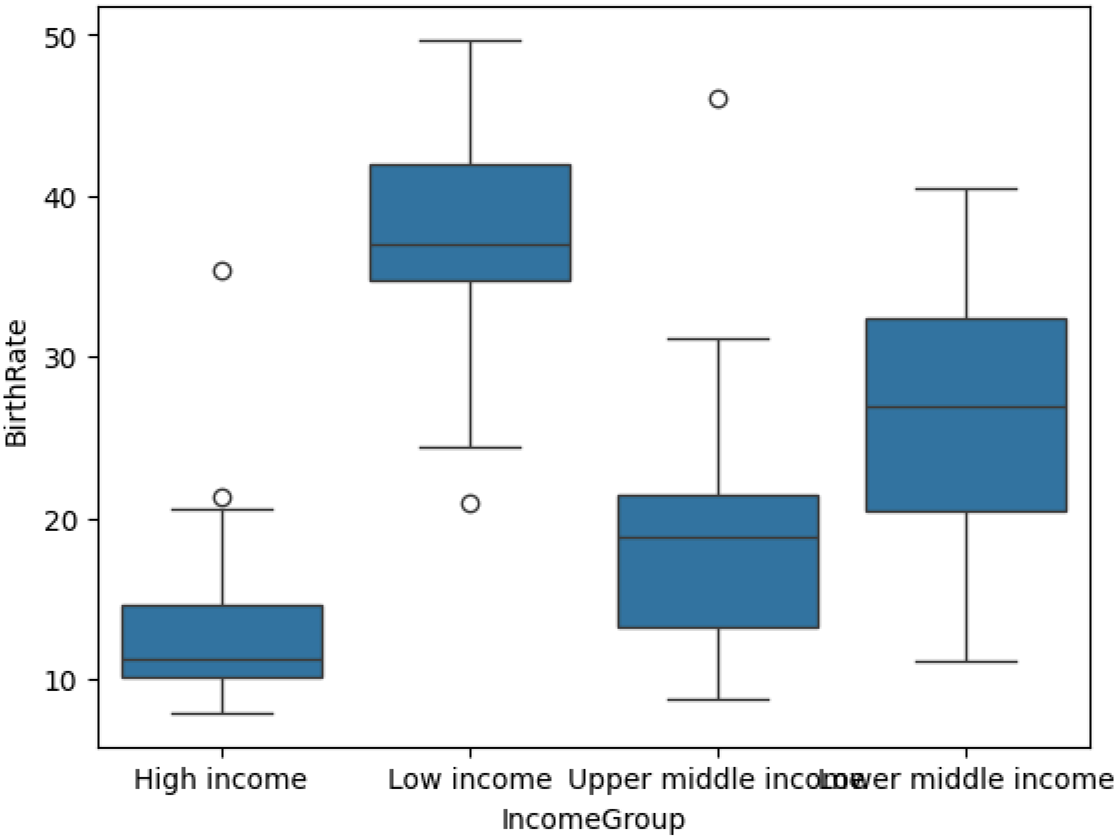
Mostly the high income countries having low birth rate <=20

Mostly coutries with low income having higher birth rate >20

Mostly middle income countries having birth rates [10<birth rate<40]

In [60]:

```
vis4=sns.boxplot(data=df,x='IncomeGroup',y='BirthRate')   #plot the graph uisn
plt.show(vis4)
```



In [ ]:

```
# outlier is the datapoint which is very far from  other datapoints and it is
# ml algos that handle outliers ---logistic regression  (sigmoid)  and  navie
```

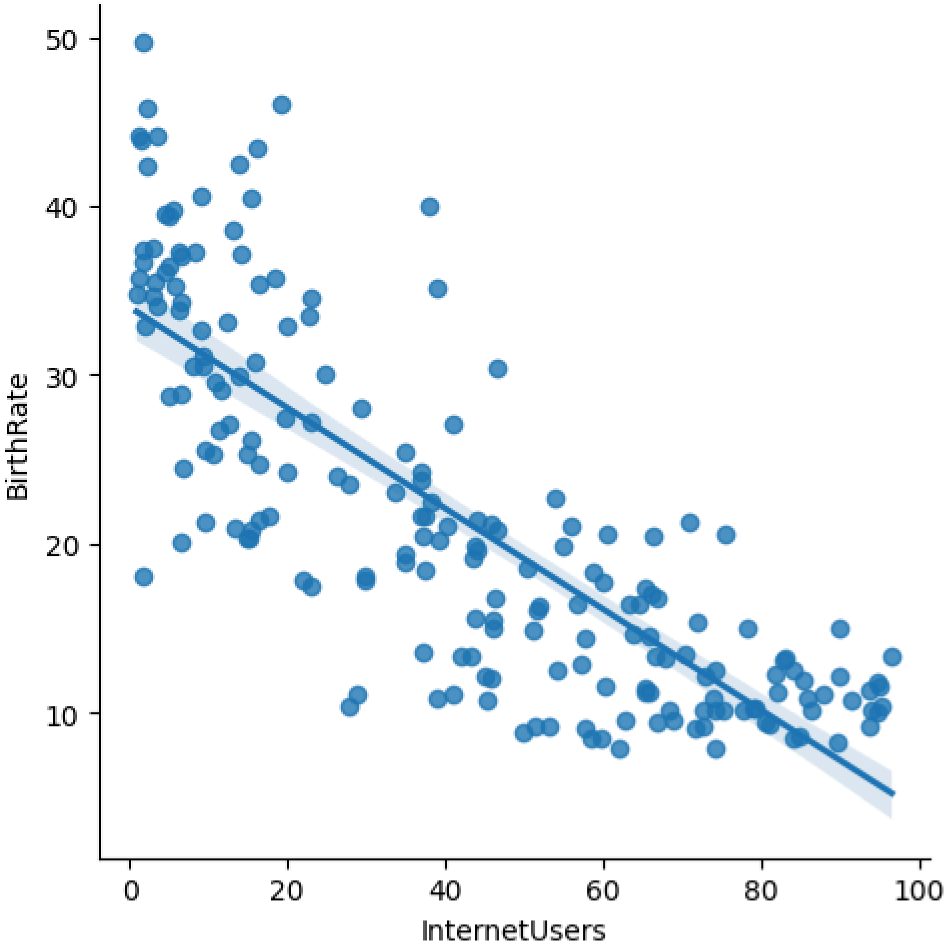
BirthRate VS InternetUsers

High-income countries → high internet use, low birth rate.

Low-income countries → low internet use, high birth rate.

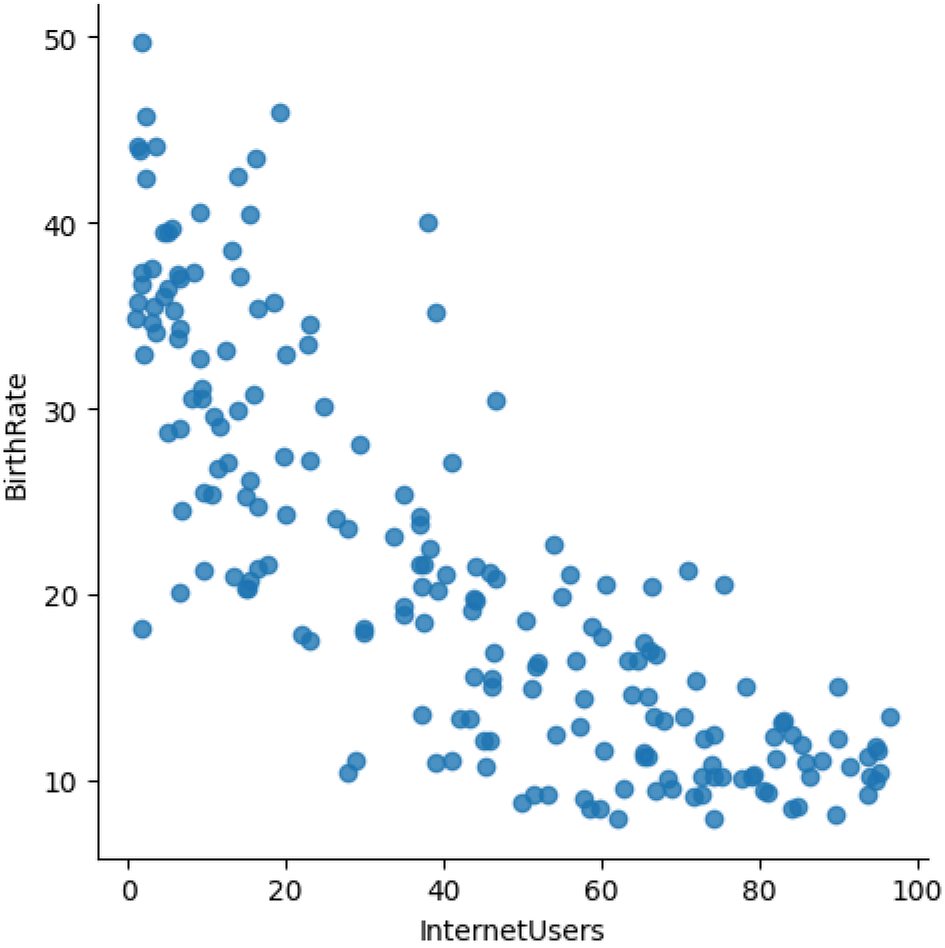
In [157…

```
vis5=sns.lmplot(data=df,x='InternetUsers',y='BirthRate')   #plot the graph uis
plt.show(vis5)   #linear model graph
```



In [66]:

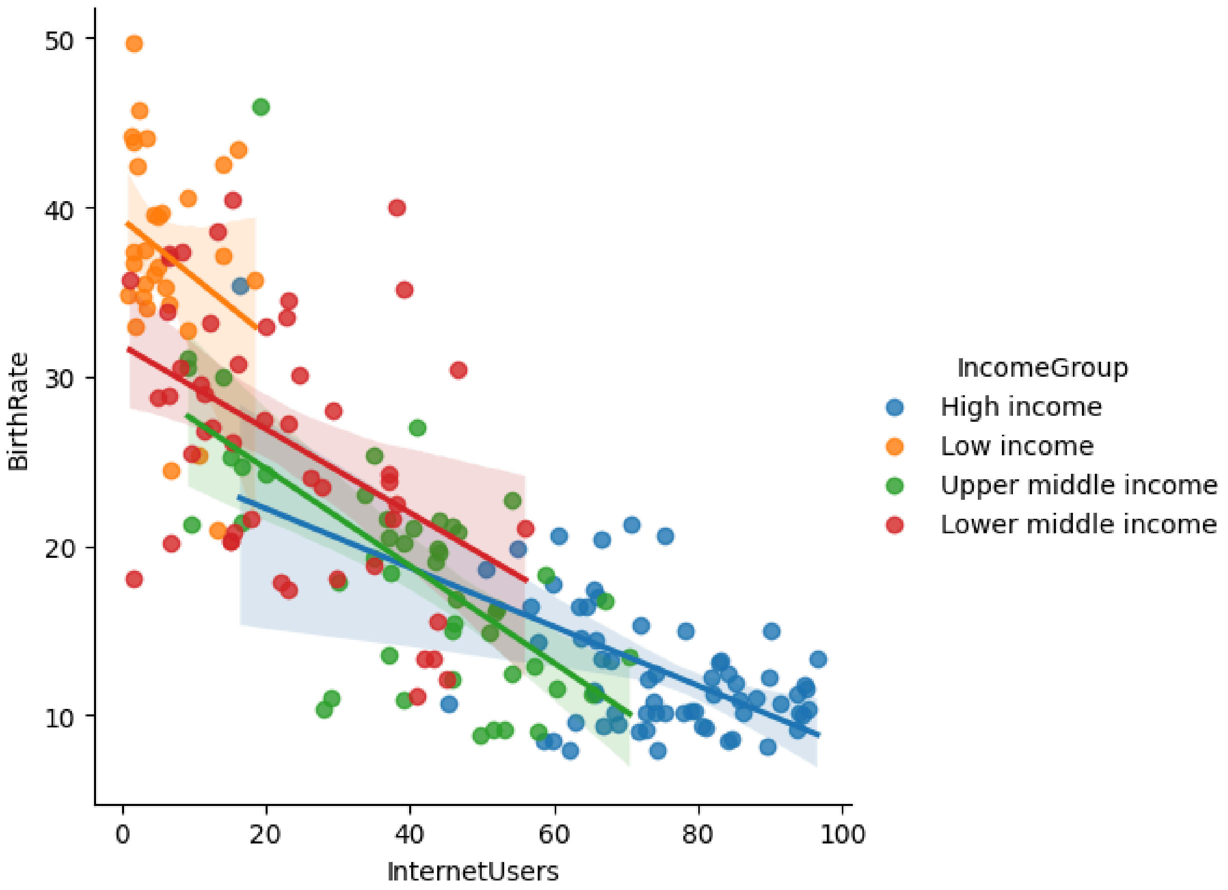
```
vis5=sns.lmplot(data=df,x='InternetUsers',y='BirthRate',fit_reg=False)   #plot
plt.show(vis5)
```



In [64]:

```
vis5=sns.lmplot(data=df,x='InternetUsers',y='BirthRate',fit_reg=True,hue='Inco

plt.show(vis5)
```



In [ ]:

```
vis5=sns.scatterplot(data=df,x='InternetUsers',y='BirthRate',hue='IncomeGroup'

plt.show(vis5)
```

In [166…

```
grouped_df=df.groupby('IncomeGroup')[['InternetUsers','BirthRate']].mean()
```

In [168…

```
grouped_df
```

Out[168…

	InternetUsers	BirthRate
IncomeGroup		
High income	74.231684	12.753433
Low income	5.988333	37.238267
Lower middle income	22.366386	26.309140
Upper middle income	40.279577	18.740646

As income increases, Internet access rises and birth rates drop.

In [ ]: