Dataframe in python and how to import the dataset

pandas are very good package for dataframes &its perfect for dataset& very powerfull packages

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

In [2]:
```
stats=pd.read_csv(r"D:\NIT\20NOV\17th,18th\DataFrame_ Pandas\data.csv")
```

In [3]:
```
stats
```

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(stats)
```

Out[4]:
```
195
```

In [5]:
```
stats.shape
```

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

In [6]:
```
stats.columns
```

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

In [7]:
```
type(stats)
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [8]:
```
stats.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 [9]:
```
#4. Number of columns
len(stats.columns)
```

Out[9]:
```
5
```

In [10]:
```
#5. top rows
#head()
stats.head()
```

Out[10]:

	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 [11]:
```
stats.tail()
```

Out[11]:

	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 [12]:
```
stats.head(10)
```

Out[12]:

	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 [13]:
```
#6. Bottom rows
stats.tail(6)
```

Out[13]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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 [14]:
```
stats.describe()  # descriptive stats
#stats.describe . DESCRBE WIL LGIVE ONLY NUMERICAL INFORMATION
```

Out[14]:

	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 [15]:
```
stats.describe().transpose()
```

Out[15]:

	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 [18]:
```
stats.columns=["a","s","j","d","d"]
```

In [19]:
```
stats
```

Out[19]:

	a	s	j	d	d
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 [20]:
```
stats.columns = ['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers','Inc
```

In [21]:
```
stats
```

Out[21]:

	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 [22]:
```
# subsetting a dataframes in pandas

#1. Rows
#2. Columns
#3. combine the two
```

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 × 5 columns

In [25]:
```
# Rows:
stats[0:5]  #how python know that only this is rows based on index
```

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

In [26]:
```
stats[:]      #it will show entire dataframe
```

Out[26]:

	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 [27]:
```
stats[:10]
```

Out[27]:

	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 [28]:
```
stats.head(10)
```

Out[28]:

	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 [29]:
```
# How to reverse the dataframe

stats[: : -1]
```

Out[29]:

	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 [31]:
```
# get only every 20th row
stats[: :  9]
```

Out[31]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9000	High income
9	Austria	AUT	9.400	80.6188	High income
18	Bahamas, The	BHS	15.339	72.0000	High income
27	Bhutan	BTN	18.134	29.9000	Lower middle income
36	Congo, Rep.	COG	37.011	6.6000	Lower middle income
45	Germany	DEU	8.500	84.1700	High income
54	Estonia	EST	10.300	79.4000	High income
63	Ghana	GHA	33.131	12.3000	Lower middle income
72	Guam	GUM	17.389	65.4000	High income
81	Ireland	IRL	15.000	78.2477	High income
90	Kazakhstan	KAZ	22.730	54.0000	Upper middle income
99	Liberia	LBR	35.521	3.2000	Low income
108	Macao SAR, China	MAC	11.256	65.8000	High income
117	Myanmar	MMR	18.119	1.6000	Lower middle income
126	New Caledonia	NCL	17.000	66.0000	High income
135	Pakistan	PAK	29.582	10.9000	Lower middle income
144	French Polynesia	PYF	16.393	56.8000	High income
153	Solomon Islands	SLB	30.578	8.0000	Lower middle income
162	Slovenia	SVN	10.200	72.6756	High income
171	Turkmenistan	TKM	21.322	9.6000	Upper middle income
180	Uruguay	URY	14.374	57.6900	High income
189	Samoa	WSM	26.172	15.3000	Lower middle income

In [32]:
```
stats['IncomeGroup']
```

Out[32]:
```
0              High income
1               Low income
2      Upper middle income
3      Upper middle income
4              High income
              ...
190    Lower middle income
191    Upper middle income
192             Low income
193    Lower middle income
194             Low income
Name: IncomeGroup, Length: 195, dtype: object
```

In [33]:
```
['CountryName','BirthRate']
```

Out[33]:
```
['CountryName', 'BirthRate']
```

In [34]:
```
stats[['CountryName','BirthRate']]
```

Out[34]:

	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 [35]:
```
stats['BirthRate'].head()
```

Out[35]:
```
0    10.244
1    35.253
2    45.985
3    12.877
4    11.044
Name: BirthRate, dtype: float64
```

In [36]:
```
stats[4:8]
```

Out[36]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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

In [37]:
```
# combine the two
stats[4:8][['CountryName','BirthRate']]
```

Out[37]:

	CountryName	BirthRate
4	United Arab Emirates	11.044
5	Argentina	17.716
6	Armenia	13.308
7	Antigua and Barbuda	16.447

In [38]:
```
df1 = stats [['CountryName','BirthRate']]
```

In [39]:
```
df1
```

Out[39]:

	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 [40]:
```
df2 = stats[4:8]
```

In [41]:
```
df2
```

Out[41]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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

In [42]:
```
stats[['CountryCode','BirthRate','InternetUsers']][4:8] #subet dataframe
```

Out[42]:

	CountryCode	BirthRate	InternetUsers
4	ARE	11.044	88.0
5	ARG	17.716	59.9
6	ARM	13.308	41.9
7	ATG	16.447	63.4

In [43]:
```
#Mathmetical operation =
stats.BirthRate * stats.InternetUsers
```

Out[43]:
```
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 [44]:
```
# Add a column

stats['myCalc'] = stats.BirthRate * stats.InternetUsers
```

In [45]:
```
stats
```

Out[45]:

	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 [47]:
```
stats['myCalc']
```

Out[47]:
```
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
Name: myCalc, Length: 195, dtype: float64
```

In [48]:
```
len(stats.columns)
```

Out[48]:
```
6
```

In [49]:
```
stats = stats.drop('myCalc',axis = 1)
```

In [50]:
```
stats
```

Out[50]:

	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 [51]:
```
stats.columns[3:4]
```

Out[51]:
```
Index(['InternetUsers'], dtype='object')
```

In [52]:
```
stats.InternetUsers<2 #we are checking given condition if its correct true or fa
```

Out[52]:
```
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 [53]:
```
Filter = stats.InternetUsers < 2
Filter ## internet user<2
```

Out[53]:
```
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 [54]:
```
stats[Filter]
```

Out[54]:

	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 [55]:
```
len(stats[Filter])
```

Out[55]:
```
9
```

In [56]:
```
stats.BirthRate>40
```

Out[56]:
```
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 [57]:
```
Filter2 = stats.BirthRate>40
```

In [58]:
```
stats[Filter2]
```

Out[58]:

	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 [59]:
```
len(Filter2)
```

Out[59]:
```
195
```

In [60]:
```
#Filter and Filter2
Filter & Filter2
```

Out[60]:
```
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 [61]:
```
stats[Filter & Filter2]
```

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

In [62]:
```
stats[(stats.BirthRate > 40) & (stats.InternetUsers < 2)]
```

Out[62]:

	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

In [65]:

```
stats[stats.IncomeGroup == 'Low income']
```

Out[65]:

	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 [66]:
```
# How to get the unique categories

stats.IncomeGroup.unique()
```

Out[66]:
```
array(['High income', 'Low income', 'Upper middle income',
       'Lower middle income'], dtype=object)
```

In [69]:
```
stats.IncomeGroup.nunique()
```

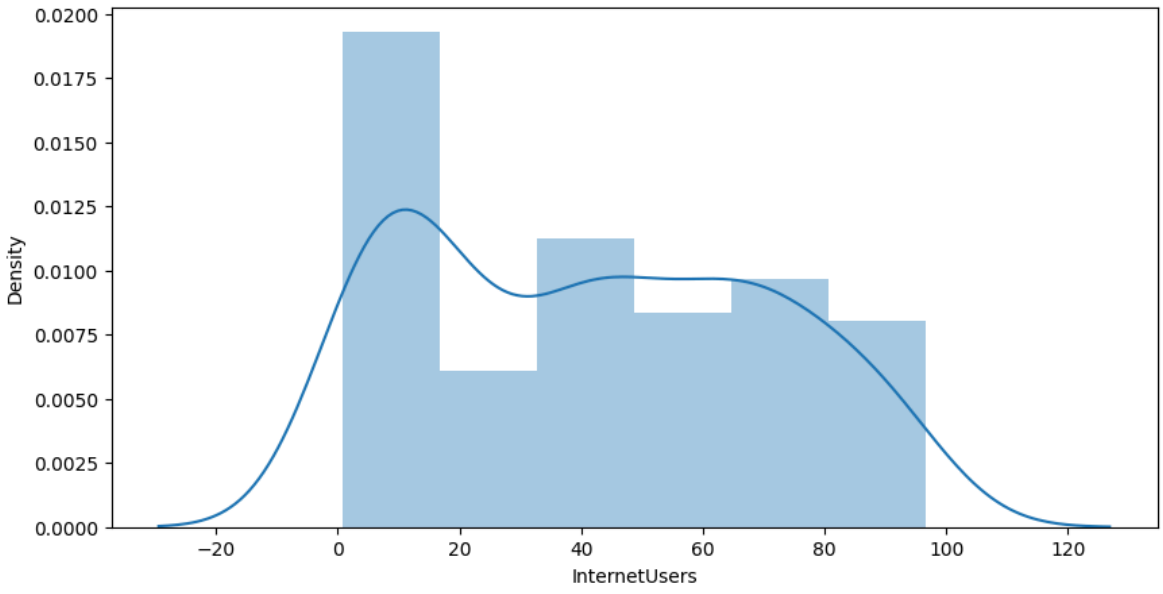
Out[69]:
```
4
```

In [70]:
```
# Introduction to seaborn # seaborn is very powerfull visualizatio(STATISTIC VIS

import matplotlib.pyplot as plt # visulaiztion
import seaborn as sns # distribution visualtion
# seaborn are used for advance visualization e.x --> distribution plot, line plo

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

import warnings
warnings.filterwarnings('ignore') # os error
```

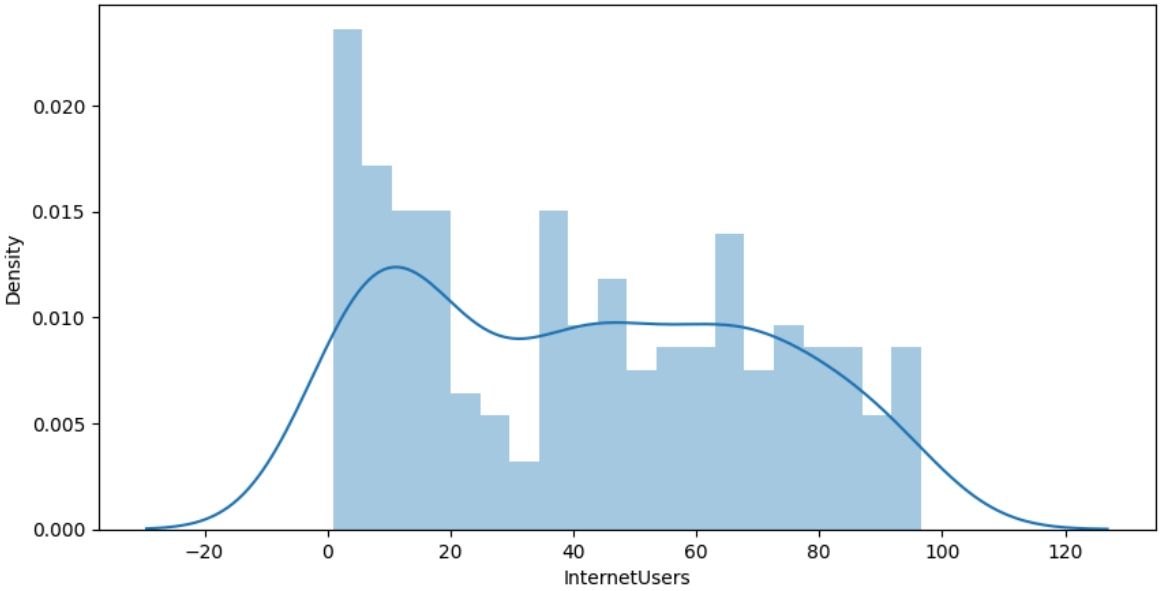
In [72]:
```
stats["InternetUsers"]
```

Out[72]:
```
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
```

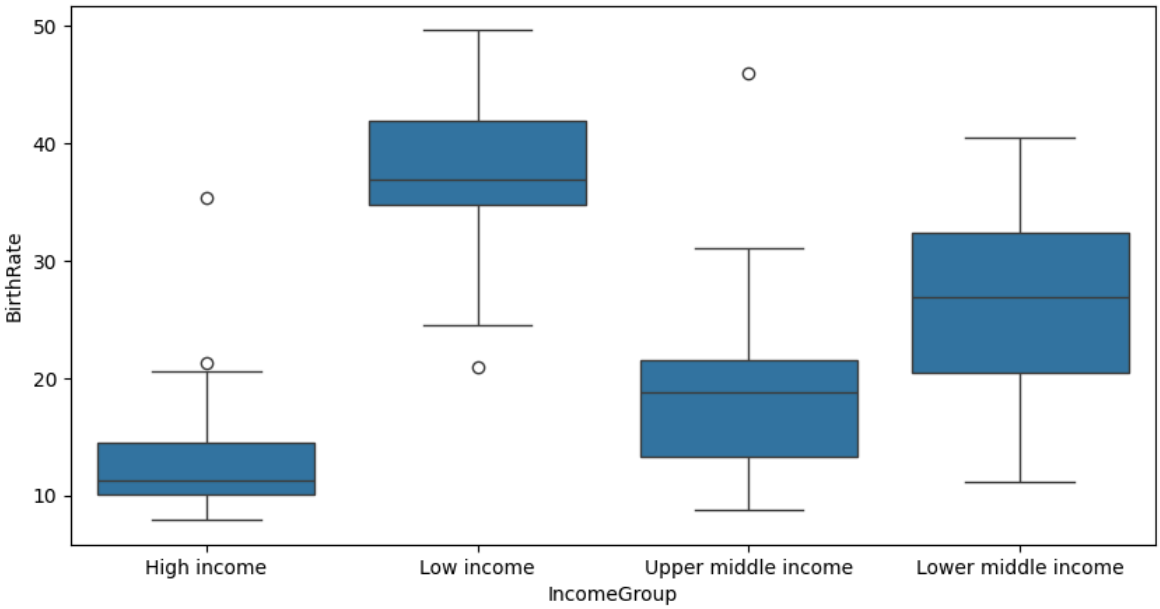
In [71]:
```
# Distributions:
vis1 = sns.distplot(stats["InternetUsers"])  # UNIVARIATE ANALYSIS - statistics
```



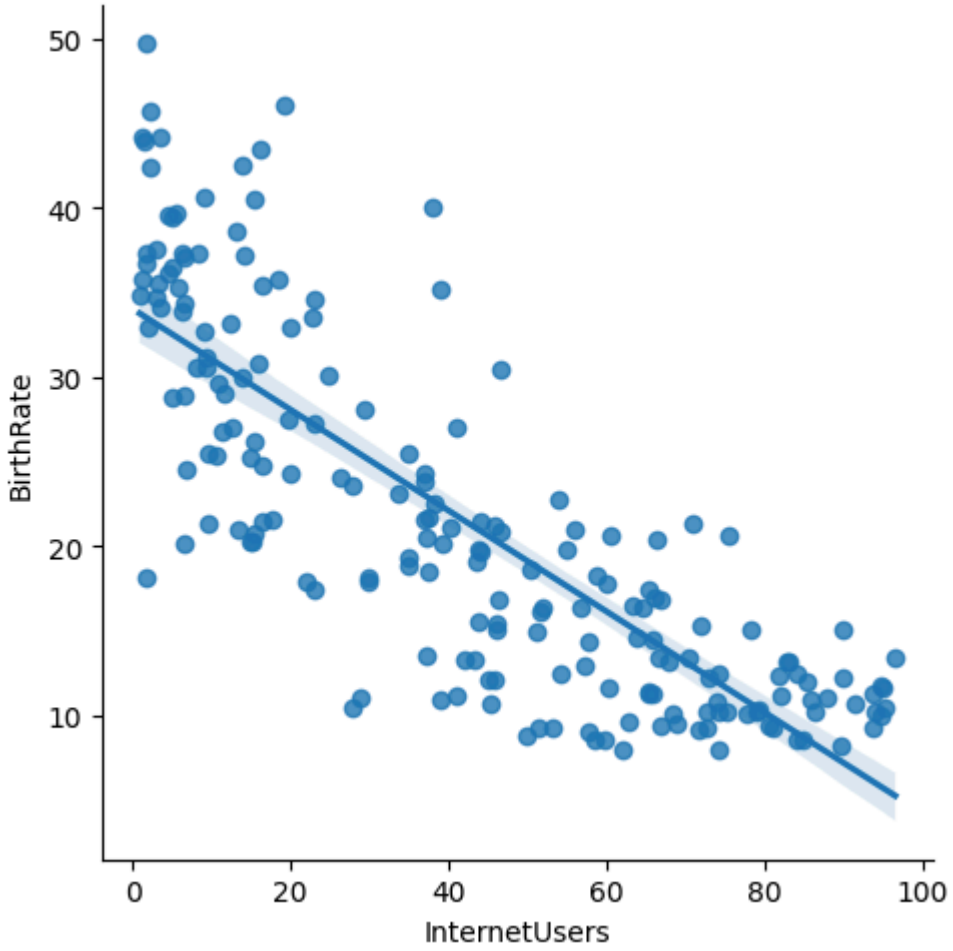
In [74]:
```
vis1=sns.distplot(stats
                  ["InternetUsers"],bins=20)
```



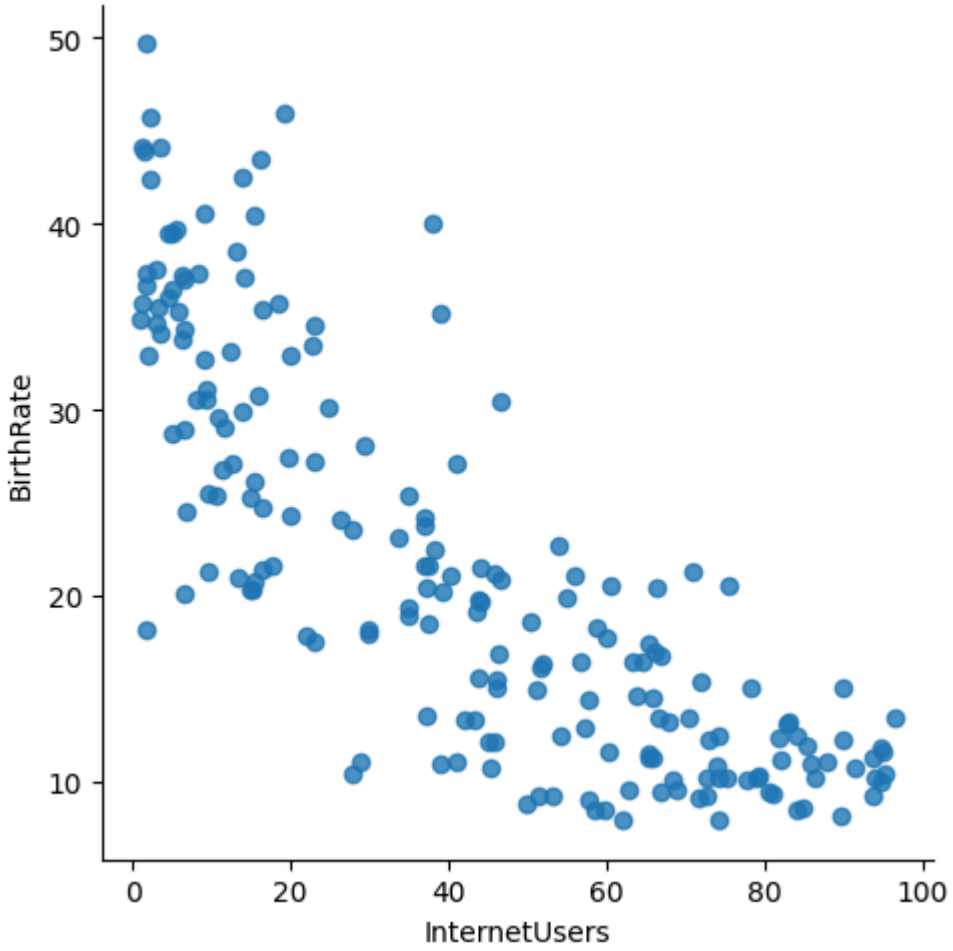
In [75]:
```
#BOX PLOTS:
vis2 = sns.boxplot( data = stats, x="IncomeGroup", y='BirthRate')  #BI-VARIATE A
```



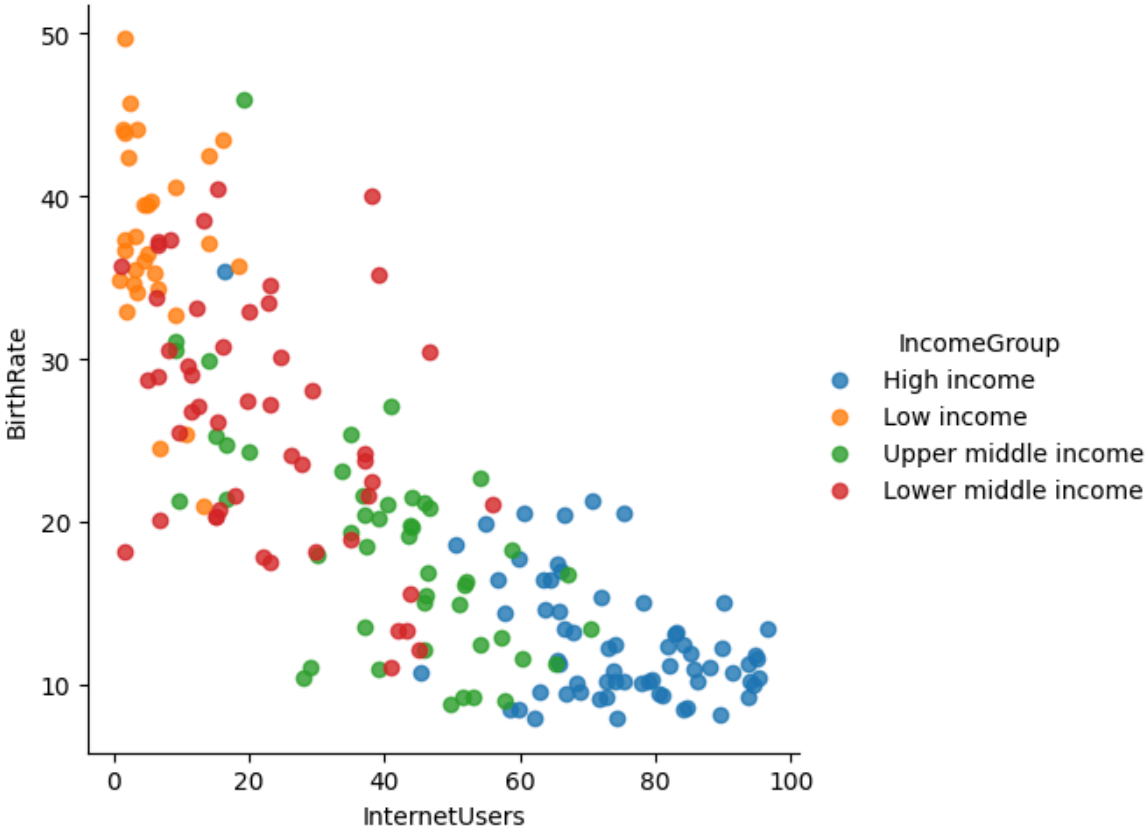
In [76]:
```
vis4 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate')
```



In [77]:
```
vis3 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate', fit_reg = F
```



In [78]:
```
vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate',
                  fit_reg = False,hue = 'IncomeGroup') #hue - parameter for colo
```



In [ ]: