

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

```
In [2]: df=pd.read_csv(r"C:\Users\Arabinda\Downloads\data.csv")
df.head(5)
```

Out[2]:

	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 [3]: df.tail(2)
```

Out[3]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

```
In [4]: df
```

Out[4]:

	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 [5]: df.iloc[2]
```

```
Out[5]: CountryName          Angola
         CountryCode           AGO
         BirthRate            45.985
         InternetUsers        19.1
         IncomeGroup          Upper middle income
         Name: 2, dtype: object
```

```
In [6]: df.max()
```

```
Out[6]: CountryName          Zimbabwe
         CountryCode           ZWE
         BirthRate            49.661
         InternetUsers        96.5468
         IncomeGroup          Upper middle income
         dtype: object
```

```
In [7]: df[:::-1]
```

	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 [8]: df[:5]
```

	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 [9]: `df[0:200:10]`

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.900000	High income
10	Azerbaijan	AZE	18.300	58.700000	Upper middle income
20	Belarus	BLR	12.500	54.170000	Upper middle income
30	Canada	CAN	10.900	85.800000	High income
40	Costa Rica	CRI	15.022	45.960000	Upper middle income
50	Ecuador	ECU	21.070	40.353684	Upper middle income
60	Gabon	GAB	30.555	9.200000	Upper middle income
70	Greenland	GRL	14.500	65.800000	High income
80	India	IND	20.291	15.100000	Lower middle income
90	Kazakhstan	KAZ	22.730	54.000000	Upper middle income
100	Libya	LBY	21.425	16.500000	Upper middle income
110	Moldova	MDA	12.141	45.000000	Lower middle income
120	Mozambique	MOZ	39.705	5.400000	Low income
130	Netherlands	NLD	10.200	93.956400	High income
140	Poland	POL	9.600	62.849200	High income
150	Sudan	SDN	33.477	22.700000	Lower middle income
160	Suriname	SUR	18.455	37.400000	Upper middle income
170	Tajikistan	TJK	30.792	16.000000	Lower middle income
180	Uruguay	URY	14.374	57.690000	High income
190	Yemen, Rep.	YEM	32.947	20.000000	Lower middle income

In [10]: `df.describe() # it gives numerical staticstical description`

Out[10]:

	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 [11]: df.describe().transpose() #row-->column, column-->row

Out[11]:

	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 [12]: df.describe().T #row-->column, column-->row

Out[12]:

	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 [13]: df.columns

Out[13]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'IncomeGroup'],
dtype='object')

In [14]: df.columns=["a","b","c","d","e"] #Rename the column name

In [15]: df.head()

Out[15]:

	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

In [16]: df.columns=['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'IncomeGroup']

In [17]: df.head()

	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 [18]: df.dtypes #show the date types

```

Out[18]: CountryName    object
         CountryCode   object
         BirthRate     float64
         InternetUsers float64
         IncomeGroup   object
         dtype: object
  
```

In [19]: df['CountryName'].head()

```

Out[19]: 0      Aruba
         1  Afghanistan
         2      Angola
         3     Albania
         4 United Arab Emirates
         Name: CountryName, dtype: object
  
```

In [20]: df.columns

```

Out[20]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
               'IncomeGroup'],
               dtype='object')
  
```

In [21]: df[['CountryName', 'BirthRate']]

Out[21]:

	CountryName	BirthRate
0	Aruba	10.244
1	Afghanistan	35.253
2	Angola	45.985
3	Albania	12.877
4	United Arab Emirates	11.044
...
190	Yemen, Rep.	32.947
191	South Africa	20.850
192	Congo, Dem. Rep.	42.394
193	Zambia	40.471
194	Zimbabwe	35.715

195 rows × 2 columns

In [22]:

```
df1=df[0:4][['CountryName','BirthRate']]
print(df1)
```

	CountryName	BirthRate
0	Aruba	10.244
1	Afghanistan	35.253
2	Angola	45.985
3	Albania	12.877

In [23]:

```
df2=df[['CountryName','BirthRate']][0:4]
print(df2)
```

	CountryName	BirthRate
0	Aruba	10.244
1	Afghanistan	35.253
2	Angola	45.985
3	Albania	12.877

In [24]:

```
df3=df.BirthRate*df.InternetUsers # Calculate the multiplication
df3
```

```
Out[24]: 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 [25]: df['BirthRate_InternetUser']=df3 #We can add a column in that table
```

```
In [26]: df
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	BirthRate_Intern
0	Aruba	ABW	10.244	78.9	High income	80
1	Afghanistan	AFG	35.253	5.9	Low income	20
2	Angola	AGO	45.985	19.1	Upper middle income	87
3	Albania	ALB	12.877	57.2	Upper middle income	73
4	United Arab Emirates	ARE	11.044	88.0	High income	97
...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income	65
191	South Africa	ZAF	20.850	46.5	Upper middle income	96
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	9
193	Zambia	ZMB	40.471	15.4	Lower middle income	62
194	Zimbabwe	ZWE	35.715	18.5	Low income	66

195 rows × 6 columns



```
In [27]: df.head()
```

Out[27]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	BirthRate_ Internet
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



In [28]: `df.drop('BirthRate_ InternetUser', axis=1)`

Out[28]:

	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 [29]: `df.columns[2]`

Out[29]: 'BirthRate'

In [55]: `Filter=df.InternetUsers<2 # According to this condition it print true and false`
`Filter`

```
Out[55]: 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 [31]: df.InternetUsers[df.InternetUsers>78] #print the values those are >78
```

```
Out[31]: 0      78.90000
         4      88.00000
         8      83.00000
         9      80.61880
        12      82.17020
        17      90.00004
        22      95.30000
        30      85.80000
        31      86.34000
        45      84.17000
        47      94.62970
        54      79.40000
        56      91.51440
        58      81.91980
        61      89.84410
        81      78.24770
        84      96.54680
        89      89.71000
        95      84.77000
       102      93.80000
       106      93.77650
       130      93.95640
       131      95.05340
       133      82.78000
       145      85.30000
       152      81.00000
       163      94.78360
       181      84.20000
Name: InternetUsers, dtype: float64
```

```
In [32]: df.InternetUsers[df.InternetUsers<78]
```

```
Out[32]: 1      5.9
         2     19.1
         3     57.2
         5     59.9
         6     41.9
         ...
        190    20.0
        191    46.5
        192     2.2
        193    15.4
        194    18.5
Name: InternetUsers, Length: 167, dtype: float64
```

```
In [33]: Filter
```

```
Out[33]: 0      True
         1     False
         2     False
         3     False
         4      True
         ...
        190    False
        191    False
        192    False
        193    False
        194    False
Name: InternetUsers, Length: 195, dtype: bool
```

```
In [51]: len(Filter)
```

```
Out[51]: 195
```

```
In [37]: df[Filter]
```

Out[37]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	BirthRate_Intern
0	Aruba	ABW	10.244	78.90000	High income	808.2
4	United Arab Emirates	ARE	11.044	88.00000	High income	971.8
8	Australia	AUS	13.200	83.00000	High income	1095.6
9	Austria	AUT	9.400	80.61880	High income	757.8
12	Belgium	BEL	11.200	82.17020	High income	920.3
17	Bahrain	BHR	15.040	90.00004	High income	1353.6
22	Bermuda	BMU	10.400	95.30000	High income	991.1
30	Canada	CAN	10.900	85.80000	High income	935.2
31	Switzerland	CHE	10.200	86.34000	High income	880.6
45	Germany	DEU	8.500	84.17000	High income	715.4
47	Denmark	DNK	10.000	94.62970	High income	946.2
54	Estonia	EST	10.300	79.40000	High income	817.8
56	Finland	FIN	10.700	91.51440	High income	979.2
58	France	FRA	12.300	81.91980	High income	1007.6
61	United Kingdom	GBR	12.200	89.84410	High income	1096.0
81	Ireland	IRL	15.000	78.24770	High income	1173.7
84	Iceland	ISL	13.400	96.54680	High income	1293.7
89	Japan	JPN	8.200	89.71000	High income	735.6
95	Korea, Rep.	KOR	8.600	84.77000	High income	729.0
102	Liechtenstein	LIE	9.200	93.80000	High income	862.9
106	Luxembourg	LUX	11.300	93.77650	High income	1059.6
130	Netherlands	NLD	10.200	93.95640	High income	958.3
131	Norway	NOR	11.600	95.05340	High income	1102.6
133	New Zealand	NZL	13.120	82.78000	High income	1086.0
145	Qatar	QAT	11.940	85.30000	High income	1018.4
152	Singapore	SGP	9.300	81.00000	High income	753.3
163	Sweden	SWE	11.800	94.78360	High income	1118.4
181	United States	USA	12.500	84.20000	High income	1052.5



```
In [39]: df.BirthRate>40 #Those country birthRate above 40 returns true rest are false
```

```
Out[39]: 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 [41]: Filter2=df.BirthRate>40
```

```
Filter2
```

```
Out[41]: 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 [43]: df[Filter2]
```

Out[43]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	BirthRate_Intern
2	Angola	AGO	45.985	19.1	Upper middle income	87
11	Burundi	BDI	44.151	1.3	Low income	5
14	Burkina Faso	BFA	40.551	9.1	Low income	36
65	Gambia, The	GMB	42.525	14.0	Low income	59
115	Mali	MLI	44.138	3.5	Low income	15
127	Niger	NER	49.661	1.7	Low income	8
128	Nigeria	NGA	40.045	38.0	Lower middle income	152
156	Somalia	SOM	43.891	1.5	Low income	6
167	Chad	TCD	45.745	2.3	Low income	10
178	Uganda	UGA	43.474	16.2	Low income	70
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	9
193	Zambia	ZMB	40.471	15.4	Lower middle income	62



In [45]: Filter&Filter2

```
Out[45]: 0    False
1    False
2    False
3    False
4    False
...
190   False
191   False
192   False
193   False
194   False
Length: 195, dtype: bool
```

In [57]: df[Filter&Filter2]

Out[57]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	BirthRate_Intern
11	Burundi	BDI	44.151	1.3	Low income	5
127	Niger	NER	49.661	1.7	Low income	8
156	Somalia	SOM	43.891	1.5	Low income	6



In [59]: df.IncomeGroup.unique()

Out[59]: array(['High income', 'Low income', 'Upper middle income', 'Lower middle income'], dtype=object)

In [63]: import matplotlib.pyplot as plt
import seaborn as sns%matplotlib inline
plt.rcParams['figure.figsize'] = 8,4

#import warnings
#warnings.filterwarnings('ignore')

UsageError: Line magic function `%%matplotlib` not found.

In [65]: df.head()

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	BirthRate_Internet
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

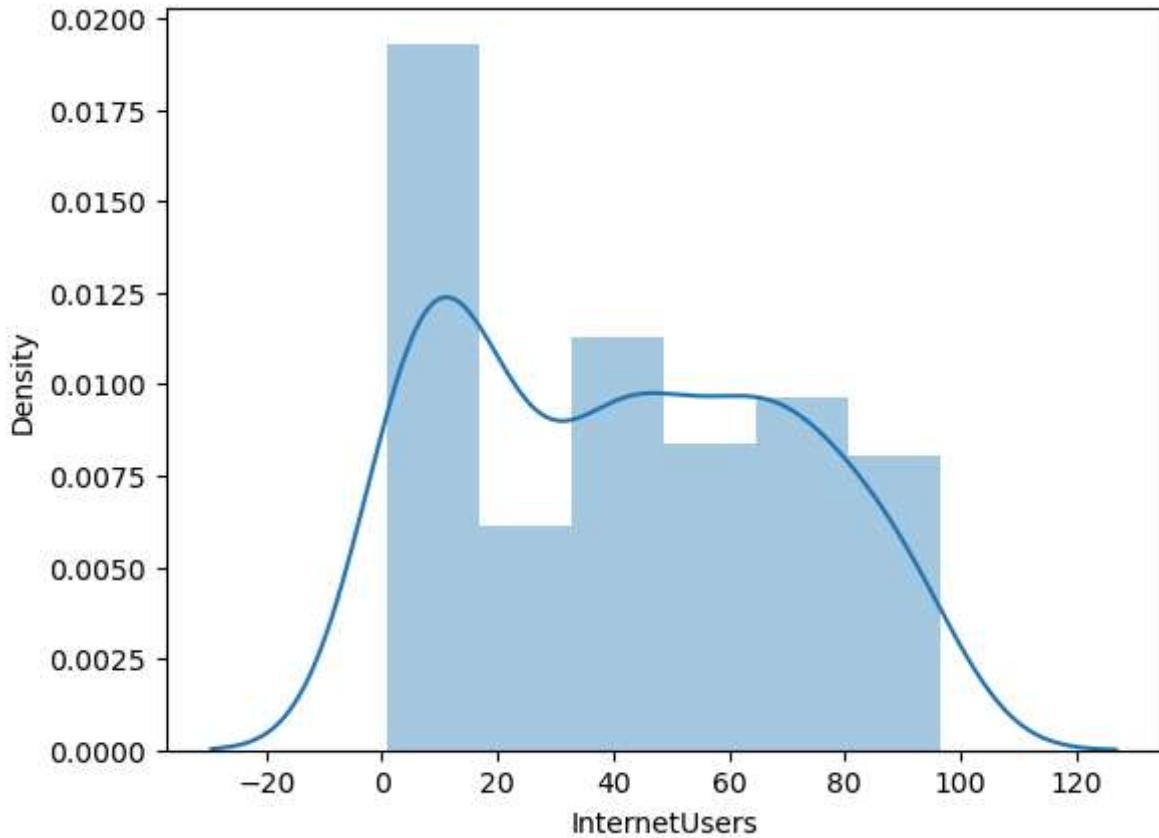


In [69]: # Distributions:

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

```
C:\Users\Arabinda\AppData\Local\Temp\ipykernel_6700\1307174048.py:2: 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
```

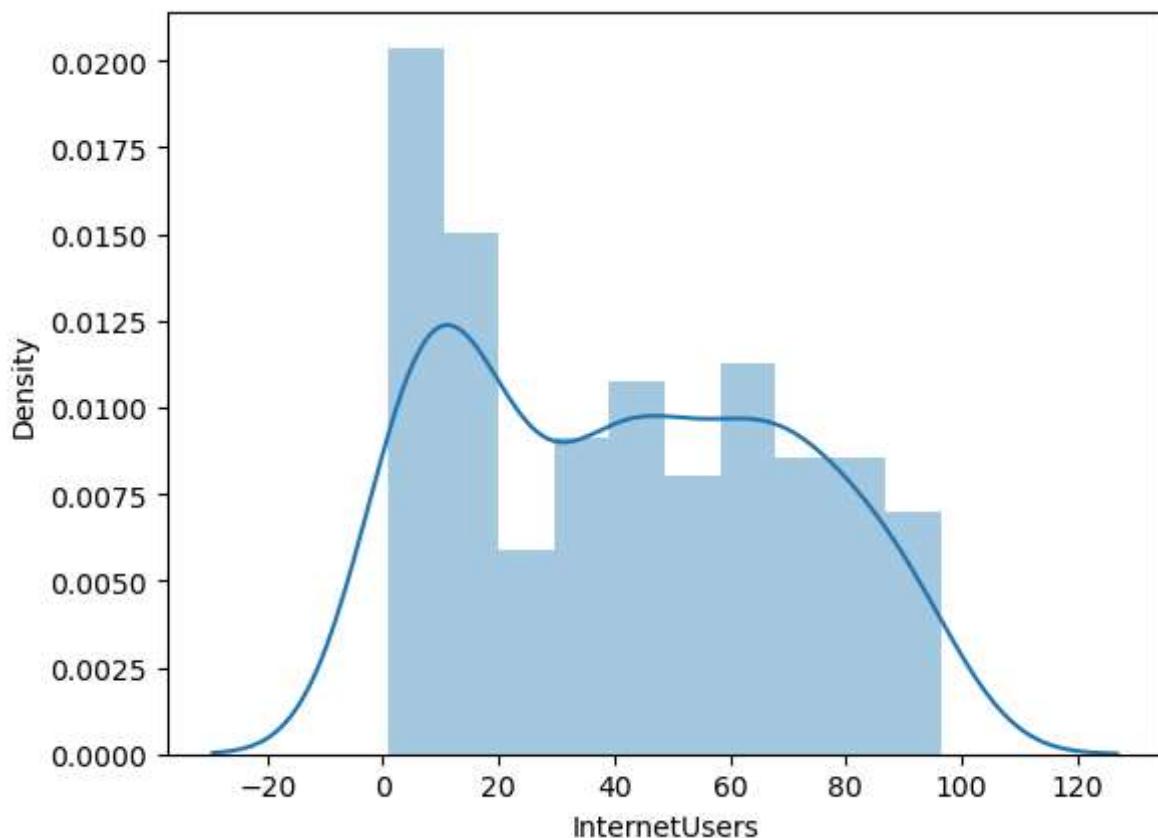
```
vis1 = sns.distplot(df["InternetUsers"])
```



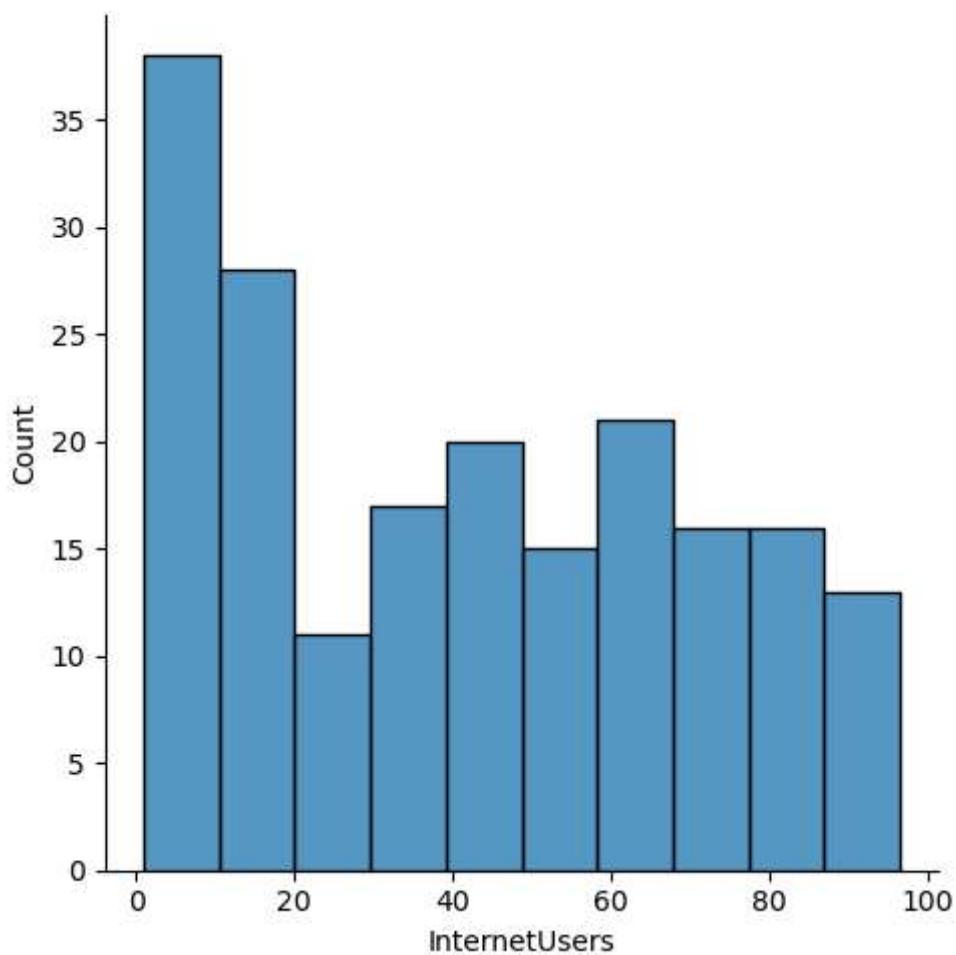
```
In [71]: vis1 = sns.distplot(df["InternetUsers"], bins=10)
```

```
C:\Users\Arabinda\AppData\Local\Temp\ipykernel_6700\902015660.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
```

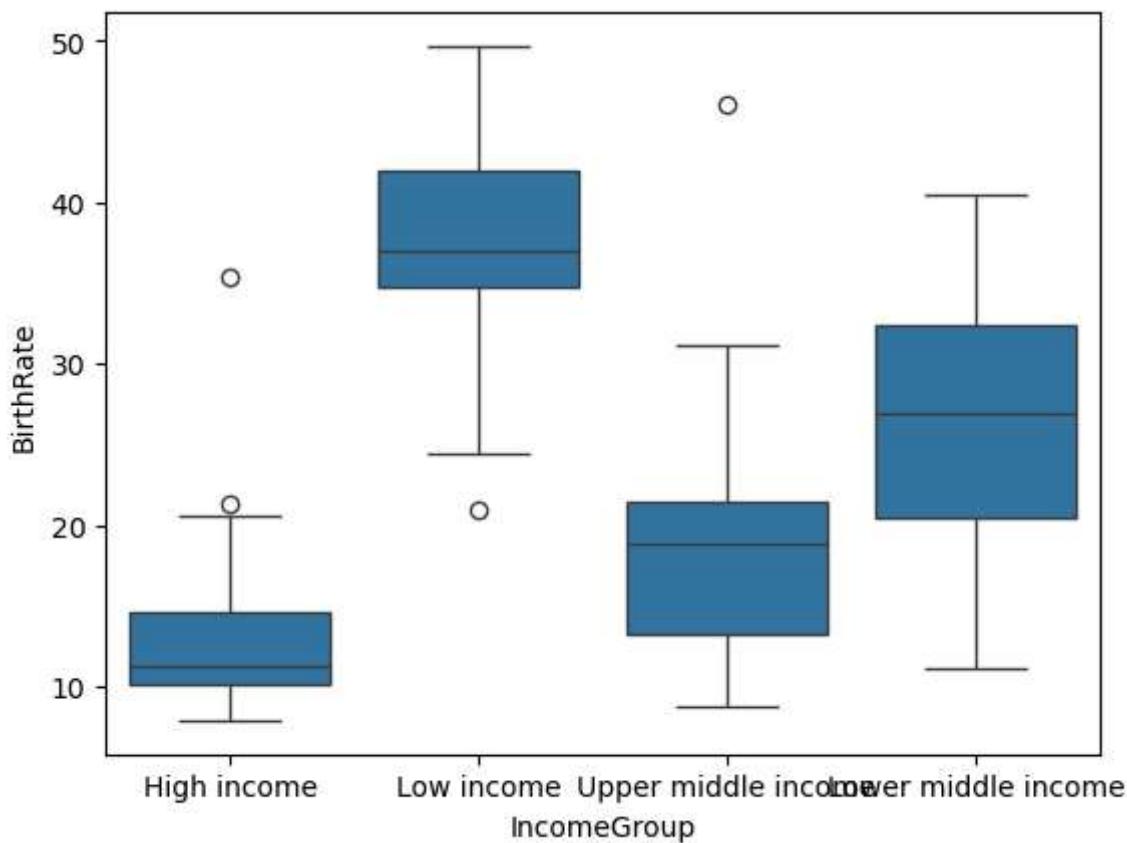
```
vis1 = sns.distplot(df["InternetUsers"], bins=10)
```



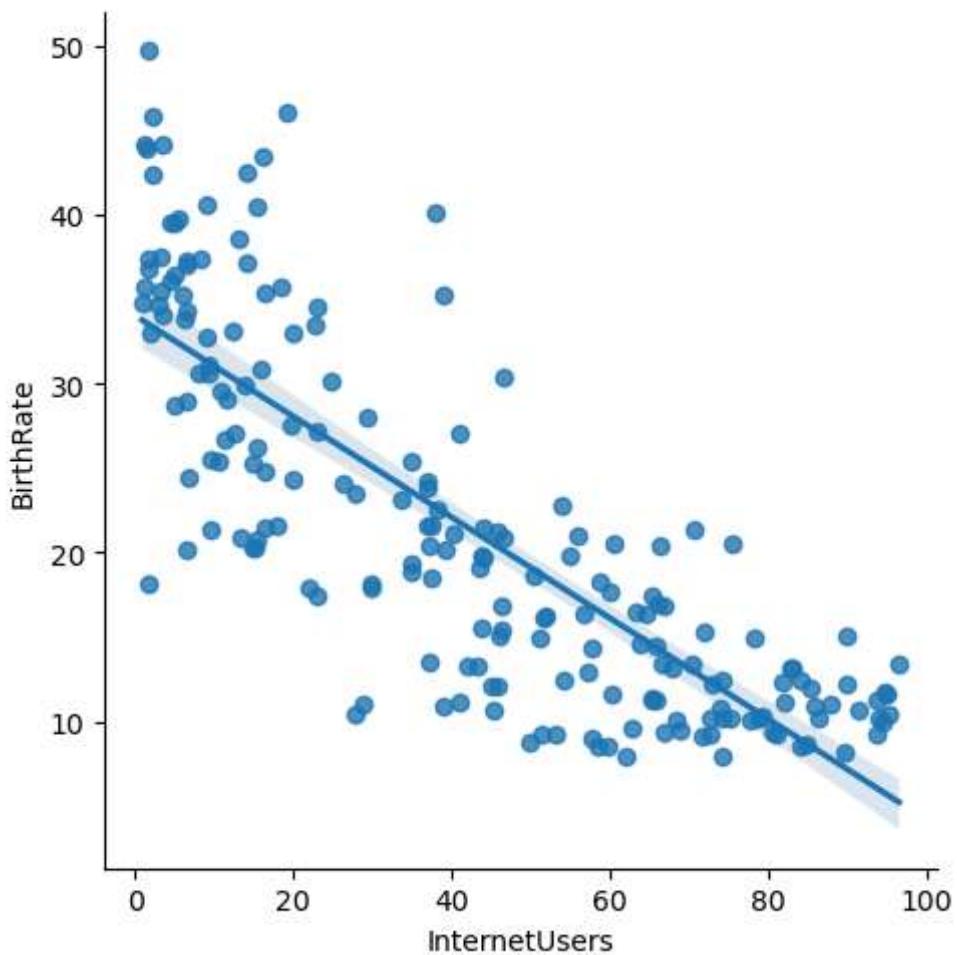
```
In [73]: vis1 = sns.displot(df["InternetUsers"], bins=10)
```



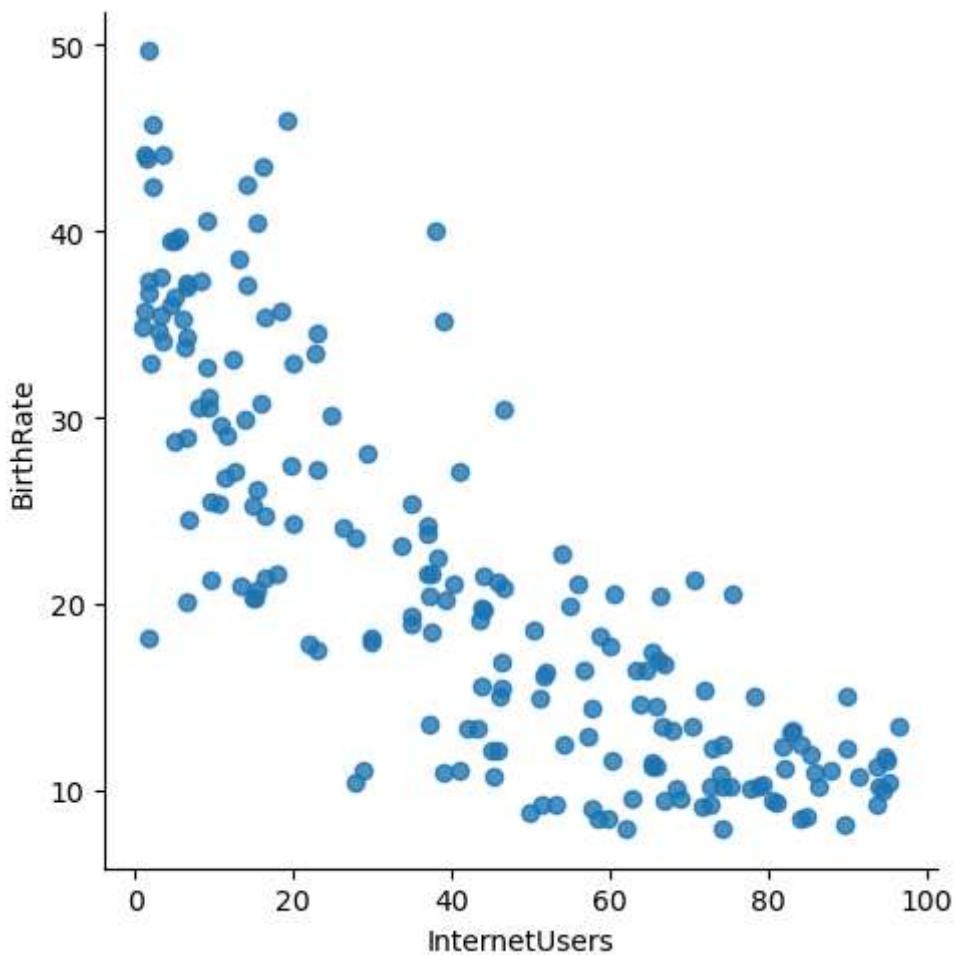
```
In [75]: # Draw a BOX PLOT  
sns2=sns.boxplot(data=df,x="IncomeGroup",y="BirthRate")
```



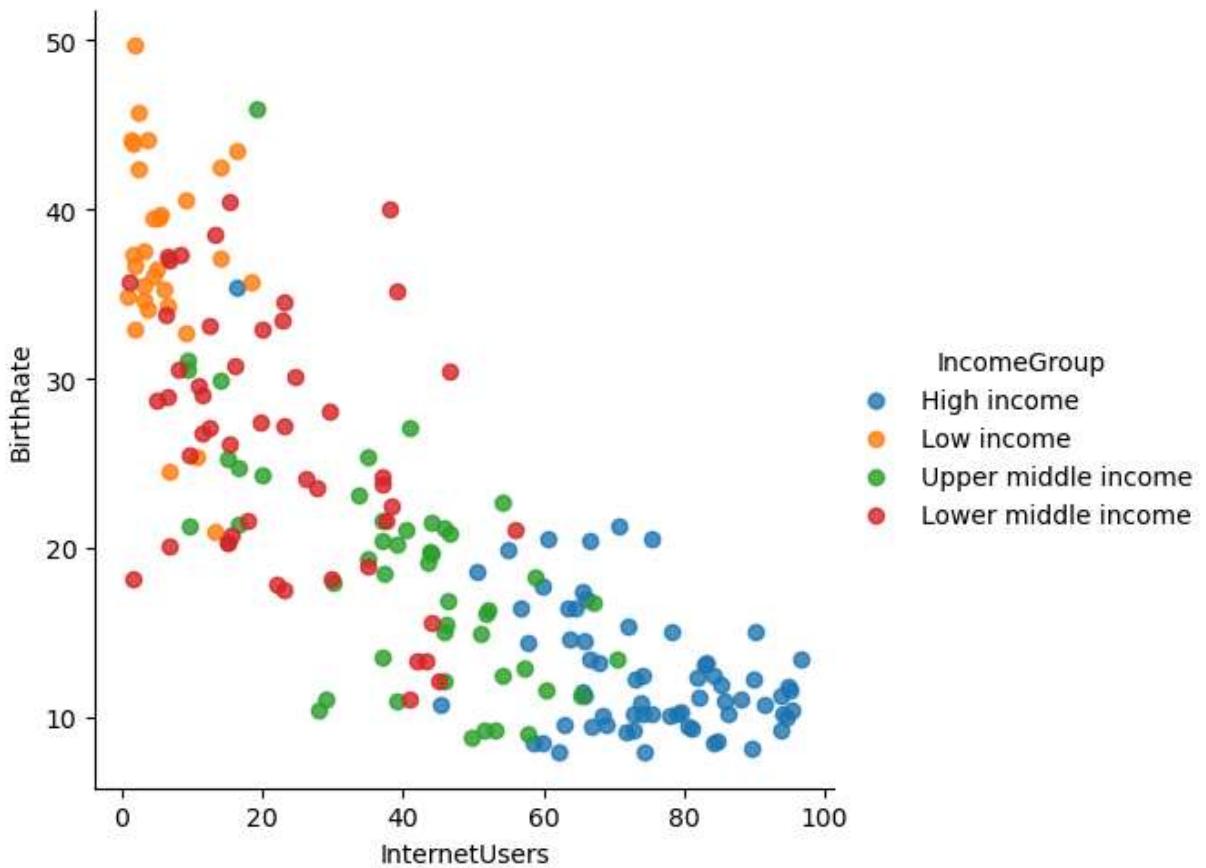
```
In [83]: vis3=sns.lmplot(data=df,x="InternetUsers",y="BirthRate",fit_reg=True)
```



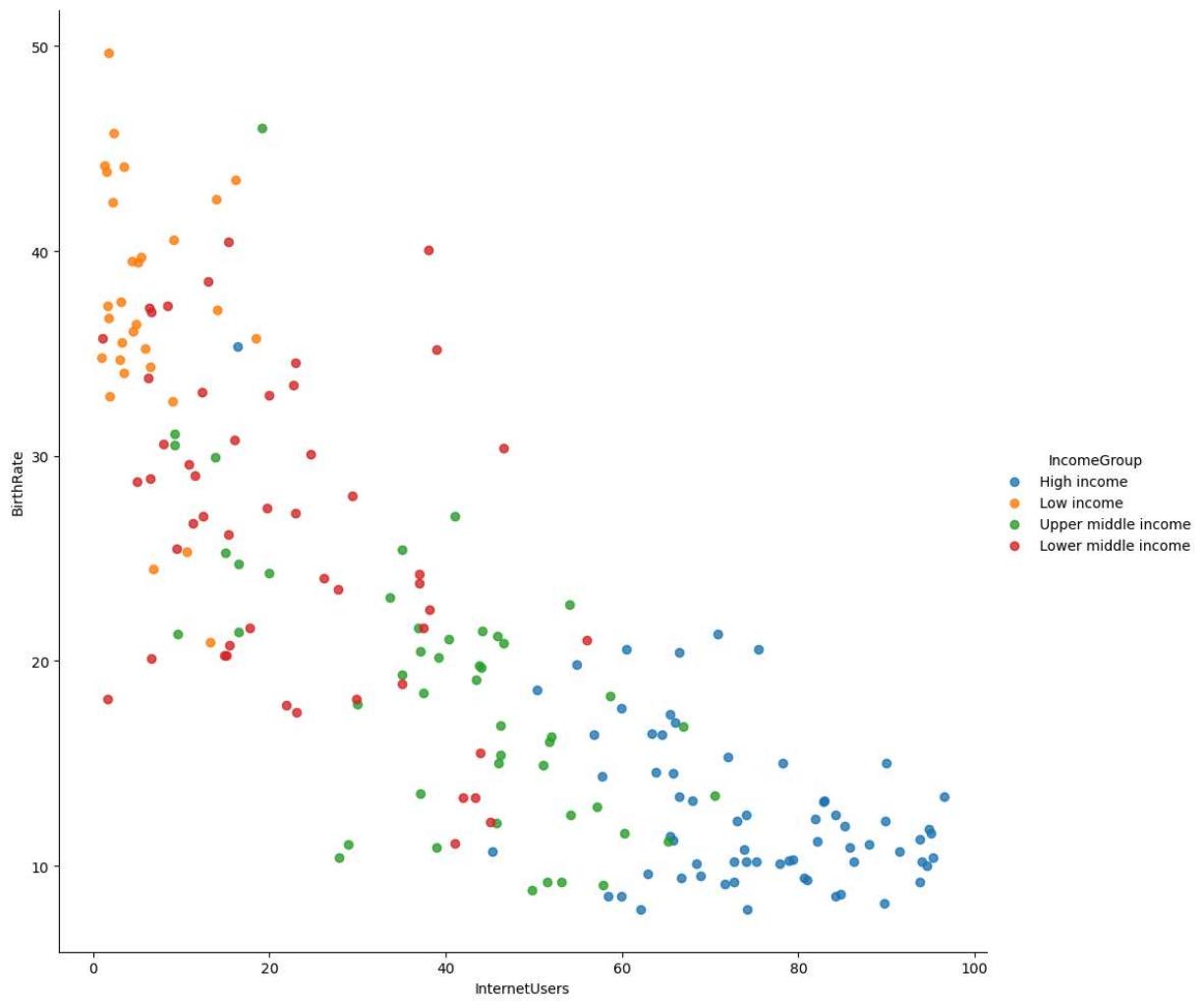
```
In [85]: vis3=sns.lmplot(data=df,x="InternetUsers",y="BirthRate",fit_reg=False)
```



```
In [89]: vis5 = sns.lmplot(data = df,x = 'InternetUsers', y = 'BirthRate',
                      fit_reg = False,hue = 'IncomeGroup')
```



```
In [109]: vis5 = sns.lmplot(data = df ,x = 'InternetUsers' , y = 'BirthRate',
                      fit_reg = False,hue = 'IncomeGroup',height=10)
```



In []: