

Pandas ---> Pandas is a library to handle dataframe(Excel sheet data---->Data set)

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

```
In [2]: pd.__version__
```

```
Out[2]: '2.2.2'
```

```
In [3]: df = pd.read_csv(r'C:\Users\world\Desktop\FullStackDSandAI\Day22-09July2025\9th,10th')
```

```
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]: id(df)
```

```
Out[5]: 2196778379664
```

```
In [6]: len(df) # it gives number of records in dataset
```

```
Out[6]: 195
```

```
In [7]: df.columns # it gives names of columns
```

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

```
In [8]: len(df.columns) # it gives number of columns in data set
```

```
Out[8]: 5
```

```
In [9]: #To check for any missing values in dataset
        # Result 'False' means no missing value
        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]: # To check for missing values, same as isnull()
         df.isna()
```

Out[10]:

	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

To get count of missing values

df.isnull().sum()

In [11]: `df.isnull().sum()`

```
Out[11]: CountryName    0
CountryCode    0
BirthRate      0
InternetUsers   0
IncomeGroup     0
dtype: int64
```

isna().sum()

In [12]: `df.isna().sum()`

```
Out[12]: CountryName    0
CountryCode    0
BirthRate      0
InternetUsers   0
IncomeGroup     0
dtype: int64
```

To print top 5 rows ----> head()

```
In [13]: df.head()
```

```
Out[13]:
```

	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

To print bottom 5 rows -----> tail()

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

```
Out[14]:
```

	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

To get information of dataframe -----> df.info()

```
In [15]: 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
```

Slicing

```
In [16]: df[:]
```

Out[16]:

	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 [17]:

df[1:11]

Out[17]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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
10	Azerbaijan	AZE	18.300	58.7000	Upper middle income

In [18]:

df[:, -1]

Out[18]:

	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 [19]:

```
df[1:100:10]
```

Out[19]:

	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

In [20]:

```
df.head(2) # It gives top 2 records
```

Out[20]:

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

```
In [21]: df.describe() # it generates descriptive statistics, it always describe numerical d
```

Out[21]:

	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 [22]: df.head(1)
```

Out[22]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income

```
In [23]: df['CountryName']
```

Out[23]:

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

Out[24]:

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

195 rows × 2 columns

In [25]: `df[['CountryName', 'CountryCode', 'IncomeGroup']]`

Out[25]:

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

Out[26]:

	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 [27]: print(len(df.columns))
         print(len(df_cat.columns))
```

5
3

```
In [28]: print(df.columns)

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

```
In [29]: print(df_cat.columns)

Index(['CountryName', 'CountryCode', 'IncomeGroup'], dtype='object')
```

```
In [30]: df_cat.describe()
```

Out[30]:

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

```
In [31]: df_num=df[['BirthRate','InternetUsers']]
```

```
In [32]: 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   CountryName     195 non-null   object
1   CountryCode     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()
```

Out[36]:

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

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.describe().T
```

Out[38]:

	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 [39]: df.columns=['a','b','c','d','e']
```

```
In [40]: df.head(1)
```

Out[40]:

	a	b	c	d	e
0	Aruba	ABW	10.244	78.9	High income

```
In [41]: df.columns=['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'IncomeGroup']
```

```
In [42]: df.head(1)
```

Out[42]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income

subset

```
In [43]: df[['CountryName', 'BirthRate', 'InternetUsers']][4:8]
```

Out[43]:

	CountryName	BirthRate	InternetUsers
4	United Arab Emirates	11.044	88.0
5	Argentina	17.716	59.9
6	Armenia	13.308	41.9
7	Antigua and Barbuda	16.447	63.4

```
In [44]: df[4:8][['CountryName', 'BirthRate', 'InternetUsers']]
```

Out[44]:

	CountryName	BirthRate	InternetUsers
4	United Arab Emirates	11.044	88.0
5	Argentina	17.716	59.9
6	Armenia	13.308	41.9
7	Antigua and Barbuda	16.447	63.4

```
In [45]: df.columns
```

```
Out[45]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',  
              'IncomeGroup'],  
              dtype='object')
```

Mathematical Operation

```
In [46]: df.BirthRate * df.InternetUsers
```

```
Out[46]: 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 [47]: df.head(2)
```

```
Out[47]:
```

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

Add new column to existing data frame

```
In [48]: df['newColumn'] =df.BirthRate * df.InternetUsers
```

```
In [49]: df.head()
```

```
Out[49]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	newColumn
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 [50]: len(df.columns)
```

```
Out[50]: 6
```

Delete a column from dataframe

```
In [51]: df=df.drop('newColumn',axis=1)
```

```
In [52]: df.head(2)
```

Out[52]:

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

In [53]:

df

Out[53]:

	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

Conditions

In [54]:

df.InternetUsers<2

Out[54]:

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

Display the countries details which has less than 2 Internet Usage

```
In [55]: df[df.InternetUsers<2]
```

Out[55]:

	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 [56]: len(df[df.InternetUsers<2])
```

Out[56]: 9

Display country details which has Birthrate more than 40 percent

```
In [57]: df[df.BirthRate>40]
```

Out[57]:

	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 [58]: len(df[df.BirthRate>40])
```

```
Out[58]: 12
```

```
In [59]: low_internetuser_country=df.InternetUsers<2
```

```
In [60]: high_birthrate_country=df.BirthRate>40
```

```
In [61]: df[low_internetuser_country & high_birthrate_country]
```

Out[61]:

	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

Filter Records who has high income

```
In [62]: df[df.IncomeGroup== 'High income']
```

Out[62]:

	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 [63]: df[df.IncomeGroup== 'Low income']
```


Out[63]:

	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

Return unique values present in IncomeGroup column ---> unique()

```
In [64]: df.IncomeGroup.unique()
```

```
Out[64]: array(['High income', 'Low income', 'Upper middle income',  
              'Lower middle income'], dtype=object)
```

nunique() ---> returns number of unique elements in the object

```
In [65]: df.IncomeGroup.nunique()
```

```
Out[65]: 4
```

Introduction to seaborn

Seaborn is very powerful visualization(Statistic Visualization) pkg in python

Seaborn are used for advance visualization. Ex:distribution plot, Line plot

```
In [66]: import matplotlib.pyplot as plt    # visualization  
import seaborn as sns    # Statistic visualization, advanced visualization  
        # plot the graph in the line  
%matplotlib inline  
plt.rcParams['figure.figsize']=6,2    # rcParams parameter comes from plt library w  
  
import warnings  
warnings.filterwarnings('ignore')    # whenever OS will update, ignore the OS error
```

```
In [67]: df.columns
```

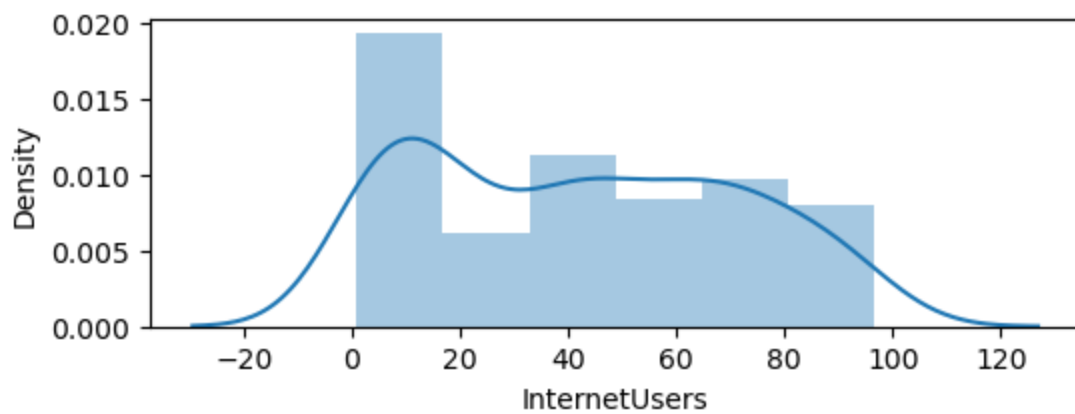
```
Out[67]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',  
              'IncomeGroup'],  
              dtype='object')
```

```
In [68]: df['InternetUsers']
```

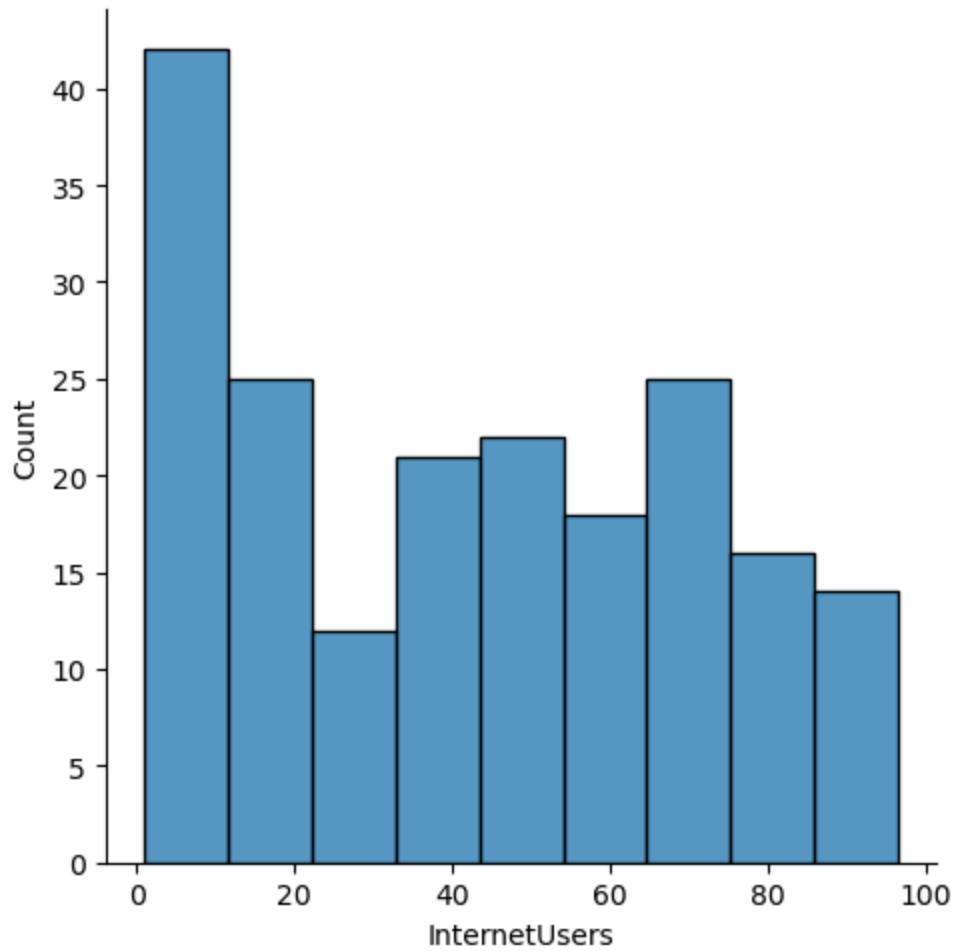
```
Out[68]: 0      78.9
         1       5.9
         2      19.1
         3      57.2
         4      88.0
         ...
        190     20.0
        191     46.5
        192       2.2
        193     15.4
        194     18.5
        Name: InternetUsers, Length: 195, dtype: float64
```

```
In [69]: vis1=sns.distplot(df['InternetUsers'])
```

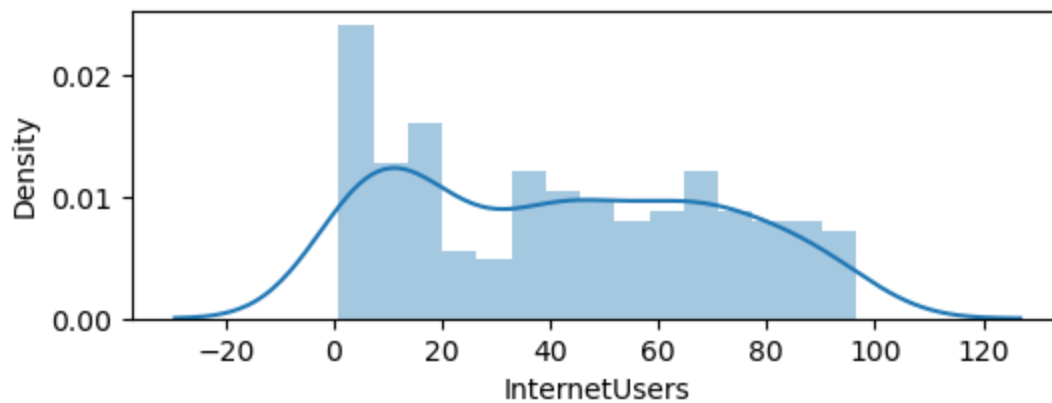
```
In [70]: plt.show(vis1)
```



```
In [71]: vis2=sns.displot(df['InternetUsers'])
         plt.show(vis2)
```

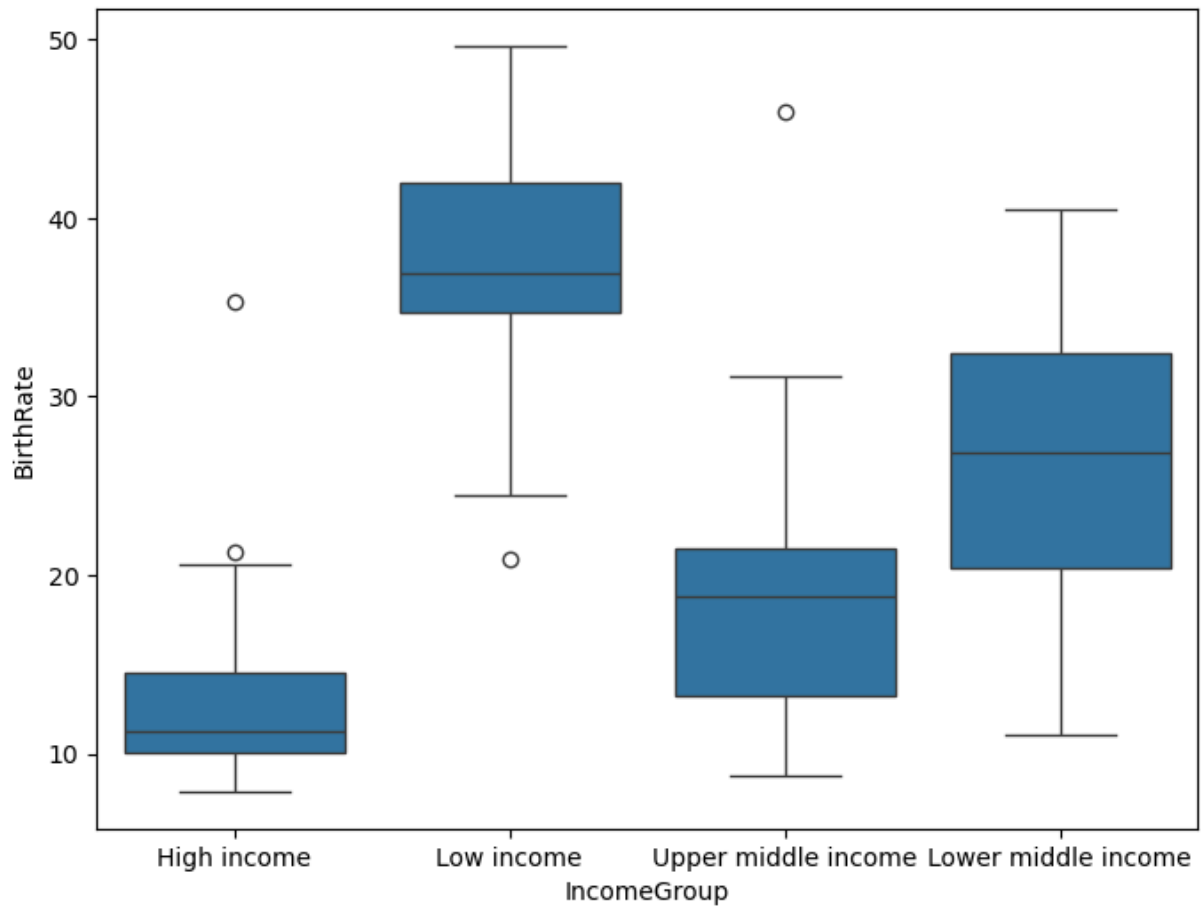


```
In [72]: vis3=sns.distplot(df['InternetUsers'], bins=15)  
plt.show(vis3)
```



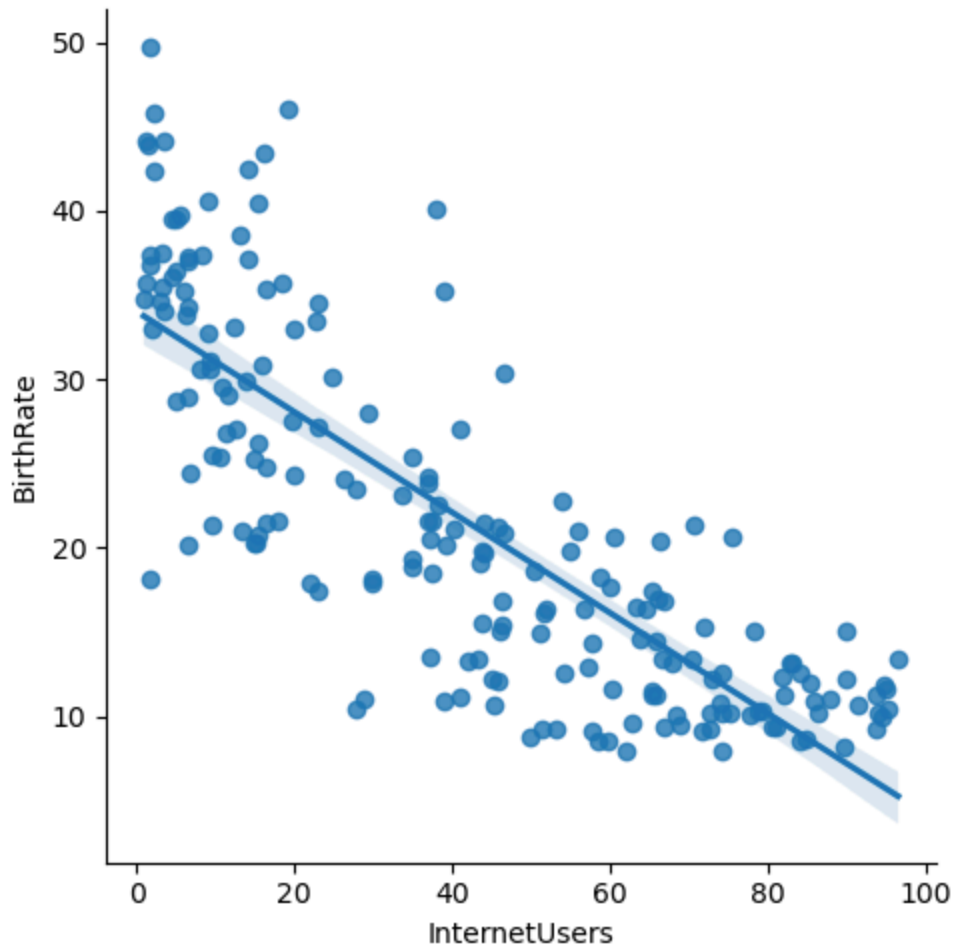
Box Plot

```
In [73]: plt.rcParams['figure.figsize']=8,6  
vis4=sns.boxplot(data=df,x="IncomeGroup", y='BirthRate')  
plt.show(vis4)
```

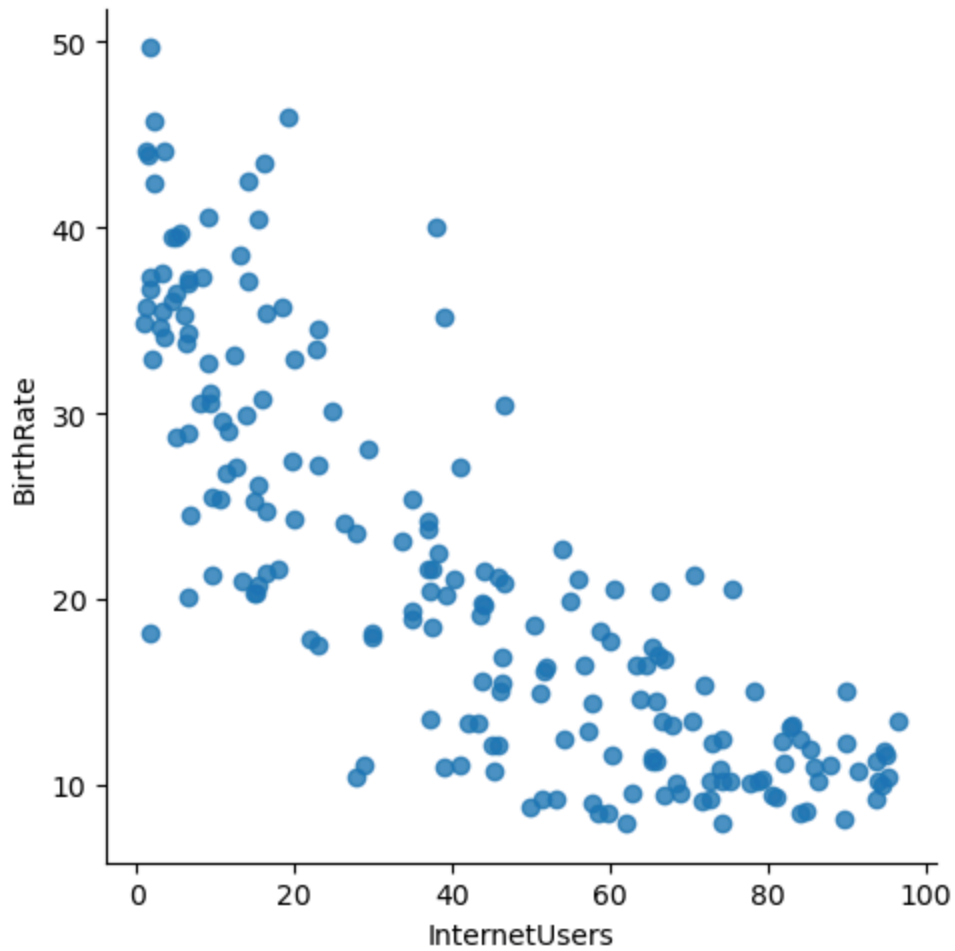


Linear Model(LM) Plot

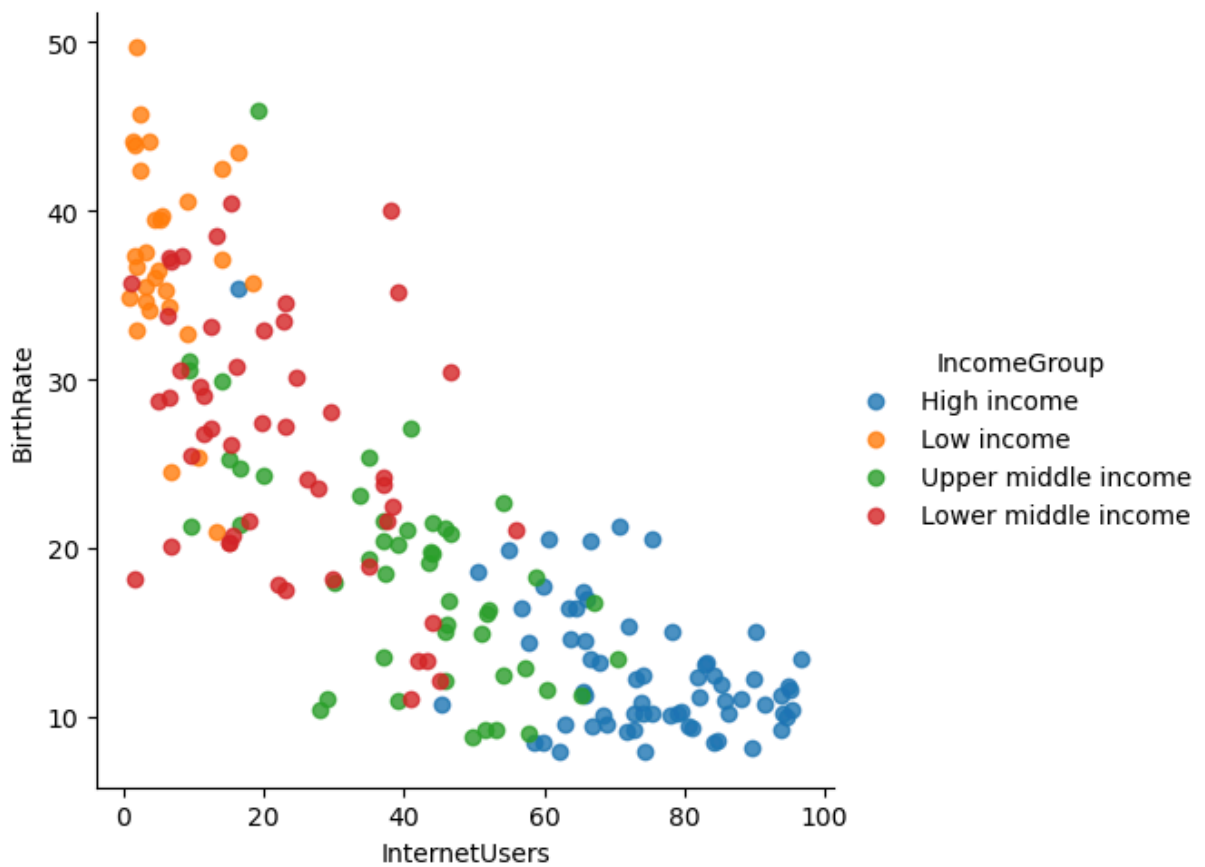
```
In [74]: vis5=sns.lmplot(data=df, x='InternetUsers', y='BirthRate')  
plt.show(vis5)
```



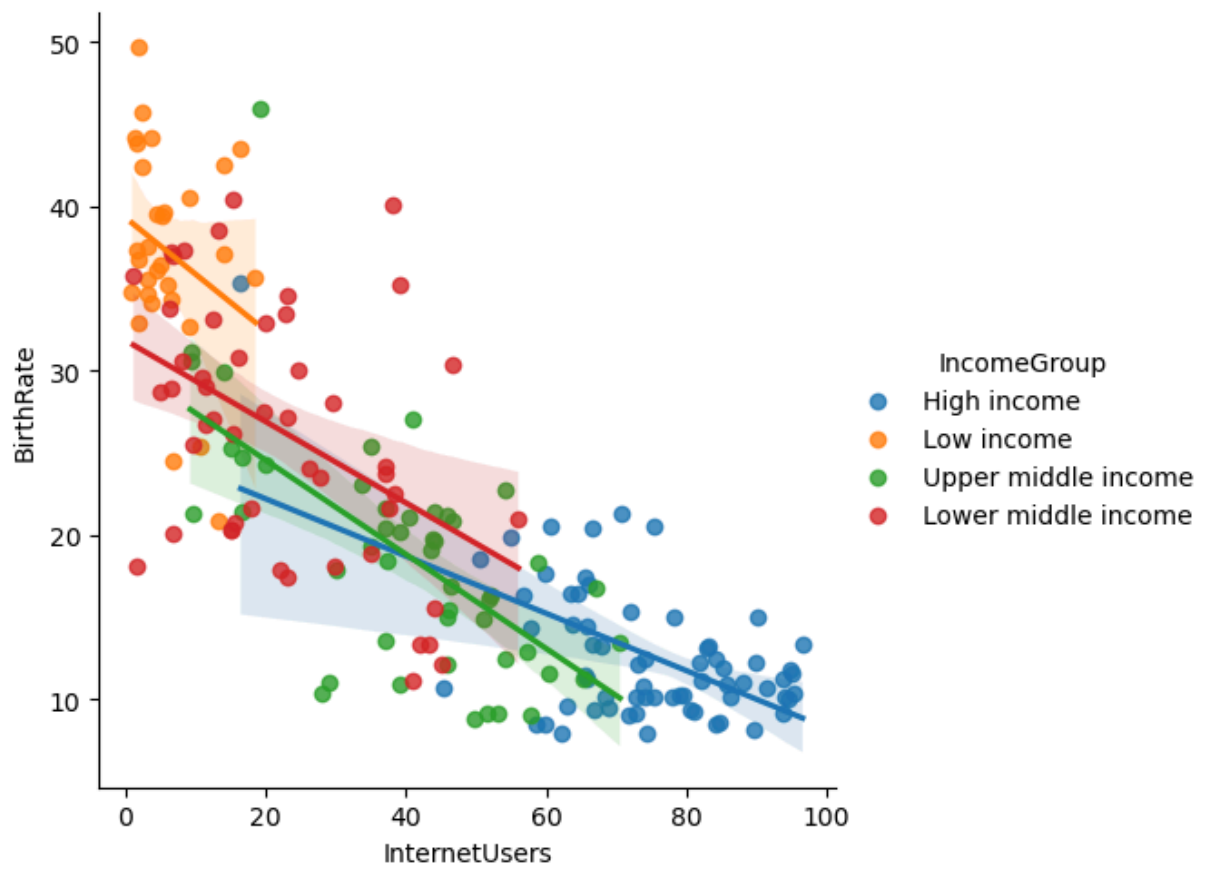
```
In [77]: # Regression Line from LMplot will disappear  
vis6=sns.lmplot(data=df, x='InternetUsers', y='BirthRate', fit_reg=False)  
plt.show(vis6)
```



```
In [79]: # hue draws data points for each category in the hue column
vis7=sns.lmplot(data=df, x='InternetUsers', y='BirthRate', fit_reg=False, hue='Inco
plt.show(vis7)
```



```
In [80]: vis8=sns.lmplot(data=df, x='InternetUsers', y='BirthRate', fit_reg=True, hue='IncomeGroup',  
plt.show(vis8)
```

In []:

In []:

In []: