

Country GDP analysis

```
In [1]: # Importing required package "pandas" to load data
import pandas as pd
```

```
In [2]: # importing data using pandas
df = pd.read_csv('data.csv')
df
```

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
...
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 [3]: # For more clarification view print fist 5 rows
df.head(5)
```

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

```
In [4]: # For identify how many rows of the data print last 5 rows
df.tail(5)
```

```
Out[4]:
```

	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 [5]: len(df)
```

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

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

```
Out[6]: (195, 5)
```

```
In [7]: df.columns
```

```
Out[7]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
              'IncomeGroup'],
              dtype='object')
```

```
In [8]: type(df)
```

```
Out[8]: pandas.core.frame.DataFrame
```

```
In [9]: df.isnull()
```

```
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()
```

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

```
In [11]: 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 [12]: df.describe()
```

```
Out[12]:
```

	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 [13]: df.describe().transpose()
```

```
Out[13]:
```

	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 [14]: df.dtypes
```

```
Out[14]: CountryName      object
CountryCode      object
BirthRate      float64
InternetUsers    float64
IncomeGroup      object
dtype: object
```

```
In [15]: df[::-1] # To print data from bottom to top
```

```
Out[15]:
```

	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 [16]: df_categorical = df[['CountryName', 'CountryCode', 'IncomeGroup']]
df_categorical
```

```
Out[16]:
```

	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 [17]: df['mycalc']=df.BirthRate * df.InternetUsers
df
```

Out[17]:

	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 [18]: df["InternetUsers"]
```

Out[18]:

```
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 [19]: df[['CountryCode', 'BirthRate', 'InternetUsers']][4:18]
```

```
Out[19]:
```

	CountryCode	BirthRate	InternetUsers
4	ARE	11.044	88.00000
5	ARG	17.716	59.90000
6	ARM	13.308	41.90000
7	ATG	16.447	63.40000
8	AUS	13.200	83.00000
9	AUT	9.400	80.61880
10	AZE	18.300	58.70000
11	BDI	44.151	1.30000
12	BEL	11.200	82.17020
13	BEN	36.440	4.90000
14	BFA	40.551	9.10000
15	BGD	20.142	6.63000
16	BGR	9.200	53.06150
17	BHR	15.040	90.00004

```
In [20]: Filter=df.InternetUsers < 2  
Filter
```

```
Out[20]: 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 [21]: df[Filter]
```

```
Out[21]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
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 [22]: len(df[Filter])
```

```
Out[22]: 9
```

```
In [23]: Filter2= df.BirthRate >40  
Filter2
```

```
Out[23]: 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 [24]: df[Filter2]
```

```
Out[24]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
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 [25]: len(df[Filter2])
```

```
Out[25]: 12
```

```
In [26]: Filter & Filter2
```

```
Out[26]: 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 [27]: len(Filter & Filter2)
```

```
Out[27]: 195
```



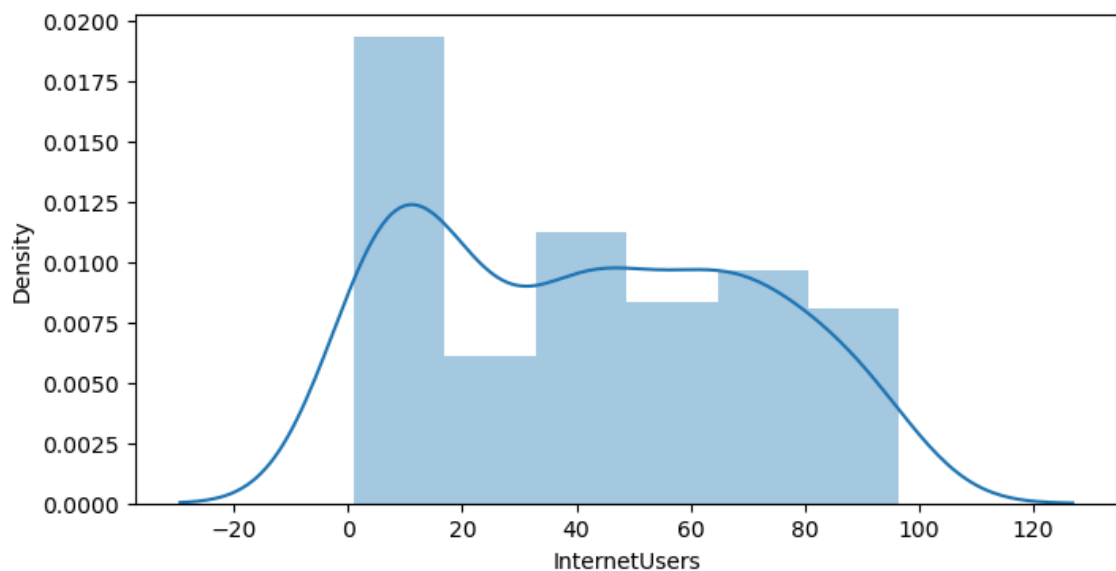
```
In [28]: df[Filter & Filter2]
```

Out[28]:

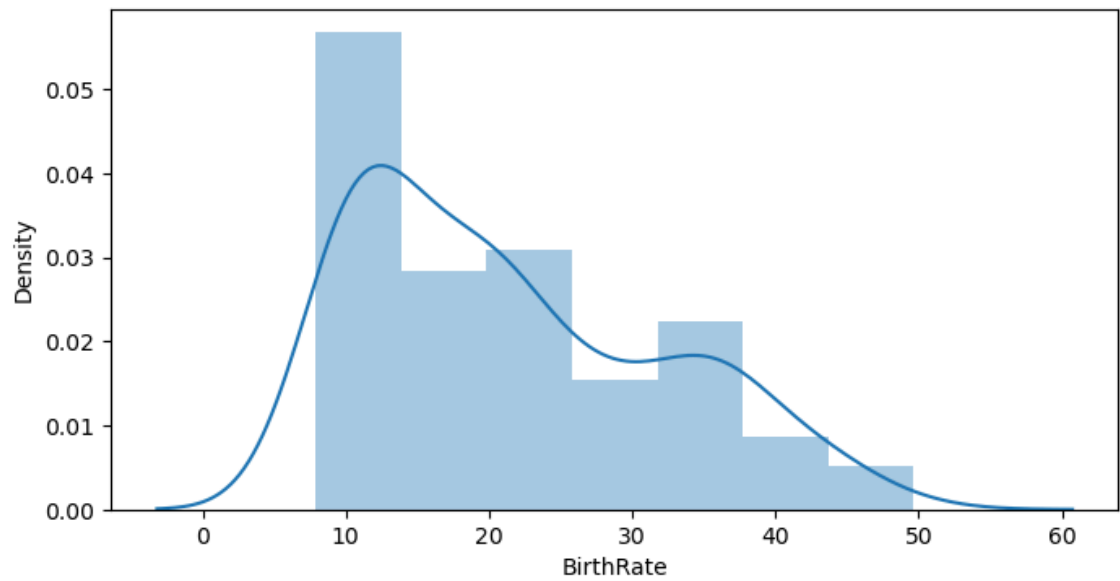
	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
11	Burundi	BDI	44.151	1.3	Low income	57.3963
127	Niger	NER	49.661	1.7	Low income	84.4237
156	Somalia	SOM	43.891	1.5	Low income	65.8365

```
In [29]: # Introduction to seaborn # seaborn is very powerfull visualizatio(STATISTI
import matplotlib.pyplot as plt # visulaiztion
import seaborn as sns # distribution visualtion
# seaborn are used for advance visualization e.x --> distribution plot, lin
%matplotlib inline
plt.rcParams['figure.figsize'] = 8,4
import warnings
warnings.filterwarnings('ignore') # os error
```

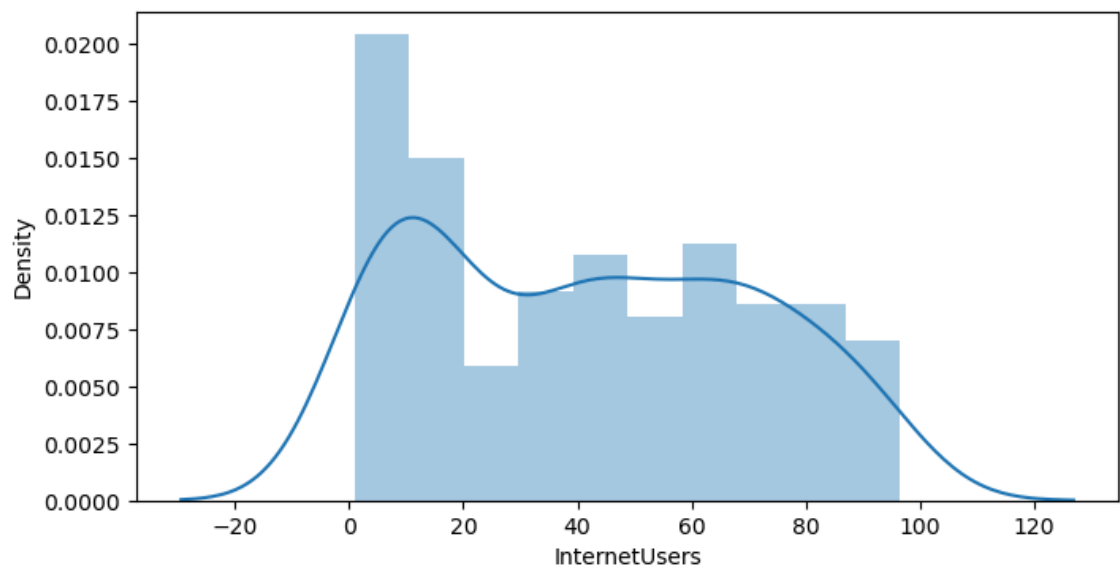
```
In [30]: vis1 = sns.distplot(df["InternetUsers"])
```



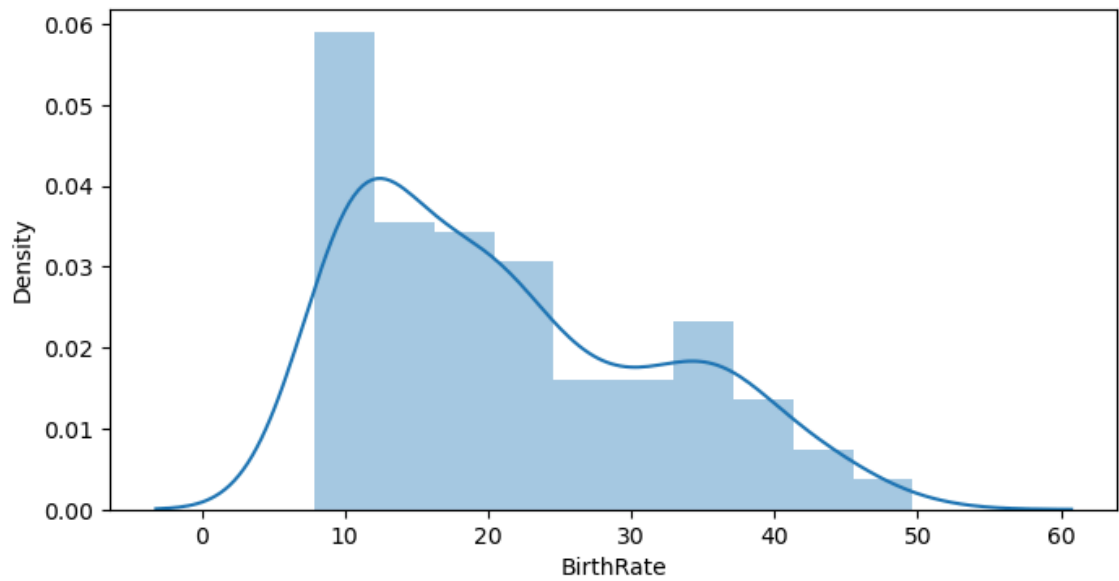
```
In [31]: vis1 = sns.distplot(df["BirthRate"])
```



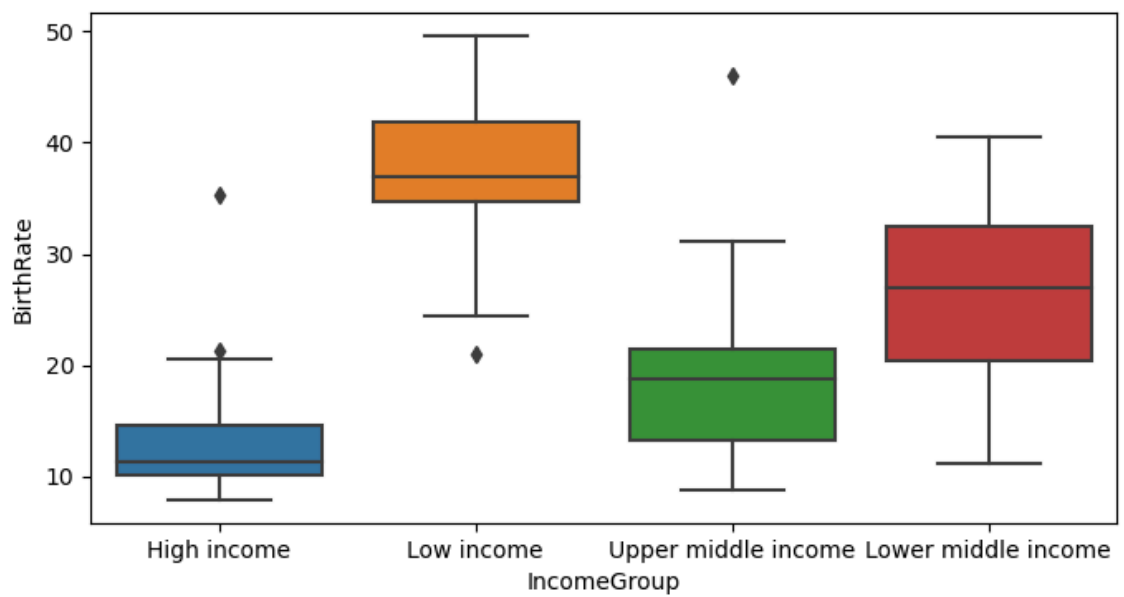
```
In [32]: vis1 = sns.distplot(df["InternetUsers"],bins = 10)
```



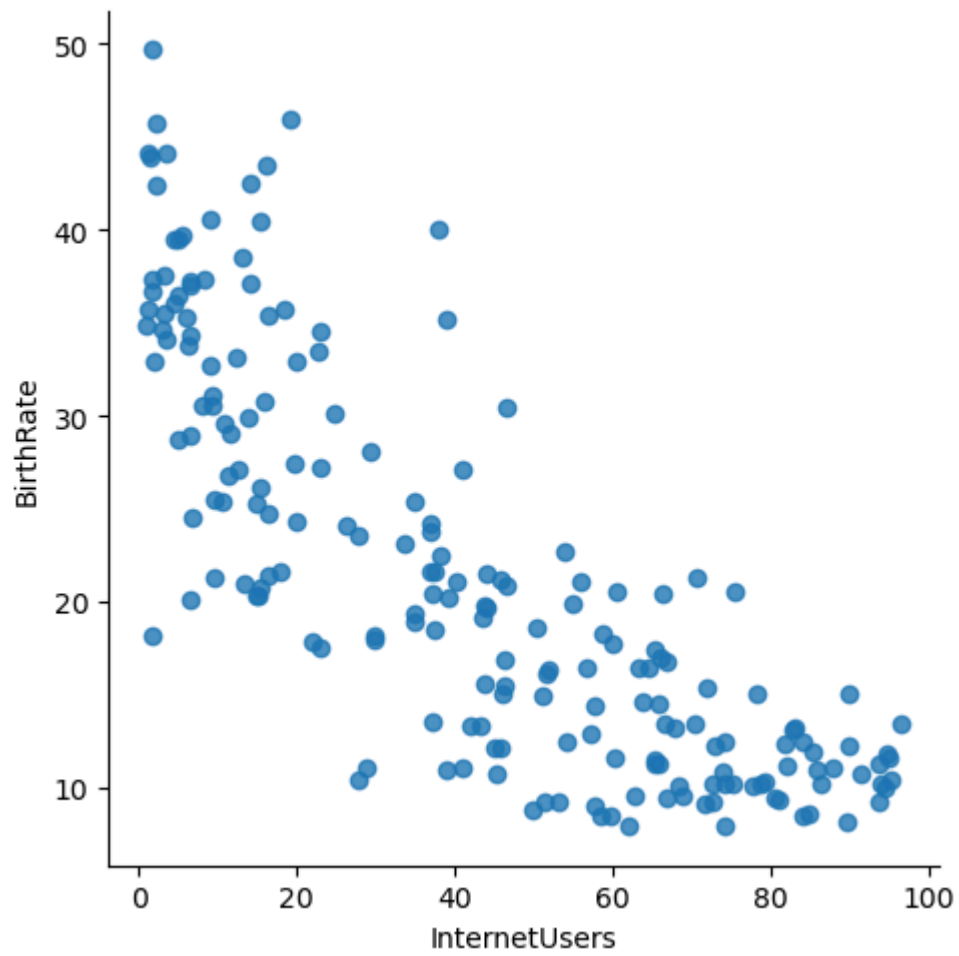
```
In [33]: vis1 = sns.distplot(df["BirthRate"],bins = 10)
```



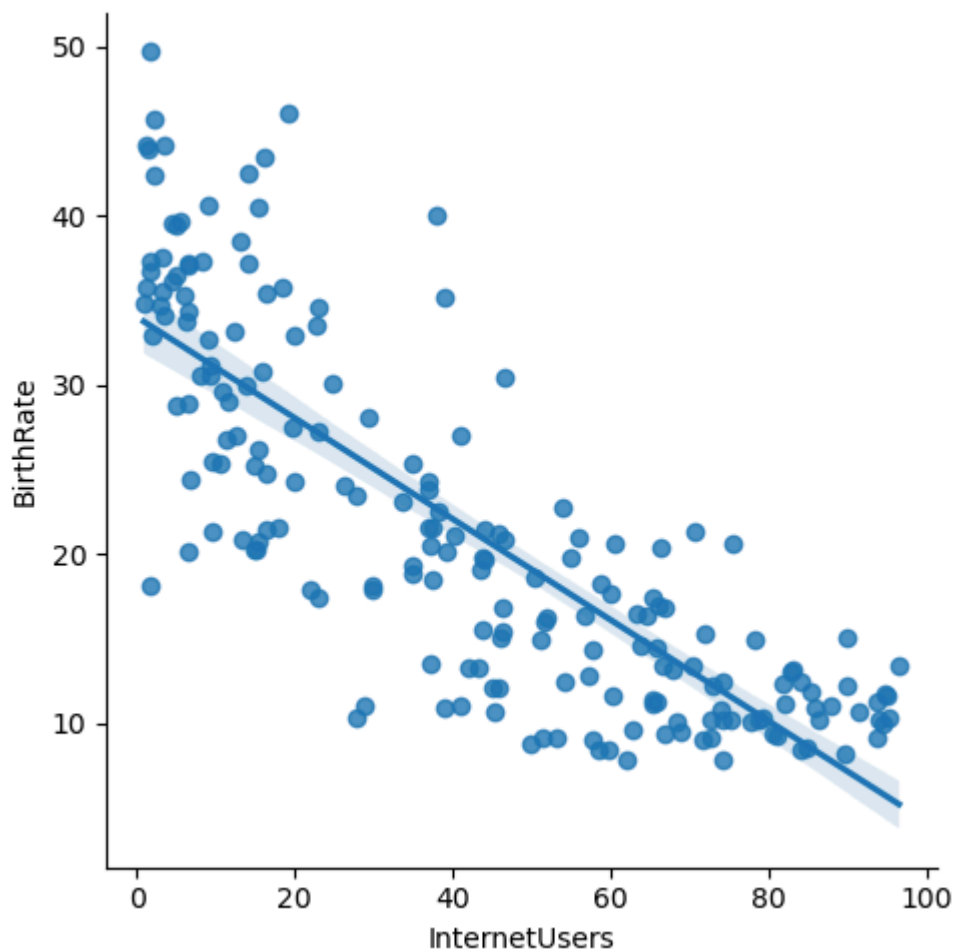
```
In [34]: vis2=sns.boxplot(data=df,x="IncomeGroup",y='BirthRate')
```



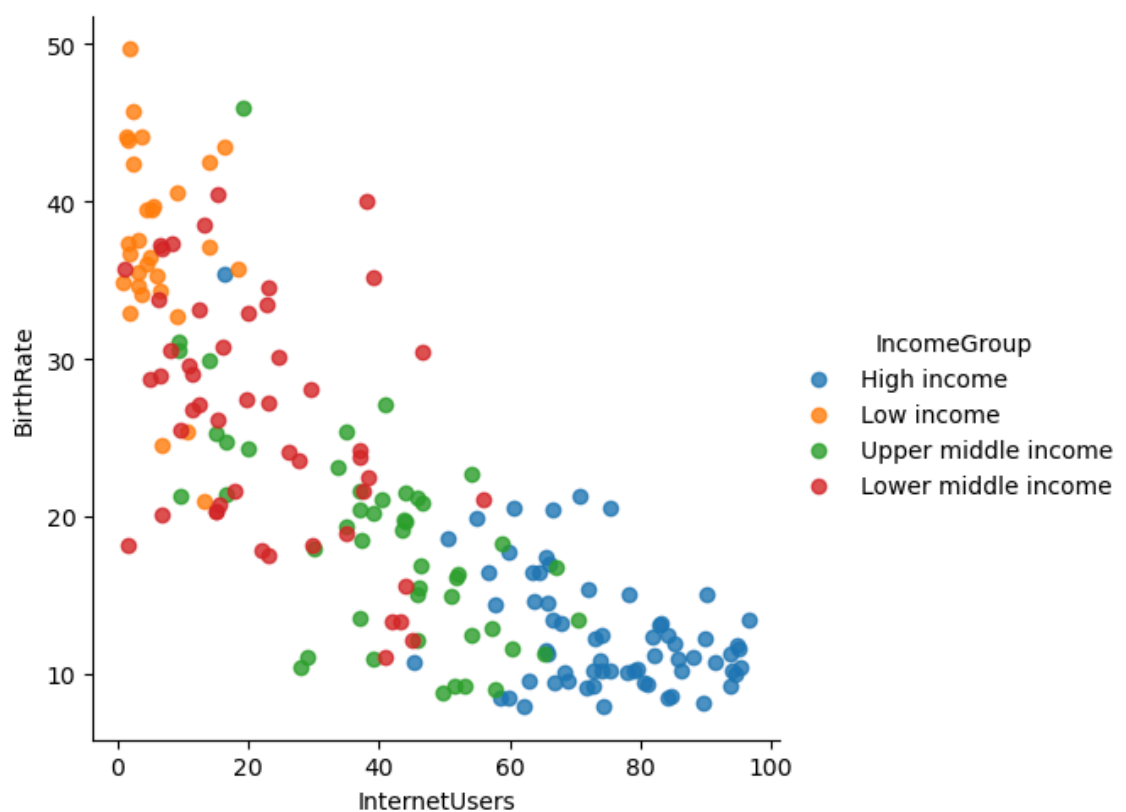
```
In [35]: vis3 = sns.lmplot(data = df, x="InternetUsers", y="BirthRate", fit_reg= Fa]
```



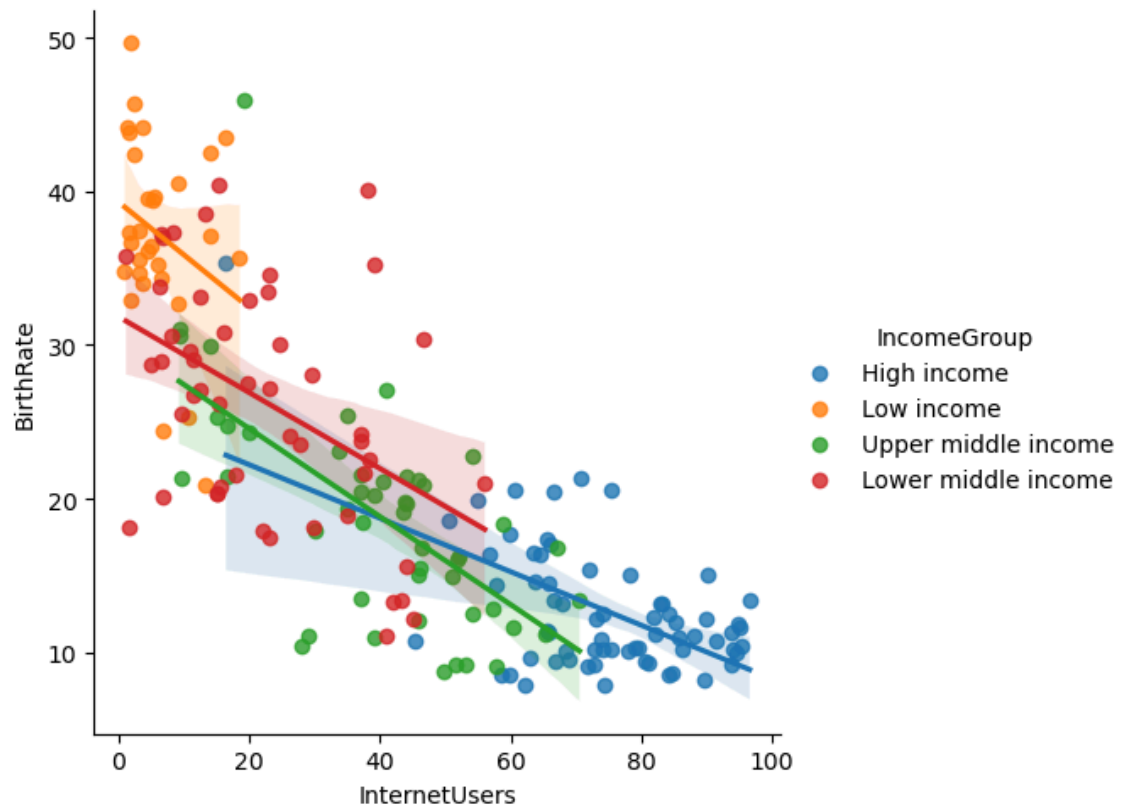
```
In [36]: vis4 = sns.lmplot(data=df, x='InternetUsers', y='BirthRate')
```



```
In [37]: vis5 = sns.lmplot(data=df, x='InternetUsers', y='BirthRate', fit_reg=False,
```



```
In [38]: vis5 = sns.lmplot(data=df, x='InternetUsers', y='BirthRate', hue='IncomeGroup')
```



```
In [ ]:
```