

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 #USE FOR DATAFRAMES`

In [2]: `# How to read the dataet`

```
stats = pd.read_csv(r"D:\datasciene\classwork\17th,20th\DataFrame_ Pandas\Demogra
#stats = pd.read_csv("C:\\Users\\kdata\\Desktop\\AMXWAM\\AMXWAM PILOT BATCH - A1
[...]
```

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 [3]: `# Explore data in python
#1. Full dataframe
#2. How many rows & columns. you have to chk the row because the no. of raw should be same
len(stats) #195 rows imported (this is for tracking later part)`

Out[3]: 195

In [5]: `#3. see columns
stats.columns`

Out[5]: `Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
 'IncomeGroup'],
 dtype='object')`

In [6]: #4. Number of columns

```
len(stats.columns)
```

Out[6]: 5

In [7]: #5. top rows

```
stats.head() # it will print top 5 rows
```

Out[7]:

	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 [8]: stats.head(2)

Out[8]:

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

In [9]: #6. Bottom rows

```
stats.tail() #last 5 rows
```

Out[9]:

	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 [10]: stats.tail(3)

Out[10]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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]: #7. information of the column

```
stats.info() #strings are called as object
```

```
<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]: #8. get stats on the columns

```
stats.describe() #it will work like a statistic fun
```

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]: stats.describe().transpose() #transpose convert column into rows

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]: # Renaming columns of a dataframe

```
stats.head()
```

Out[14]:

	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 [15]: stats.columns

Out[15]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'IncomeGroup'], dtype='object')

In [16]: stats.columns = ['a', 'b', 'c', 'd', 'e']
stats.head()

Out[16]:

	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 [17]: stats.columns = ['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'IncomeGroup']

In [18]: stats.head()

Out[18]:

	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 [19]: # subsetting a dataframe in pandas

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

In [20]: # Rows:

```
stats[21:26] #how python know that only this is rows based on index
```

Out[20]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
21	Belize	BLZ	23.092	33.60	Upper middle income
22	Bermuda	BMU	10.400	95.30	High income
23	Bolivia	BOL	24.236	36.94	Lower middle income
24	Brazil	BRA	14.931	51.04	Upper middle income
25	Barbados	BRB	12.188	73.00	High income

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]: stats[:10]

Out[22]:

	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 [23]: stats.head(10)

Out[23]:

	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 [24]: # How to reverse the dataframe

```
stats[ : : -1]
```

Out[24]:

	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 [25]: stats

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
...
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 [26]: # get only every 20th row

```
stats[::20]
```

Out[26]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9000	High income
20	Belarus	BLR	12.500	54.1700	Upper middle income
40	Costa Rica	CRI	15.022	45.9600	Upper middle income
60	Gabon	GAB	30.555	9.2000	Upper middle income
80	India	IND	20.291	15.1000	Lower middle income
100	Libya	LBY	21.425	16.5000	Upper middle income
120	Mozambique	MOZ	39.705	5.4000	Low income
140	Poland	POL	9.600	62.8492	High income
160	Suriname	SUR	18.455	37.4000	Upper middle income
180	Uruguay	URY	14.374	57.6900	High income

In [27]: # COLUMNS:

```
stats.columns
```

Out[27]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
 'IncomeGroup'],
 dtype='object')

In [28]: stats.head()

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

In [30]: stats.columns = [['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
 'IncomeGroup']]

In [31]: `stats['CountryName'].head()`

Out[31]:

CountryName	
0	Aruba
1	Afghanistan
2	Angola
3	Albania
4	United Arab Emirates

In [32]: `['CountryName', 'BirthRate']`

Out[32]: `['CountryName', 'BirthRate']`

In [33]: `stats[['CountryName', 'BirthRate']].head()`

Out[33]:

	CountryName	BirthRate
0	Aruba	10.244
1	Afghanistan	35.253
2	Angola	45.985
3	Albania	12.877
4	United Arab Emirates	11.044

In [34]: `stats.head()`

Out[34]:

	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 [35]: `stats['BirthRate']`

Out[35]:

BirthRate	
0	10.244
1	35.253
2	45.985
3	12.877
4	11.044
...	...
190	32.947
191	20.850
192	42.394
193	40.471
194	35.715

195 rows × 1 columns

In [36]: `# combine the two`

`stats[4:8][['CountryName', 'BirthRate']]`

Out[36]:

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

In [37]:

`stats[['CountryName', 'BirthRate']][4:8]`

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 [39]: `df1 = stats[['CountryName', 'BirthRate', 'InternetUsers']]`

In [40]: df1

Out[40]:

	CountryName	BirthRate	InternetUsers
0	Aruba	10.244	78.9
1	Afghanistan	35.253	5.9
2	Angola	45.985	19.1
3	Albania	12.877	57.2
4	United Arab Emirates	11.044	88.0
...
190	Yemen, Rep.	32.947	20.0
191	South Africa	20.850	46.5
192	Congo, Dem. Rep.	42.394	2.2
193	Zambia	40.471	15.4
194	Zimbabwe	35.715	18.5

195 rows × 3 columns

In [41]: df2 = stats[4:8]

In [42]: df2

Out[42]:

	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 [43]: # Basic operation of dataframe
stats.head()

Out[43]:

	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 [44]: `stats[['CountryCode', 'BirthRate', 'InternetUsers']][4:8] #subset dataframe`

Out[44]:

	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 [45]: `stats.head()`

Out[45]:

	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 [46]: `stats[['BirthRate', 'InternetUsers']]`

Out[46]:

	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 [48]: `stats.head()`

Out[48]:

	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 [51]: `stats['BirthRate']`

Out[51]:

	BirthRate
0	10.244
1	35.253
2	45.985
3	12.877
4	11.044
...	...
190	32.947
191	20.850
192	42.394
193	40.471
194	35.715

195 rows × 1 columns

```
In [52]: stats['InternetUsers']
```

Out[52]:

InternetUsers	
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

195 rows × 1 columns

```
In [61]: #Mathmetical operation =  
stats.BirthRate * stats.InternetUsers
```

```
Out[61]: 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 [62]: # Add a column
```

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

In [63]: `stats.head()`

Out[63]:

	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

In [64]: `#Remove a column`

`stats.drop('myCalc',axis = 1)`

Out[64]:

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

In [66]: `stats.head()`

Out[66]:

	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 [67]: stats.columns[2]
```

```
Out[67]: 'BirthRate'
```

```
In [68]: stats.InternetUsers<2 #we are checking given condition if its correct true or fal
```

```
Out[68]: 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 [69]: Filter = stats.InternetUsers < 2
```

```
In [70]: Filter
```

```
Out[70]: 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 [71]: stats[3:7]
```

```
Out[71]:
```

	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

In [72]: stats[30:40]

Out[72]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
30	Canada	CAN	10.900	85.80	High income
31	Switzerland	CHE	10.200	86.34	High income
32	Chile	CHL	13.385	66.50	High income
33	China	CHN	12.100	45.80	Upper middle income
34	Cote d'Ivoire	CIV	37.320	8.40	Lower middle income
35	Cameroon	CMR	37.236	6.40	Lower middle income
36	Congo, Rep.	COG	37.011	6.60	Lower middle income
37	Colombia	COL	16.076	51.70	Upper middle income
38	Comoros	COM	34.326	6.50	Low income
39	Cabo Verde	CPV	21.625	37.50	Lower middle income

In [73]: stats[Filter] # IT WILL take that row which are false

Out[73]:

	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 [74]: stats.BirthRate>40

Out[74]:

```
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 [75]: Filter2 = stats.BirthRate>40
```

```
In [76]: Filter2
```

```
Out[76]: 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 [77]: stats[Filter2]
```

```
Out[77]:
```

	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 [78]: #Filter and Filter2
Filter & Filter2

Out[78]: 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 [79]: stats[Filter & Filter2]

Out[79]:

	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 [80]: stats[(stats.BirthRate > 40) & (stats.InternetUsers < 2)]

Out[80]:

	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 [81]: stats.head()

Out[81]:

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

Out[82]:

	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

In [83]: `# How to get the unique categories`

```
stats.IncomeGroup.unique()
```

Out[83]: `array(['High income', 'Low income', 'Upper middle income', 'Lower middle income'], dtype=object)`

In [84]: `# Introduction to seaborn # seaborn is very powerfull visualizatio(STATISTIC VISU)`

```
import matplotlib.pyplot as plt # visulaiztion
import seaborn as sns # distribution visualtion

%matplotlib inline
plt.rcParams['figure.figsize'] = 8,4

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

In [85]: `stats.head()`

Out[85]:

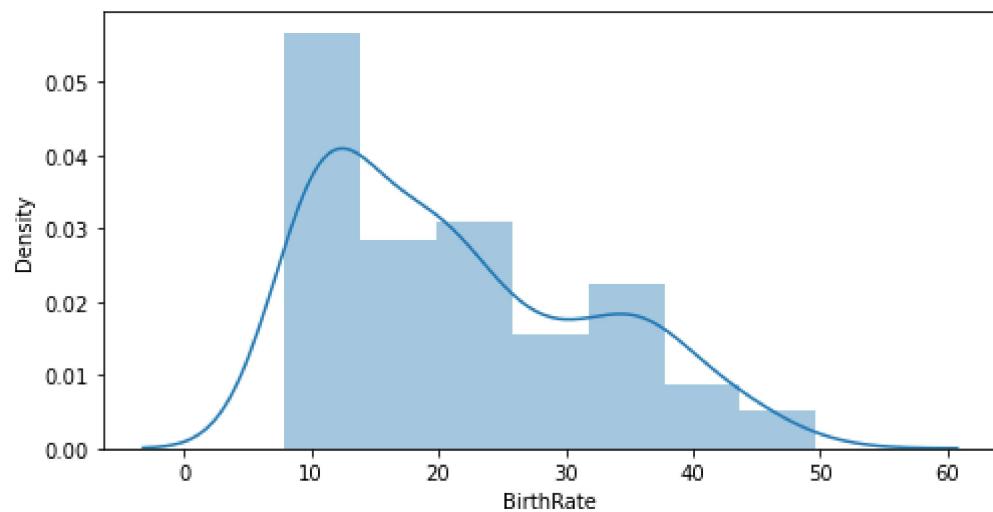
	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income

In [92]: # Distributions:

```
vis1 = sns.distplot(stats["BirthRate"])
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Future Warning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

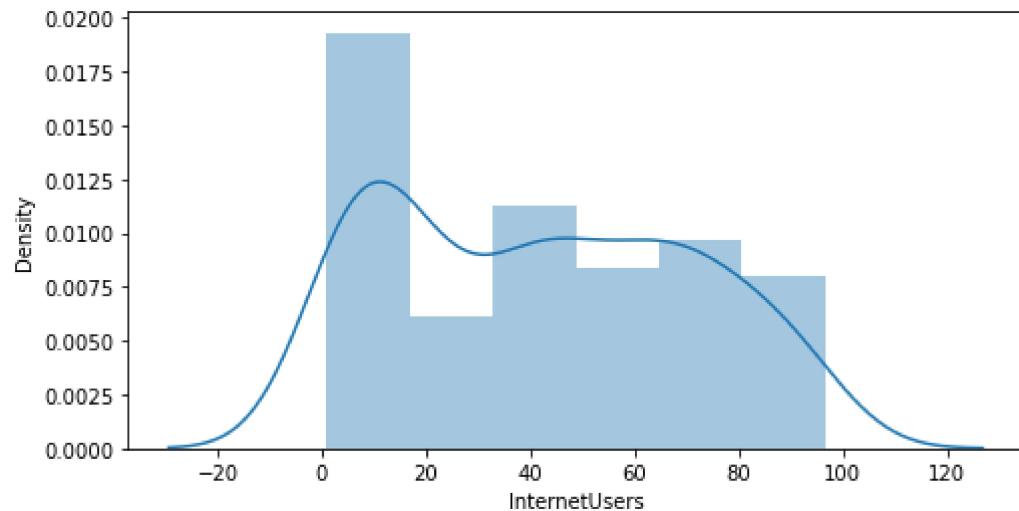


In [93]: # Distributions:

```
vis2 = sns.distplot(stats["InternetUsers"])
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Future Warning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

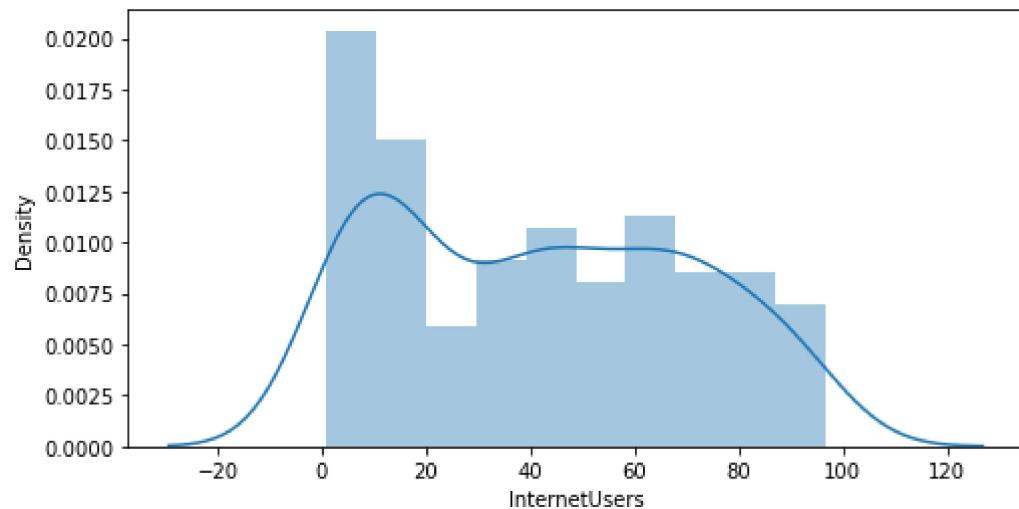
```
warnings.warn(msg, FutureWarning)
```



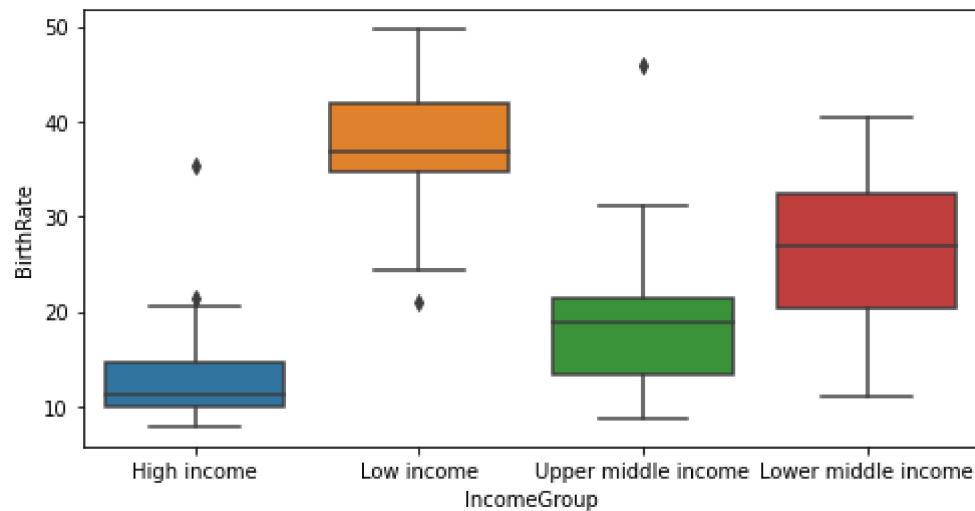
```
In [101]: vis1 = sns.distplot(stats["InternetUsers"], bins=10)
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



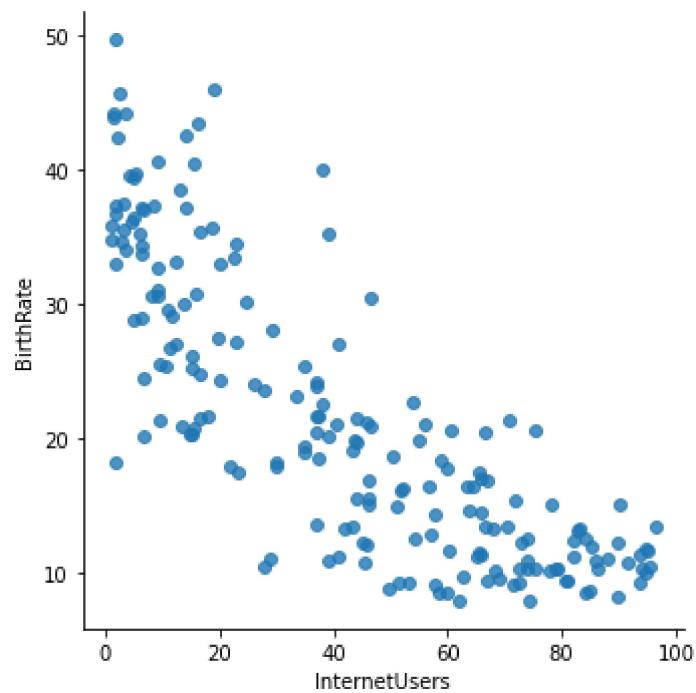
```
In [102]: #BOX PLOTS:  
vis2 = sns.boxplot(data = stats, x="IncomeGroup", y='BirthRate')
```



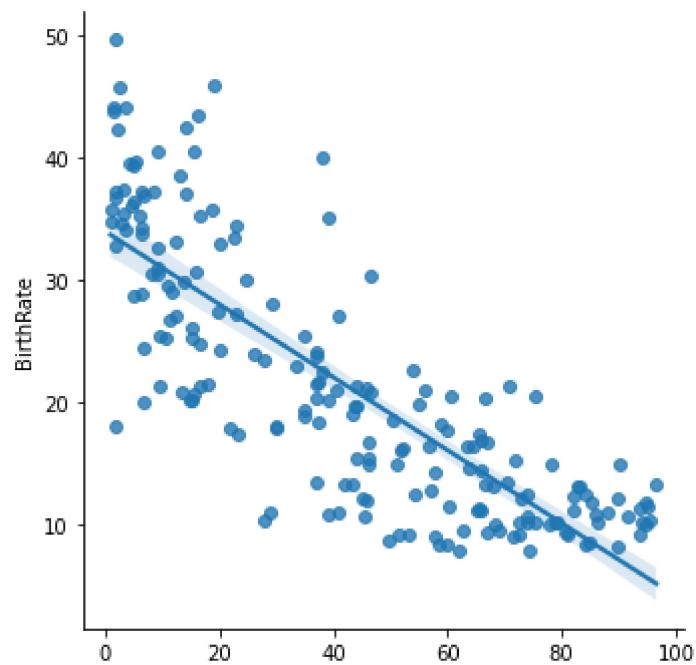
```
In [1]: # refer to seaborn gallery
```

```
In [2]: # visualizing with seaborn
```

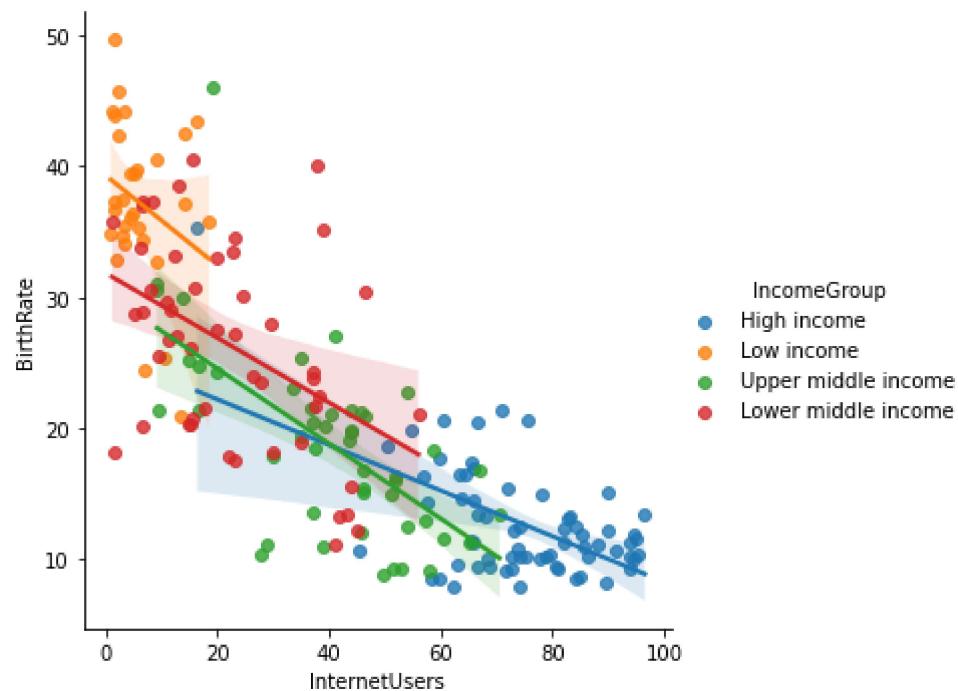
```
In [105]: vis3 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate', fit_reg = False)
```



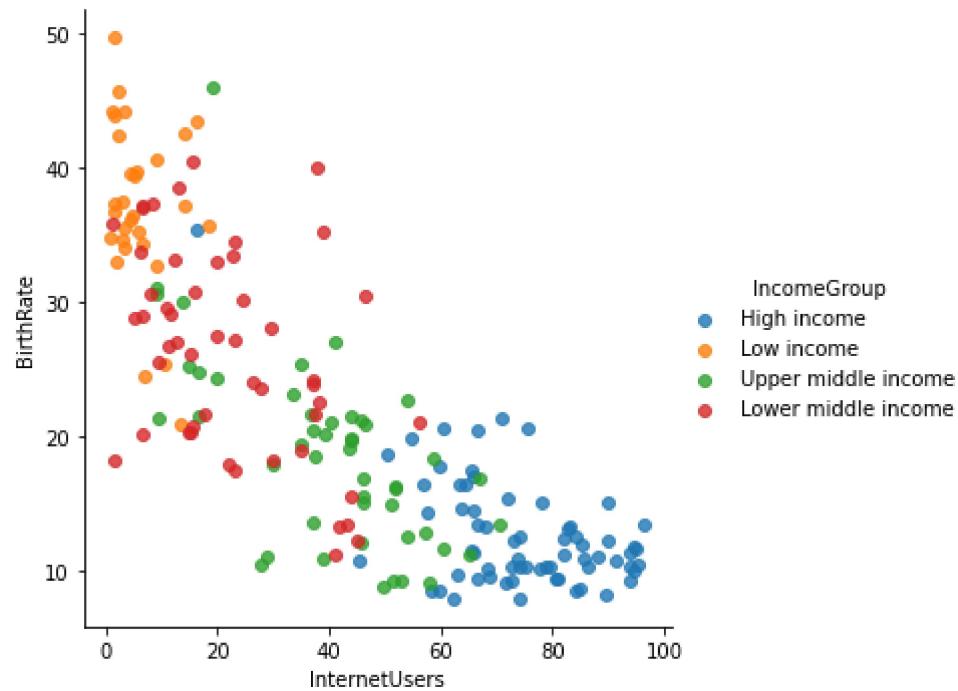
```
In [110]: vis4 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate')
```



```
In [120]: vis4 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate',hue = 'IncomeGroup')
```

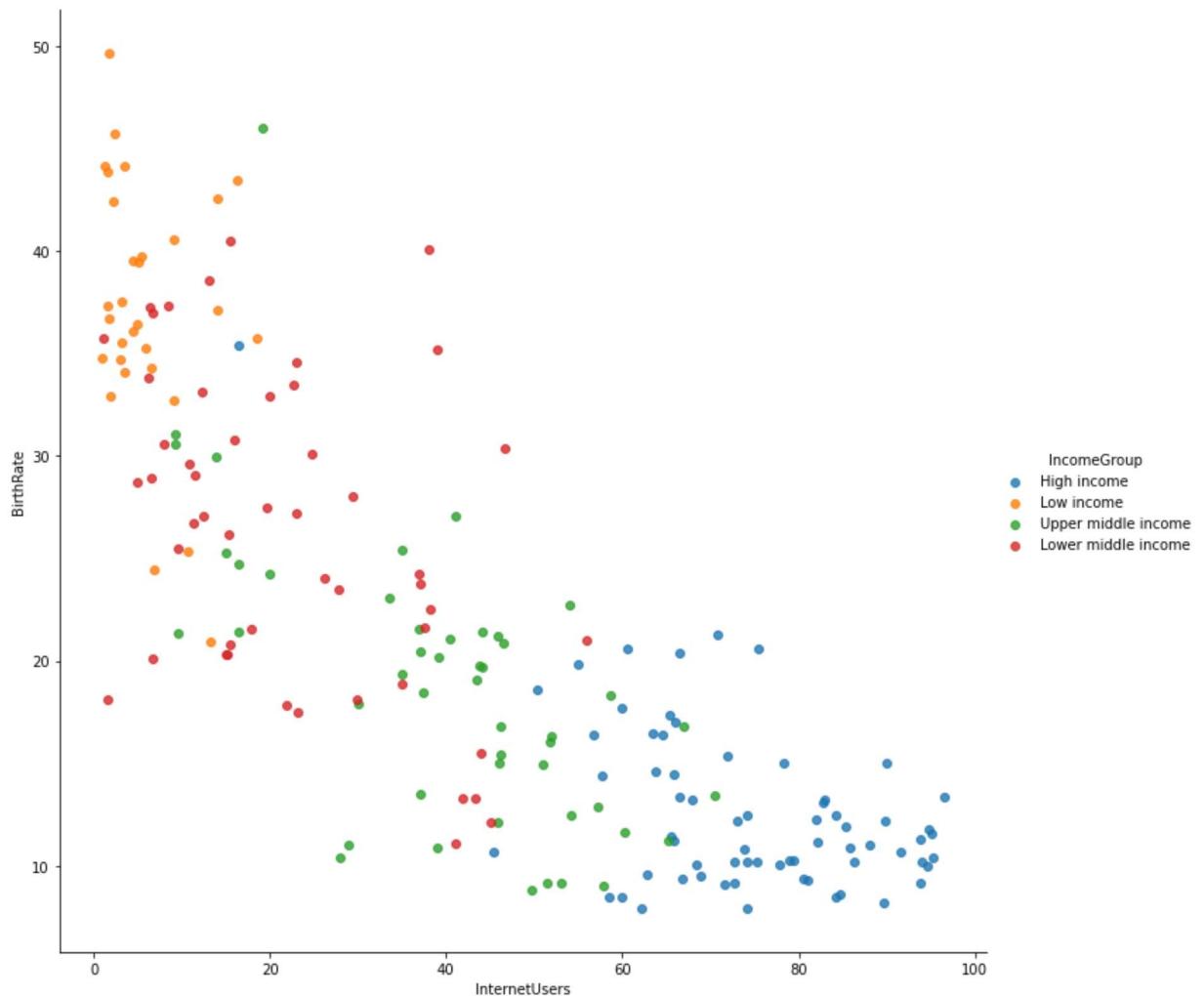


```
In [116]: vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate',  
fit_reg = False,hue = 'IncomeGroup') #hue - parameter for color
```



```
In [119]: vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate',
fit_reg = False,hue = 'IncomeGroup', size = 10)
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn\regression.py:581: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
warnings.warn(msg, UserWarning)



In this section we learned

- 1> importing data into python
- 2> Dataframe via panda
- 3> exploring datasets:
head()tail()info()describe()
- 4> Renaming columns
- 5> subsetting dataframes
- 6> Basic operations with dataframe
- 8> filtering data frames
- 9> seaborn introduction

```
In [ ]:
```

