COUNTRY GDP ANALYSIS USING PANDAS + MATPLOTLIB

In this section what we will learn....

- import data into python
- Dataframe via panda
- exploring datasets: head() tail() info() describe()
- Renaming columns
- subsetting dataframes
- Basic operations with dataframe
- filtering data frames
- seaborn introduction

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"C:\Users\Hp\Desktop\NAYAN\DATA SCIENCE\CSV_FILES\data.cs
```

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]:

```
type(stats)
```

Out[4]:

```
pandas.core.frame.DataFrame
```

with shape, columns, dtype don't use open and close brackets ()

In [5]:

```
len(stats)
```

Out[5]:

```
195
```

In [6]:

```
stats.columns
```

Out[6]:

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

In [7]:

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

Out[7]:

```
5
```

In [8]:

```
stats.head()
```

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
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 [9]:

```
stats.head(2)
```

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

In [10]:

```
stats.tail()
```

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.tail(3)
```

Out[11]:

	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 [12]:

```
# information of the column
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 [13]:

```
# get stats on the columns
stats.describe()
```

Out[13]:

	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 [14]:

```
# this function convert columns to rows
stats.describe().transpose()
```

Out[14]:

	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 [15]:
```
stats.head(2)
```

Out[15]:

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

In [16]:
```
stats.columns
```

Out[16]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [17]:
```
# renaming column name
stats.columns = ['a', 'b', 'c', 'd', 'e']
stats
```

Out[17]:

	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
...	...	...	...	...	...
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 [18]:
```
stats.head()
```

Out[18]:

	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 [19]:
```
stats.columns =  ['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers','
stats.head()
```

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

In [20]:
```
# subsetting a dataframe in pandas
# Rows
# Columns
# Combine the two
```

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

Out[21]:

	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 [22]:
```
stats[:]
```

Out[22]:

	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 [23]:

```
stats[: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]:

```
stats[::-1]  # reversing
```

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]:
```
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]:
```
stats
```

Out[27]:

	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 [28]:
```
stats['CountryName'].head()  # only 'CountryName' with head(5)
```

Out[28]:
```
0                   Aruba
1             Afghanistan
2                  Angola
3                 Albania
4    United Arab Emirates
Name: CountryName, dtype: object
```

In [29]:
```
['CountryName', 'BirthRate']  # it will print as a string
```

Out[29]:
```
['CountryName', 'BirthRate']
```

In [30]:

```
stats[['CountryName', 'BirthRate']]  # don't forget to use square brackets wit
```

Out[30]:

	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 [31]:

```
stats[['CountryName', 'BirthRate']].head()  # write this down in 2-dim
```

Out[31]:

	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 [32]:

```
stats[['CountryName', 'BirthRate', 'InternetUsers']].head()  # write this dowr
```

Out[32]:

	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

In [33]:

```
stats.head()
```

Out[33]:

	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 [34]:

```
stats['BirthRate']
```

Out[34]:

```
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
Name: BirthRate, Length: 195, dtype: float64
```

In [35]:

```
# combine the two column with slicing
stats[4:8][['CountryName', 'BirthRate']]
```

Out[35]:

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

In [36]:

```
stats [['CountryName', 'BirthRate']][4:8]  # subset dataframe
```

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]:

```
df1 = stats [['CountryName', 'BirthRate']]
df1
```

Out[37]:

	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 [38]:

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

Out[38]:

	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 [39]:
```
stats
```

Out[39]:

	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 [40]:
```
stats.BirthRate
```

Out[40]:
```
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
Name: BirthRate, Length: 195, dtype: float64
```

In [41]:
```
stats.BirthRate * stats.InternetUsers
```

Out[41]:
```
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 [42]:

```
# stats['nayan'] = stats.CountryCode + stats.CountryCode
# stats.head(10)
```

In [43]:

```
# Add a new column in a dataframe
stats['myCalc'] = stats.BirthRate * stats. InternetUsers
stats.head()  # new column name 'myCalc' has added
```

Out[43]:

	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 [44]:

```
# remove a column so use drop function
stats.drop('myCalc',axis=1)
```

Out[44]:

	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 [45]:
```
stats = stats.drop('myCalc', axis=1)  # axis =1 means columns
stats
```

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
...	...	...	...	...	...
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 [46]:
```
stats.columns[2]
```

Out[46]:
```
'BirthRate'
```

In [47]:
```
Filter = stats.InternetUsers < 2  # we are checking given condition if its Tru
Filter
```

Out[47]:
```
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 [48]:

```
stats[Filter]  # you can say this is masking that we studied in Numpy
# It will take that row which are False
```

Out[48]:

	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 [49]:

```
Filter2 = stats.BirthRate>40
Filter2
```

Out[49]:

```
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 [50]:

```
stats[Filter2]
```

Out[50]:

	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 [51]:
```
# what if we combine both filter
Filter & Filter2
```

Out[51]:
```
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 [52]:
```
stats[Filter & Filter2]  # here again we used masking (to find common term bet
```

Out[52]:

	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 [53]:
```
stats[(stats.BirthRate >40) & (stats.InternetUsers < 2)]  # only these three v
```

Out[53]:

	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 [54]:
```
stats[(stats.InternetUsers < 2) & (stats.BirthRate >40)]
```

Out[54]:

	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 [55]:
```
stats.head()
```

Out[55]:

	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 [56]:

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

Out[56]:

	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 [57]: stats[stats.IncomeGroup == 'Lower middle income']
```

Out[57]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
6	Armenia	ARM	13.308	41.9000	Lower middle income
15	Bangladesh	BGD	20.142	6.6300	Lower middle income
23	Bolivia	BOL	24.236	36.9400	Lower middle income
27	Bhutan	BTN	18.134	29.9000	Lower middle income
34	Cote d'Ivoire	CIV	37.320	8.4000	Lower middle income
35	Cameroon	CMR	37.236	6.4000	Lower middle income
36	Congo, Rep.	COG	37.011	6.6000	Lower middle income
39	Cabo Verde	CPV	21.625	37.5000	Lower middle income
46	Djibouti	DJI	25.486	9.5000	Lower middle income
51	Egypt, Arab Rep.	EGY	28.032	29.4000	Lower middle income
59	Micronesia, Fed. Sts.	FSM	23.511	27.8000	Lower middle income
62	Georgia	GEO	13.332	43.3000	Lower middle income
63	Ghana	GHA	33.131	12.3000	Lower middle income
71	Guatemala	GTM	27.465	19.7000	Lower middle income
73	Guyana	GUY	18.885	35.0000	Lower middle income
75	Honduras	HND	21.593	17.8000	Lower middle income
79	Indonesia	IDN	20.297	14.9400	Lower middle income
80	India	IND	20.291	15.1000	Lower middle income
91	Kenya	KEN	35.194	39.0000	Lower middle income
92	Kyrgyz Republic	KGZ	27.200	23.0000	Lower middle income
94	Kiribati	KIR	29.044	11.5000	Lower middle income
97	Lao PDR	LAO	27.051	12.5000	Lower middle income
103	Sri Lanka	LKA	17.863	21.9000	Lower middle income
104	Lesotho	LSO	28.738	5.0000	Lower middle income
109	Morocco	MAR	21.023	56.0000	Lower middle income
110	Moldova	MDA	12.141	45.0000	Lower middle income
117	Myanmar	MMR	18.119	1.6000	Lower middle income
121	Mauritania	MRT	33.801	6.2000	Lower middle income
128	Nigeria	NGA	40.045	38.0000	Lower middle income
129	Nicaragua	NIC	20.788	15.5000	Lower middle income
135	Pakistan	PAK	29.582	10.9000	Lower middle income
138	Philippines	PHL	23.790	37.0000	Lower middle income
139	Papua New Guinea	PNG	28.899	6.5000	Lower middle income
150	Sudan	SDN	33.477	22.7000	Lower middle income
151	Senegal	SEN	38.533	13.1000	Lower middle income
153	Solomon Islands	SLB	30.578	8.0000	Lower middle income
155	El Salvador	SLV	17.476	23.1093	Lower middle income
159	Sao Tome and Principe	STP	34.537	23.0000	Lower middle income

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
164	Swaziland	SWZ	30.093	24.7000	Lower middle income
166	Syrian Arab Republic	SYR	24.043	26.2000	Lower middle income
170	Tajikistan	TJK	30.792	16.0000	Lower middle income
172	Timor-Leste	TLS	35.755	1.1000	Lower middle income
179	Ukraine	UKR	11.100	41.0000	Lower middle income
182	Uzbekistan	UZB	22.500	38.2000	Lower middle income
186	Vietnam	VNM	15.537	43.9000	Lower middle income
187	Vanuatu	VUT	26.739	11.3000	Lower middle income
188	West Bank and Gaza	PSE	30.394	46.6000	Lower middle income
189	Samoa	WSM	26.172	15.3000	Lower middle income
190	Yemen, Rep.	YEM	32.947	20.0000	Lower middle income
193	Zambia	ZMB	40.471	15.4000	Lower middle income

In [58]:

```
# How to get the unique category
stats.IncomeGroup.unique()  # it will provide all category that exist in Incon
```

Out[58]:

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

rcParams: It's a dictionary of most matplotlib styling that you set at the start of your notebook and it will apply to all your plots.

In [59]:

```
# Seaborn us a very powerful Advance Visualization
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.rcParams['figure.figsize'] = 8, 4
import warnings
warnings.filterwarnings('ignore')  # os error
```

In [60]:

```
stats.head()
```

Out[60]:

	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

remember when you visualize your graph, you also need to visualize the excel file so that you can understand your graph better and this way your skill for analyzing the graph would improve.

In [61]:
```
"""
plot the graph using 1 variable is called -- univaraite analysis
plot the graph using 2 variables is called bivariate analysis
plot the graph more than 3 varaibles is called multivariant analysis (heatmap)
"""
```

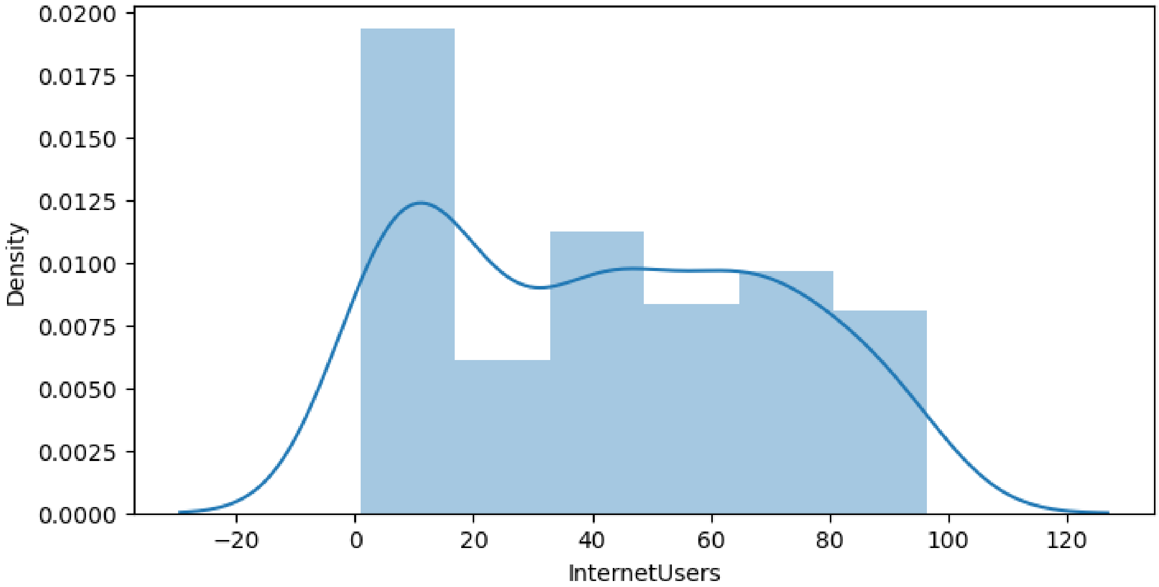
Out[61]:
```
'\nplot the graph using 1 variable is called -- univaraite analysis\nplot th
e graph using 2 variables is called bivariate analysis\nplot the graph more
than 3 varaibles is called multivariant analysis (heatmap)  - This will come
s under EDA\n'
```

Distribution Plot

Univariate Analysis - Statistics

In [62]:
```
vis1 = sns.distplot(stats['InternetUsers'])

# as you can see InternetUsers range from 0.9 to 96.4
```



In [63]:
```
vis1 = plt.distplot(stats['InternetUsers'], bins = 20)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[63], line 1
----> 1 vis1 = plt.distplot(stats['InternetUsers'], bins = 20)

AttributeError: module 'matplotlib.pyplot' has no attribute 'distplot'
```

```
In [ ]: vis1 = sns.distplot(stats['InternetUsers'], bins = 20)
        # bins -- The towers or bars of a histogram are called bins.
```

```
In [ ]: """this blue line is a Distribution line this is created by seaborn"""
```

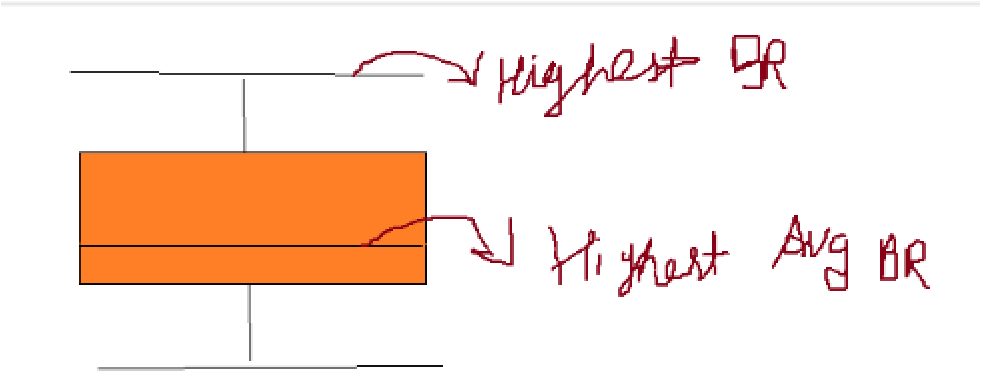
Interpretation / Insights

- in distribution plot more frequency range between 0 to 20 and less frequence range between 80 to 100

Box Plot

Bi-variate Analysis

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



```
In [ ]: # Explain this Graph in High Income
        """
        Firstly focus on x axis -- IncomeGroup and y axis -- BirthRate
        we selected High Income on x axis and then Birthrate from 7 to 35
        # 35 is outlier because all my values under 20 except 35 that's becasue it is
        """
        # in box plot middle line is called average (12.75 is the average of the High
```

Interpretation / Insights

- at High Income: minimum

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

these datapoints are plotted based on data points

In [ ]:

```
vis4 = sns.lmplot(data = stats, x = 'InternetUsers', y = "BirthRate")
# lmplot = (lm = linear model)
# this is negative linear line and this fit_reg is True by default
```

In [ ]:

```
vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate', fit_reg =
```

In [ ]:

```
vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate', fit_reg =
```

Which income group has the highest internetUsers?

- Low income group has the highest Internet Users

Where would you see the highest BirthRate?

- The highest BirthRate at Low income group

In [ ]:

```
vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate', fit_reg =
```

In [ ]:

```
vis5 = sns.lmplot(data = stats,x = 'InternetUsers', y = 'BirthRate', fit_reg =
```

In [ ]:

```
vis1 = sns.displot(stats['InternetUsers'],kind="hist")
```

In [ ]:

In [ ]: