

```
In [71]: import pandas as pd
```

```
In [3]: stats = pd.read_excel(r'/Users/mahidharreddy/Downloads/data.xlsx')
```

```
In [4]: stats
```

```
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]: len(stats) #to check the lenght of the data
```

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

```
In [6]: stats.columns #to check the columns (attributes)in the data
```

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

```
In [7]: stats['CountryName'] #to access a particular column
```

```
Out[7]: 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 [8]: stats[['BirthRate', 'InternetUsers']] #to access multiple columns
```

```
Out[8]:
```

	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 [9]: stats.head() #to view first 5 rows of the data
```

```
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]: stats.tail() #to view last 5 rows of the data
```

```
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]: stats.head(2) #to view first 2 rows of the data
```

```
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

```
In [12]: stats.tail(2) #to view last 2 rows of the data
```

```
Out[12]:
```

	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 [13]: stats.isnull() #to check for null values in the data
```

Out[13]:

	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 [14]: `stats.isnull().sum() #to check for null values in each column`

Out[14]:

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

In [15]: `stats.isna() #to check for NaN values in the data`

```
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 x 5 columns

```
In [16]: stats.isna().sum() #alternative to isnull()
```

```
Out[16]: CountryName      0
CountryCode      0
BirthRate        0
InternetUsers     0
IncomeGroup      0
dtype: int64
```

```
In [17]: stats.dtypes #to check the data types of each column
```

```
Out[17]: CountryName      object
CountryCode      object
BirthRate        float64
InternetUsers     float64
IncomeGroup      object
dtype: object
```

```
In [18]: stats.info() #to get a summary of the data
```

```
<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 [19]: stats[:, :] #to access all rows and columns
```

```
Out[19]:
```

	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

Split data

```
In [20]: stats_numerical_data = stats[['BirthRate', 'InternetUsers']]
stats_numerical_data
```

Out[20]:

	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 [21]: stats_categorical_data = stats[['CountryName', 'CountryCode', 'IncomeGroup']]
stats_categorical_data
```

Out[21]:

	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 [22]: print(stats.shape)
print(stats_numerical_data.shape)
print(stats_categorical_data.shape)
```

(195, 5)

(195, 2)

(195, 3)

In [23]: stats[:]

Out[23]:

	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 x 5 columns

In [24]: stats[3:]

Out[24]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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
6	Armenia	ARM	13.308	41.9	Lower middle income
7	Antigua and Barbuda	ATG	16.447	63.4	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

192 rows × 5 columns

In [25]: stats[:6]

Out[25]:

	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 [27]: stats.describe() *#to get statistical summary of numerical columns*

Out[27]:

	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 [28]: stats.describe().transpose() *#to get statistical summary in transpo*

Out[28]:

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

In []: stats_categorical_data.describe() *#to get statistical summary of*

Out[]:

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

In [31]: stats_categorical_data.describe().transpose()

Out[31]:

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

In [32]: stats.head(2)

Out[32]:

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

```
In [33]: stats['BirthRate'] * stats['InternetUsers'] #to multiply two colum
```

```
Out[33]: 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 [34]: stats['mycalc'] = stats['BirthRate'] * stats['InternetUsers'] #to
```

```
In [35]: stats.head(2)
```

```
Out[35]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	m
0	Aruba	ABW	10.244	78.9	High income	808
1	Afghanistan	AFG	35.253	5.9	Low income	207

```
In [36]: len(stats.columns)
```

```
Out[36]: 6
```

```
In [37]: stats = stats.drop('mycalc', axis=1) #to drop the newly created co
```

```
In [38]: stats
```

Out[38]:

	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 [39]: stats[stats['InternetUsers'] < 2] *#to filter rows based on a condi*

Out[39]:

	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 []: len(stats[stats['InternetUsers'] < 2])

Out[]: 9

```
In [43]: len(stats[stats['BirthRate'] > 40]) #to filter rows based on another column
```

Out[43]: 12

```
In [45]: stats[(stats.BirthRate > 40) & (stats.InternetUsers < 2)] #to filter rows based on two columns
```

```
Out[45]:
```

	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

```
stats[stats['IncomeGroup'] == 'High income']
```

```
In [47]: stats[stats['IncomeGroup'] == 'High income']
```

```
Out[47]:
```

	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 [ ]: stats[stats['IncomeGroup'] == 'Low income'] #to filter rows based on another column
```

```
Out[ ]:
```

	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 [49]: `stats.IncomeGroup.unique()` *#to view unique values in a categorical*

```
Out[49]: array(['High income', 'Low income', 'Upper middle income',
               'Lower middle income'], dtype=object)
```

```
In [50]: stats.IncomeGroup.nunique() #to view unique values in a categorica
```

```
Out[50]: 4
```

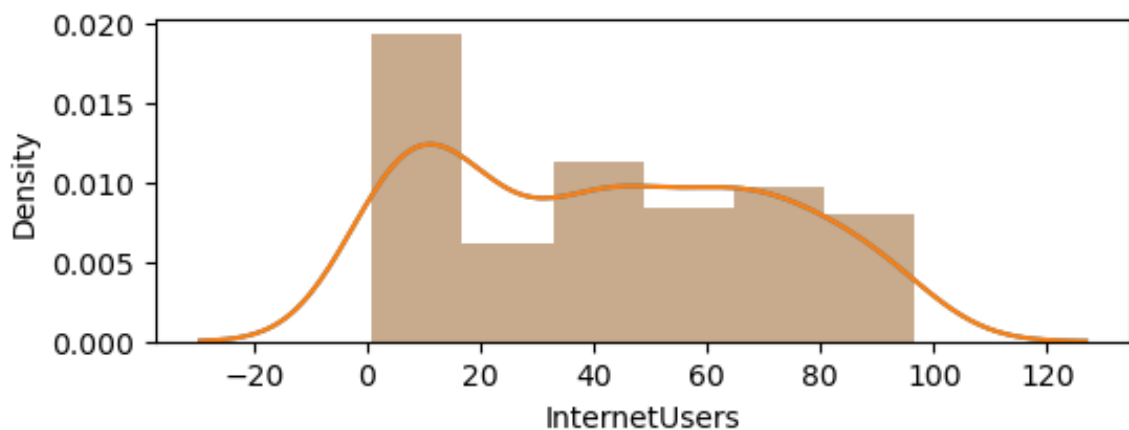
```
In [54]: import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.rcParams['figure.figsize'] = (6, 2)

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

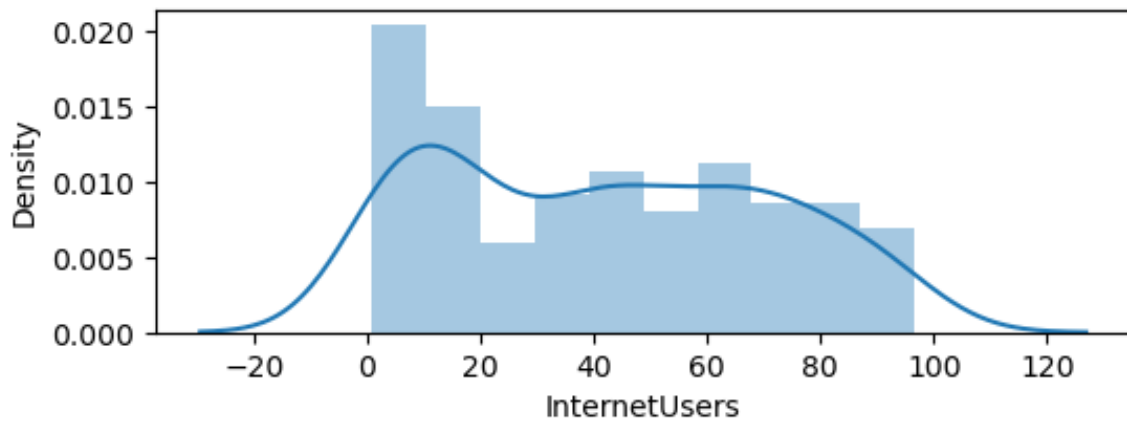
```
In [55]: stats['InternetUsers']
```

```
Out[55]: 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
```

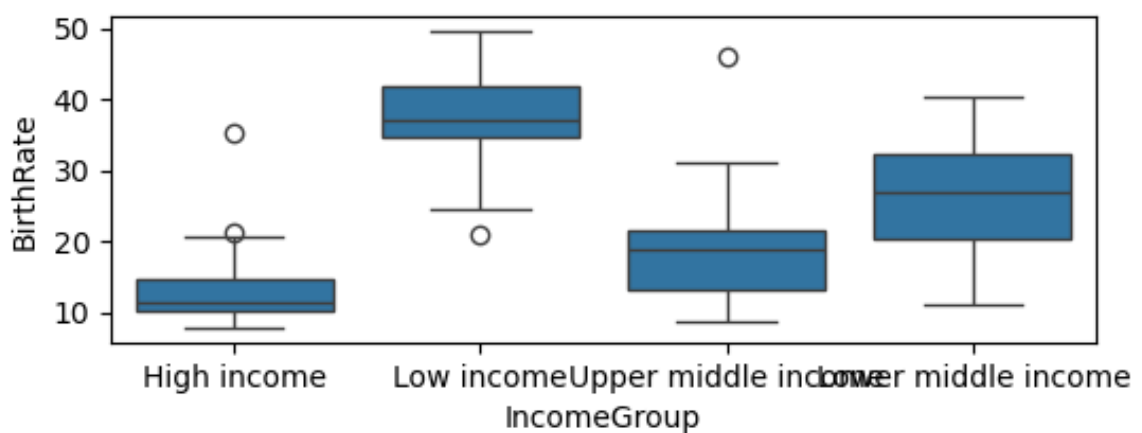
```
In [ ]: vis1 = sns.distplot(stats['InternetUsers']) #to plot the distributi
plt.show() #univariate analysis , plot the graph using one variable
```



```
In [64]: vis2 = sns.distplot(stats['InternetUsers'], bins=10) #to plot hist
plt.show()
```

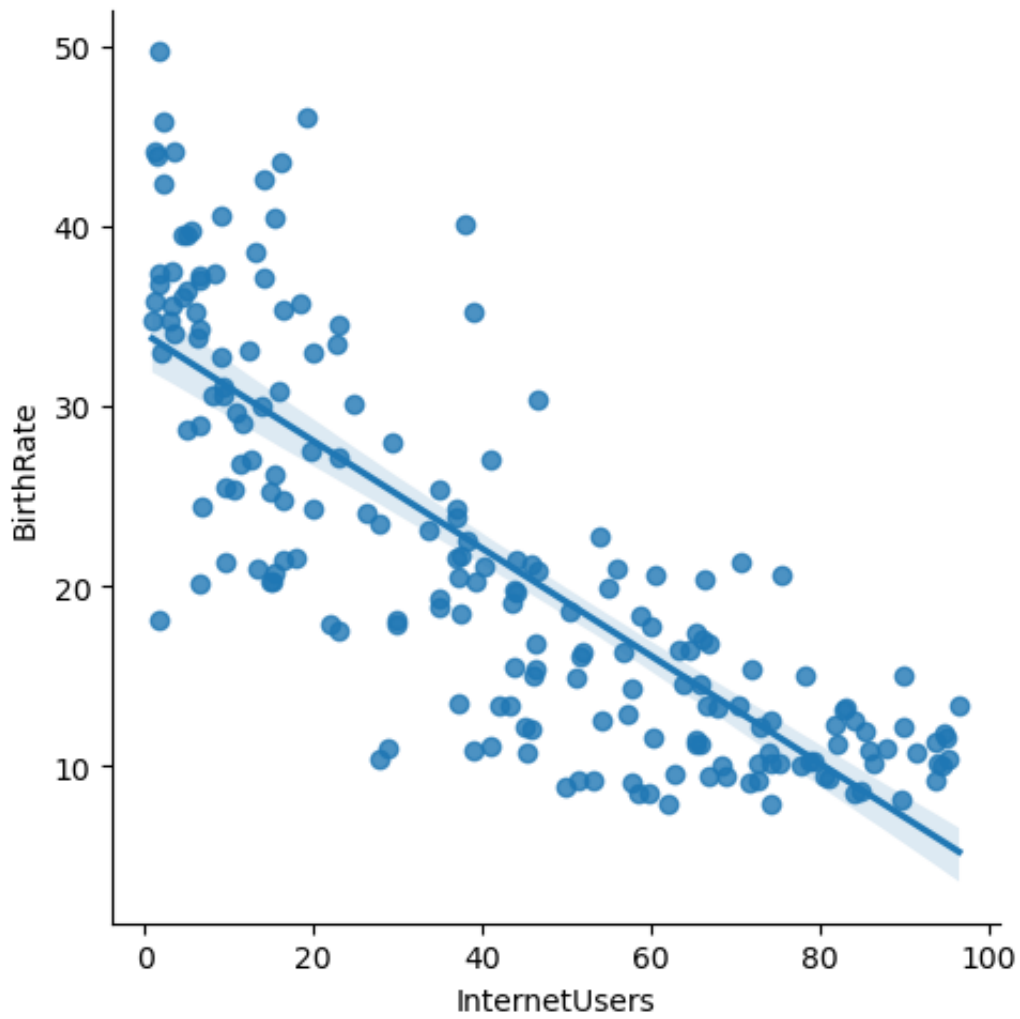


```
In [ ]: vis3 = sns.boxplot(data = stats, x='IncomeGroup', y='BirthRate') #t
plt.show()
```

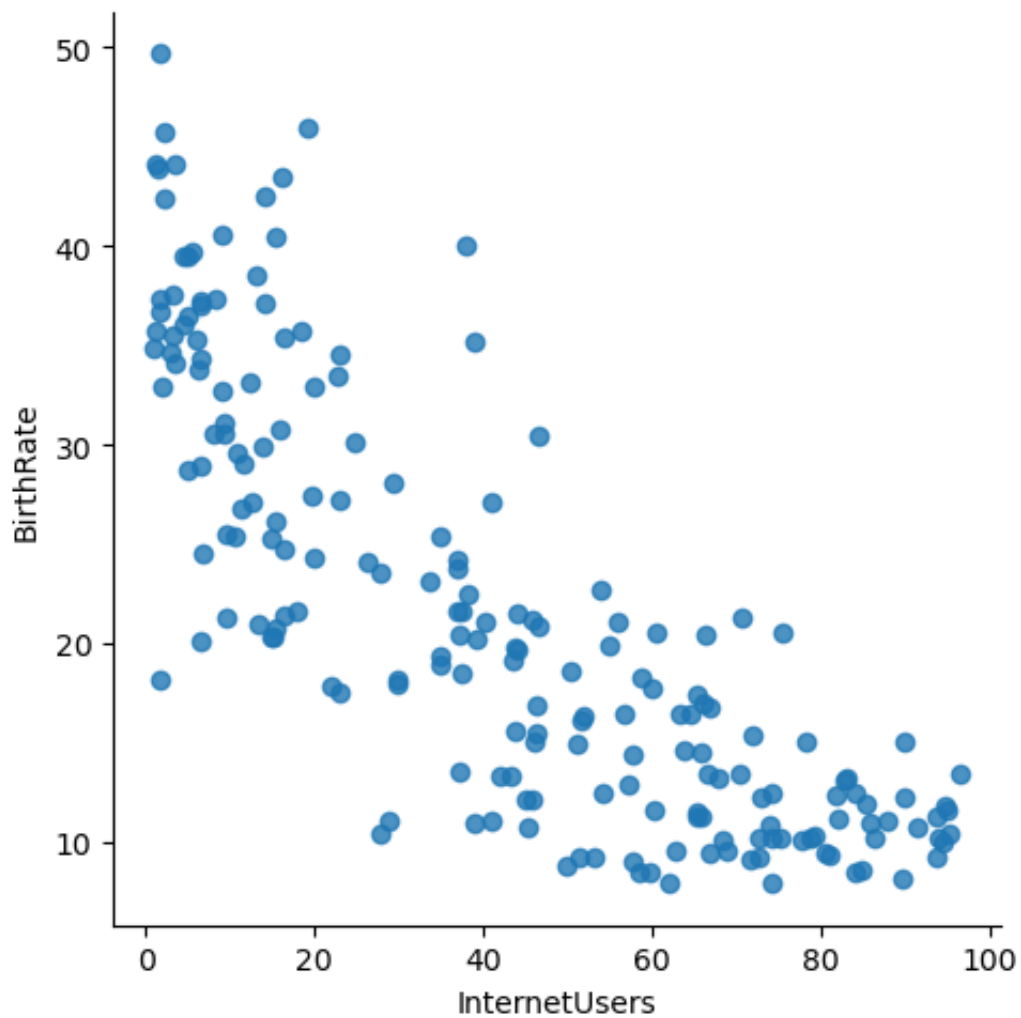


```
vis4 = sns.lmplot(data = stats, x = 'InternetUsers, y = 'BirthRate') plt.show()
```

```
In [66]: vis4 = sns.lmplot(data = stats, x = 'InternetUsers', y = 'BirthRate'
plt.show()
```

```
In [67]: vis4 = sns.lmplot(data = stats, x = 'InternetUsers', y = 'BirthRate',  
plt.show())
```



```
In [70]: vis4 = sns.lmplot(data = stats, x = 'InternetUsers', y = 'BirthRate',  
plt.show())
```

