

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

```
In [2]: data = pd.read_csv(r'S:\DOCS\1st\data.csv')
```

```
In [3]: data
```

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]: len(data)
```

Out[4]: 195

```
In [5]: data.columns
```

Out[5]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
'IncomeGroup'],
dtype='object')

```
In [7]: data.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 [8]: data.shape # total rows and columns
```

Out[8]: (195, 5)

```
In [9]: data.head() # top 5 rows by default
```

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

```
In [10]: data.tail() # last 5 rows by default
```

Out[10]:

	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 [11]: data.head(10) # top 10 rows
```

Out[11]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9000	High income
1	Afghanistan	AFG	35.253	5.9000	Low income
2	Angola	AGO	45.985	19.1000	Upper middle income
3	Albania	ALB	12.877	57.2000	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0000	High income
5	Argentina	ARG	17.716	59.9000	High income
6	Armenia	ARM	13.308	41.9000	Lower middle income
7	Antigua and Barbuda	ATG	16.447	63.4000	High income
8	Australia	AUS	13.200	83.0000	High income
9	Austria	AUT	9.400	80.6188	High income

```
In [12]: data.tail(10) # Last 10 rows
```

```
Out[12]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
185	Virgin Islands (U.S.)	VIR	10.700	45.3	High income
186	Vietnam	VNM	15.537	43.9	Lower middle income
187	Vanuatu	VUT	26.739	11.3	Lower middle income
188	West Bank and Gaza	PSE	30.394	46.6	Lower middle 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	Zimbabwe	ZWE	35.715	18.5	Low income

```
In [13]: data.describe() # will give numerical information
```

```
Out[13]:
```

	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 [14]: data.describe().transpose() # row to columns
```

```
Out[14]:
```

	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 [15]: data[0:6] # slicing
```

```
Out[15]:
```

	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
5	Argentina	ARG	17.716	59.9	High income

```
In [18]: data[:10] # upto 10 rows
```

```
Out[18]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9000	High income
1	Afghanistan	AFG	35.253	5.9000	Low income
2	Angola	AGO	45.985	19.1000	Upper middle income
3	Albania	ALB	12.877	57.2000	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0000	High income
5	Argentina	ARG	17.716	59.9000	High income
6	Armenia	ARM	13.308	41.9000	Lower middle income
7	Antigua and Barbuda	ATG	16.447	63.4000	High income
8	Australia	AUS	13.200	83.0000	High income
9	Austria	AUT	9.400	80.6188	High income

```
In [19]: data[::20] # all rows multiply by 20
```

```
Out[19]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9000	High income
20	Belarus	BLR	12.500	54.1700	Upper middle income
40	Costa Rica	CRI	15.022	45.9600	Upper middle income
60	Gabon	GAB	30.555	9.2000	Upper middle income
80	India	IND	20.291	15.1000	Lower middle income
100	Libya	LBY	21.425	16.5000	Upper middle income
120	Mozambique	MOZ	39.705	5.4000	Low income
140	Poland	POL	9.600	62.8492	High income
160	Suriname	SUR	18.455	37.4000	Upper middle income
180	Uruguay	URY	14.374	57.6900	High income

```
In [20]: data[::-10] # reverse dataset
```

```
Out[20]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
194	Zimbabwe	ZWE	35.715	18.5000	Low income
184	Venezuela, RB	VEN	19.842	54.9000	High income
174	Trinidad and Tobago	TTO	14.590	63.8000	High income
164	Swaziland	SWZ	30.093	24.7000	Lower middle income
154	Sierra Leone	SLE	36.729	1.7000	Low income
144	French Polynesia	PYF	16.393	56.8000	High income
134	Oman	OMN	20.419	66.4500	High income
124	Malaysia	MYS	16.805	66.9700	Upper middle income
114	Macedonia, FYR	MKD	11.222	65.2400	Upper middle income
104	Lesotho	LSO	28.738	5.0000	Lower middle income
94	Kiribati	KIR	29.044	11.5000	Lower middle income
84	Iceland	ISL	13.400	96.5468	High income
74	Hong Kong SAR, China	HKG	7.900	74.2000	High income
64	Guinea	GIN	37.337	1.6000	Low income
54	Estonia	EST	10.300	79.4000	High income
44	Czech Republic	CZE	10.200	74.1104	High income
34	Cote d'Ivoire	CIV	37.320	8.4000	Lower middle income
24	Brazil	BRA	14.931	51.0400	Upper middle income
14	Burkina Faso	BFA	40.551	9.1000	Low income
4	United Arab Emirates	ARE	11.044	88.0000	High income

```
In [21]: data['CountryName'] # one column
```

```
Out[21]: 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 [24]: data[['CountryName', 'InternetUsers']]
```

```
Out[24]:
```

	CountryName	InternetUsers
0	Aruba	78.9
1	Afghanistan	5.9
2	Angola	19.1
3	Albania	57.2
4	United Arab Emirates	88.0
...
190	Yemen, Rep.	20.0
191	South Africa	46.5
192	Congo, Dem. Rep.	2.2
193	Zambia	15.4
194	Zimbabwe	18.5

195 rows × 2 columns

```
In [25]: data[['CountryName', 'InternetUsers']][10:30]
```

```
Out[25]:
```

	CountryName	InternetUsers
10	Azerbaijan	58.70000
11	Burundi	1.30000
12	Belgium	82.17020
13	Benin	4.90000
14	Burkina Faso	9.10000
15	Bangladesh	6.63000
16	Bulgaria	53.06150
17	Bahrain	90.00004
18	Bahamas, The	72.00000
19	Bosnia and Herzegovina	57.79000
20	Belarus	54.17000
21	Belize	33.60000
22	Bermuda	95.30000
23	Bolivia	36.94000
24	Brazil	51.04000
25	Barbados	73.00000
26	Brunei Darussalam	64.50000
27	Bhutan	29.90000
28	Botswana	15.00000
29	Central African Republic	3.50000

```
In [26]: data.BirthRate * data.InternetUsers # mathematical operation
```

```
Out[26]: 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 [27]: data['mycalc']=data.BirthRate*data.InternetUsers # add a new column name mycalc
```

```
In [28]: data
```

```
Out[28]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
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 [29]: data=data.drop('mycalc', axis=1) ''' removing column my calc using drop function
labels: The labels of the rows or columns to drop.
axis: The axis along which to drop the labels. 0 refers to the rows and 1 refers to the columns
```

```
In [30]: data
```

```
Out[30]:
```

	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 [31]: data.InternetUsers<2 # checking condition true or false
```

```
Out[31]: 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 [32]: filters=data.InternetUsers<2 # creating a filter
```

```
In [34]: filters
```

```
Out[34]: 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 [36]: data[filters] # applying filter, showing only those value which is true
```

```
Out[36]:
```

	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 [37]: filters2= data.BirthRate>40 # created 2nd filter
```

```
In [38]: data[filters2] # records with birthrate more than 40
```

```
Out[38]:
```

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

```
In [40]: data[data.IncomeGroup == 'Low income'] # shwing data that have only Low income
```

```
Out[40]:
```

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

```
In [41]: data.IncomeGroup.unique() #shows only unique value
```

```
Out[41]: array(['High income', 'Low income', 'Upper middle income',  
               'Lower middle income'], dtype=object)
```

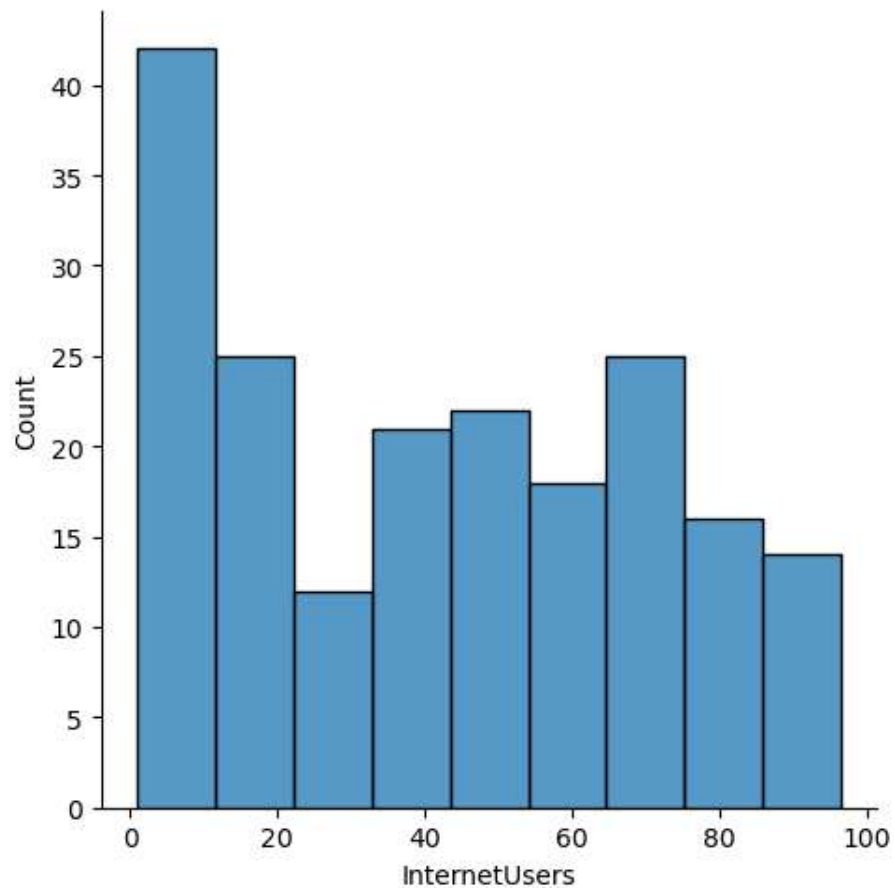
```
In [45]: import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
plt.rcParams['figure.figsize']=10,5
```

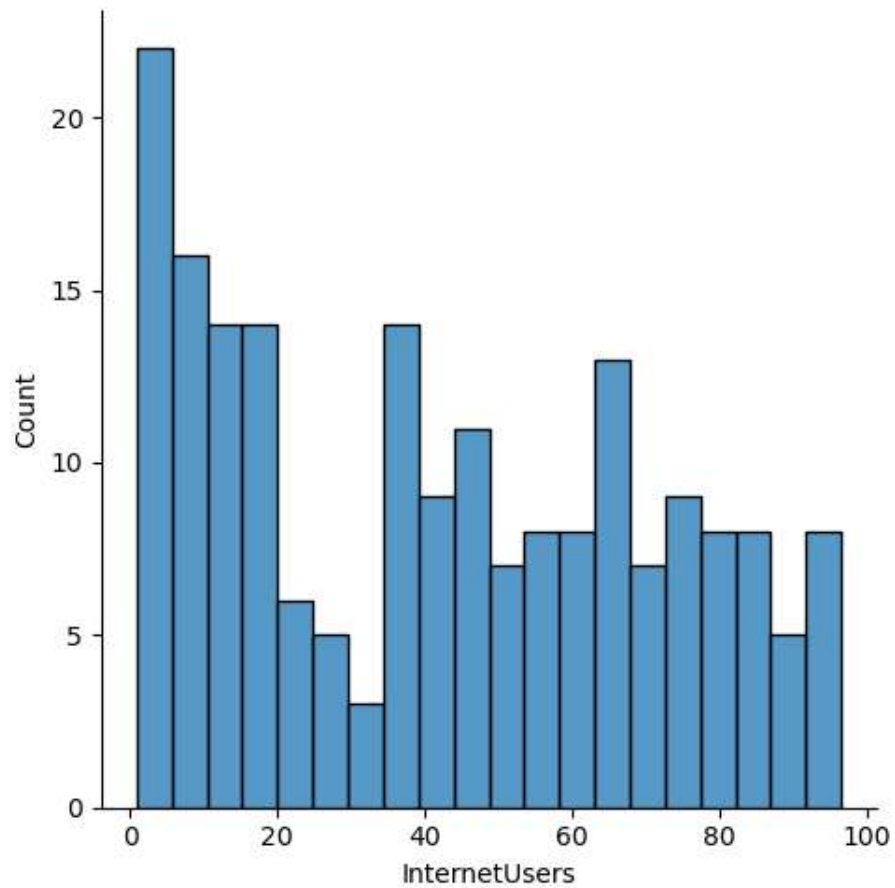
```
In [46]: data['InternetUsers']
```

```
Out[46]: 0      78.9
1       5.9
2      19.1
3      57.2
4      88.0
...
190    20.0
191    46.5
192     2.2
193    15.4
194    18.5
Name: InternetUsers, Length: 195, dtype: float64
```

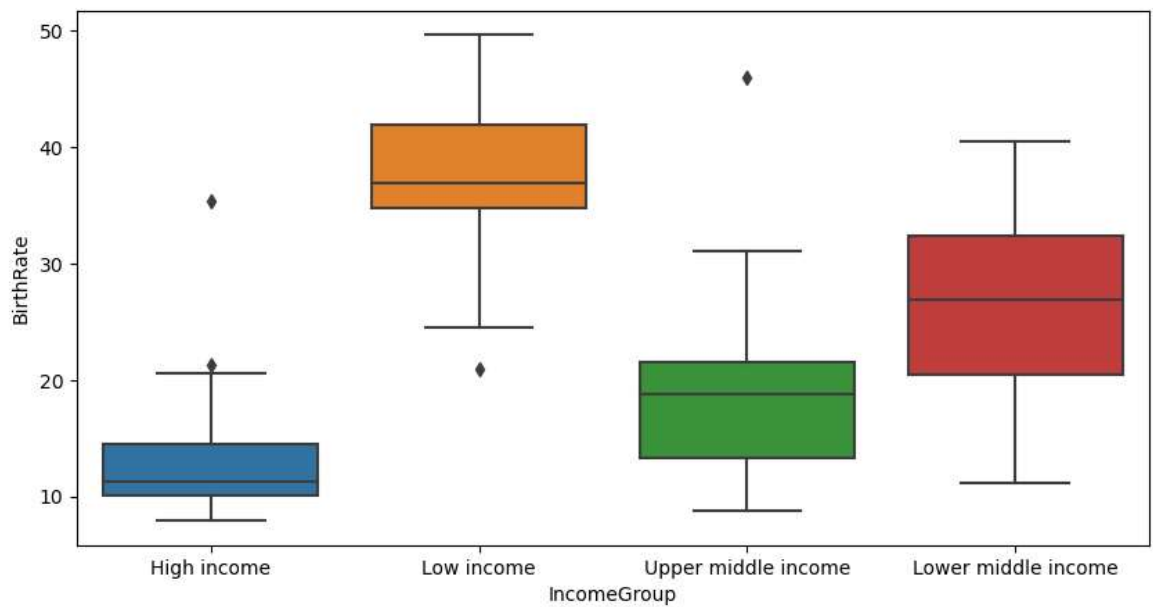
```
In [47]: vis1= sns.displot(data['InternetUsers']) # univariat Analysis
```



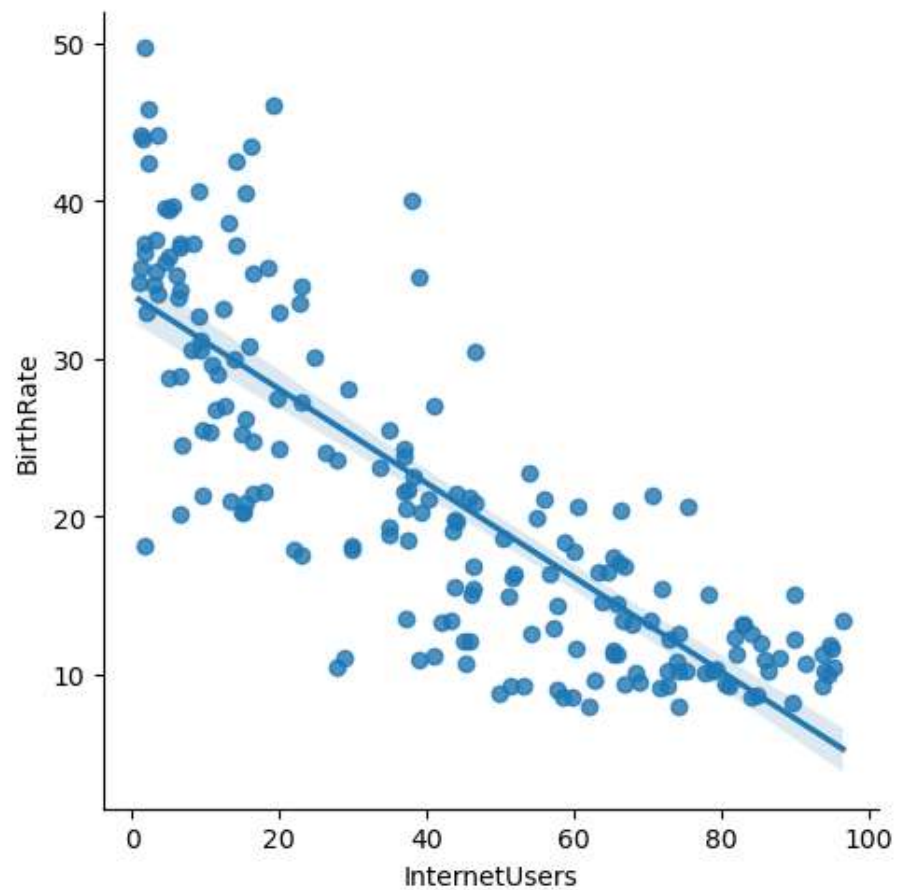
```
In [48]: vis1=sns.displot(data['InternetUsers'],bins=20) # bins: The number of bins to use
```



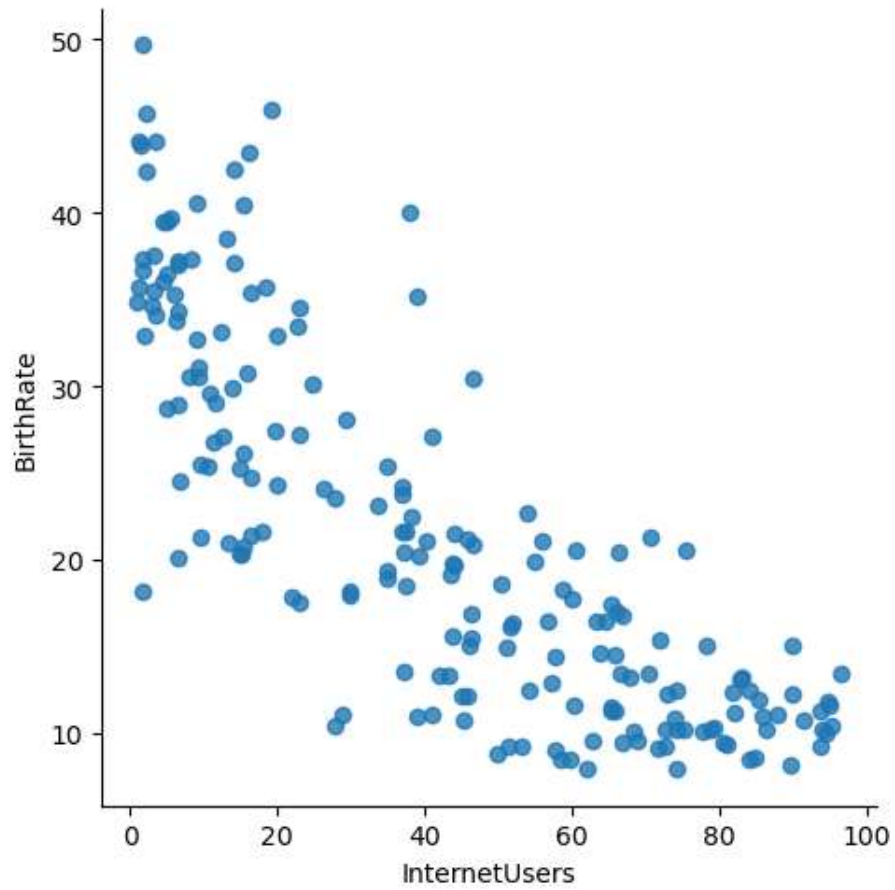
```
In [49]: vis2= sns.boxplot(data,x='IncomeGroup',y='BirthRate') # Boxplot -bi variate analy
```



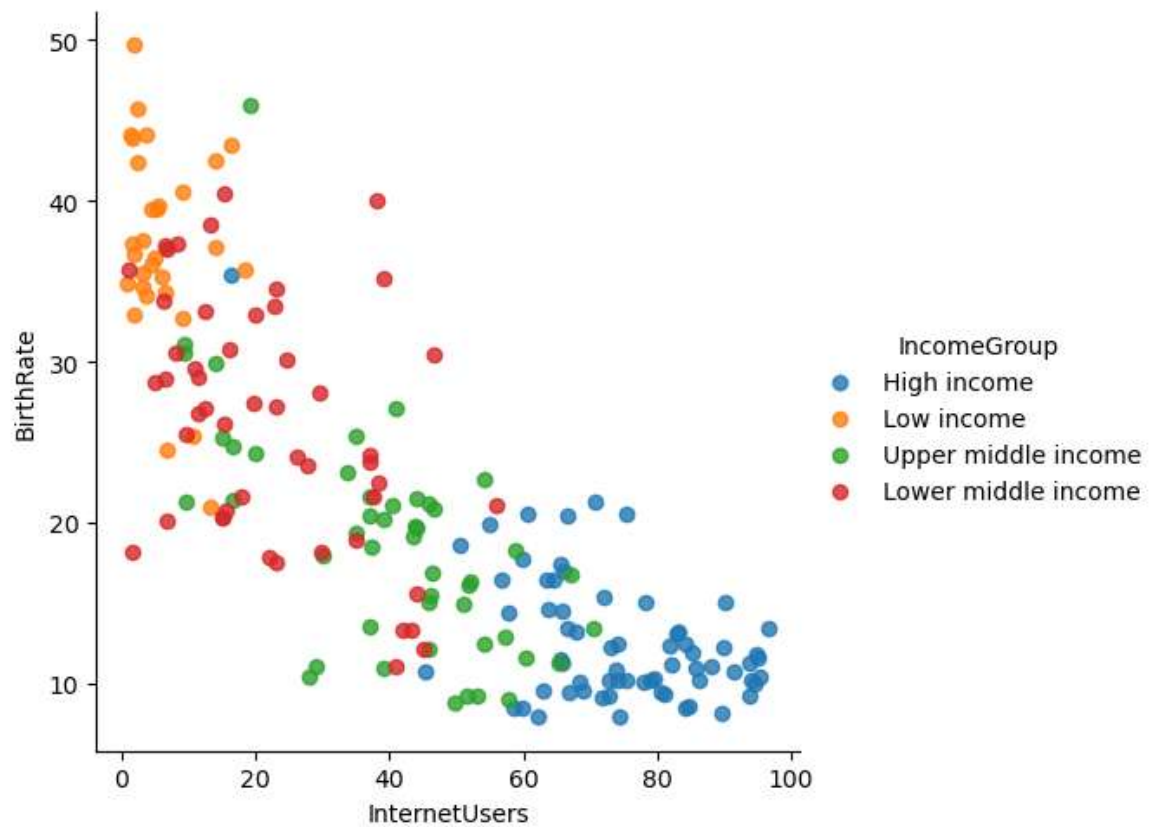
```
In [55]: vis3=sns.lmplot(data, x='InternetUsers', y='BirthRate')  
         # Linear regression plot of the InternetUsers column and the BirthRate
```



```
In [56]: vis4=sns.lmplot(data, x = 'InternetUsers', y = 'BirthRate', fit_reg = False )  
# fit_reg argument is set to False, which means that the regression line will not
```



```
In [57]: vis5 = sns.lmplot(data, x = 'InternetUsers', y = 'BirthRate',  
                           fit_reg = False, hue = 'IncomeGroup')  
# The hue argument is set to 'IncomeGroup', which means that the points in the plot  
# colored differently according to the income group.
```



In []: