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

In [3]:
```
df = pd.read_csv(r"C:\Users\AKSHAY\OneDrive\Desktop\Code\Projects\Project Codes\
```

In [5]:
```
df
```

Out[5]:

	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 [7]:
```
# HOW TO CHECK THE NUMBER OF COLUMNS AND ROWS IN THE DATASET
```

In [13]:
```
df.shape # this gives the number of rows and columns - dimensions of the dataset
```

Out[13]:
```
(195, 5)
```

In [15]:
```
# CHECKING THE LENGTH OF THE DATASET
```

In [19]:
```
len(df) # this returns the number of rows - length of rows
```

Out[19]:
```
195
```

In [21]:
```
# CHECKING THE NUMBER OF COLUMNS IN THE DATASET
```

In [23]:
```
len(df.columns)
```

Out[23]:
```
5
```

In [25]:
```
# HOW TO CHECK THE COLUMN NAMES IN THE DATASET
```

In [27]:
```
df.columns
```

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

In [29]:
```
# HOW TO CHECK THE TYPE OF THE DATASET
```

In [31]:
```
type(df)
```

Out[31]:
```
pandas.core.frame.DataFrame
```

In [33]:
```
# HOW TO CHECK WHAT DATATYPE, MISSING VALUES etc. ARE THERE IN THE DATASET
```

In [35]:
```
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 [37]:
```
# HOW TO RETURN THE DATA TYPES OF THE VALUES THAT ARE IN THE COLUMN
```

In [42]:
```
df.dtypes
```

Out[42]:
```
CountryName       object
CountryCode       object
BirthRate        float64
InternetUsers    float64
IncomeGroup       object
dtype: object
```

In [66]:
```
# HOW TO RETURN THE STATISTICAL DESCRIPTION (OR) MATHEMATICAL OPERATION OF THE D
```

In [46]:
```
df.describe()
```

Out[46]:

	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 [58]:
```
# HOW TO PRINT THE TOP ROWS OF THE DATASET
```

In [54]:
```
df.head() # it prints first 5 rows by default as there is no parameter given
```

Out[54]:

	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]:
```
df.head(5)
```

Out[56]:

	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 [60]:
```
# HOW TO PRINT THE BOTTOM ROWS OF THE DATASET
```

In [62]:
```
df.tail() # it prints bottom 5 rows by default as there is no parameter given
```

Out[62]:

	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 [64]:
```
df.tail(5)
```

Out[64]:

	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 [72]:
```
# HOW TO INTERCHANGE ROWS WITH COLUMNS AND COLUMNS WITH ROWS, IN THE DECSRIBE FU
```

In [74]:
```
df.describe().transpose() # transpose() - interchanges rows with columns and col
```

Out[74]:

	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 [76]:
```
# HOW TO CHANGE THE COLUMN NAMES IN THE DATASET
```

In [84]:
```
# We replaced the existing column names with a,b,c,d,e
df.columns = ['a','b','c','d','e']
```

In [86]:
```
df.head()
```

Out[86]:

	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 [88]:
```
# Changing back the newly created column names to original ones
df.columns = ['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'Incom
```

In [90]:
```
df.head()
```

Out[90]:

	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 [94]:
```
# PERFOMING INDEXING AND SLICING OPERATIONS ON THE DATASET
```

In [96]:
```
df[21:26]
```

Out[96]:

	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 [100…
```
df[:]
```

Out[100…

	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 [102…
```
df[:10]
```

Out[102…

	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 [104…
```
df[::-1]
```

Out[104...

	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 [110...
```
# HOW TO PRINT THE VALUES OF ONLY 1 PARTICULAR COLUMN
```

In [112...
```
df['CountryName'].head()
```

Out[112...
```
0                  Aruba
1            Afghanistan
2                 Angola
3                Albania
4    United Arab Emirates
Name: CountryName, dtype: object
```

In [114...
```
# HOW TO PRINT THE VALUES OF 2 COLUMNS
```

In [118...
```
df[['CountryName','CountryCode']].head()
```

Out[118...

	CountryName	CountryCode
0	Aruba	ABW
1	Afghanistan	AFG
2	Angola	AGO
3	Albania	ALB
4	United Arab Emirates	ARE

In [120...
```
# HOW TO PERFORM INDEXING AND SLICING OPERATIONS ON PARTICULAR COOLUMNS
```

In [122…
```
df[4:8][['CountryName','CountryCode']]
```

Out[122…

	CountryName	CountryCode
4	United Arab Emirates	ARE
5	Argentina	ARG
6	Armenia	ARM
7	Antigua and Barbuda	ATG

In [124…
```
df[['CountryName','CountryCode']][4:8]
```

Out[124…

	CountryName	CountryCode
4	United Arab Emirates	ARE
5	Argentina	ARG
6	Armenia	ARM
7	Antigua and Barbuda	ATG

In [127…
```
# MATHEMATICAL OPERATIONS ON THE DATASET
```

In [129…
```
df.BirthRate * df.InternetUsers
```

Out[129…
```
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 [131…
```
# HOW TO ADD THIS ABOVE TABLE INTO THE DATASET
```

In [135…
```
# WE NEED TO CREATE A NEW COLUMN AND THE ABOVE VALUES INTO THAT COLUMN AND IN TH
```

In [137…
```
df['mycalc'] = df.BirthRate * df.InternetUsers
```

In [139…
```
df.head()
```

Out[139...

	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 [141...
```
# HOW TO REMOVE A COLUMN
```

In [145...
```
df.drop('mycalc', axis = 1)
```

Out[145...

	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

CASE STUDIES

Case Study - 1

Print the values in the dataset that have satisfies the below condition

Condition - InternetUsers < 2

In [153…
```
df.InternetUsers < 2
```

Out[153…
```
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 [157…
```
# THE ABOVE CODE JUST GIVES THE OUTPUT OF THE CONDITION, IT DOES NOT GIVE THE RE
```

In [159…
```
# TO PRINT THE RECORD WE NEED TO USE THE BELOW CODE
```

In [161…
```
Filter = df.InternetUsers < 2
```

In [163…
```
df[Filter]
```

Out[163…

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
11	Burundi	BDI	44.151	1.3	Low income	57.3963
52	Eritrea	ERI	34.800	0.9	Low income	31.3200
55	Ethiopia	ETH	32.925	1.9	Low income	62.5575
64	Guinea	GIN	37.337	1.6	Low income	59.7392
117	Myanmar	MMR	18.119	1.6	Lower middle income	28.9904
127	Niger	NER	49.661	1.7	Low income	84.4237
154	Sierra Leone	SLE	36.729	1.7	Low income	62.4393
156	Somalia	SOM	43.891	1.5	Low income	65.8365
172	Timor-Leste	TLS	35.755	1.1	Lower middle income	39.3305

Case Study - 2

Print the values in the dataset that have satisfies the below condition

Condition - BirthRate > 40

In [167…
```
df.BirthRate > 40
```

Out[167…
```
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 [169…
```
# THE ABOVE CODE JUST GIVES THE OUTPUT OF THE CONDITION, IT DOES NOT GIVE THE RE
```

In [171…
```
# TO PRINT THE RECORD WE NEED TO USE THE BELOW CODE
```

In [173…
```
Filter2 = df.BirthRate > 40
```

In [175…
```
df[Filter2]
```

Out[175…

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
11	Burundi	BDI	44.151	1.3	Low income	57.3963
14	Burkina Faso	BFA	40.551	9.1	Low income	369.0141
65	Gambia, The	GMB	42.525	14.0	Low income	595.3500
115	Mali	MLI	44.138	3.5	Low income	154.4830
127	Niger	NER	49.661	1.7	Low income	84.4237
128	Nigeria	NGA	40.045	38.0	Lower middle income	1521.7100
156	Somalia	SOM	43.891	1.5	Low income	65.8365
167	Chad	TCD	45.745	2.3	Low income	105.2135
178	Uganda	UGA	43.474	16.2	Low income	704.2788
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	93.2668
193	Zambia	ZMB	40.471	15.4	Lower middle income	623.2534

Case Study - 3

Print the values in the dataset that have satisfies the below condition

Condition - InternetUsers < 2 & BirthRate > 40

In [189…
```
print((df.InternetUsers < 2) & (df.BirthRate > 40))
```

```
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 [191…
```
# THE ABOVE CODE DOES NOT PRINT THE PROPER VALUES
```

In [193…
```
# TO PRINT THE PROPER VALUES WITH RECODRD, WE NEED TO USE THE BELOW CODE
```

In [199…
```
df[Filter & Filter2]
```

Out[199…

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

In [201…
```
# OR WE CAN USE THIS METHOD
```

In [203…
```
df[(df.InternetUsers < 2) & (df.BirthRate > 40)]
```

Out[203…

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

In [205…
```
# HOW TO PRINT A PARTICULAR CONDITION VALUES IN A COLUMN
```

In [207…
```
df[df.IncomeGroup == 'Low income']
```

Out[207…

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
1	Afghanistan	AFG	35.253	5.90	Low income	207.99270
11	Burundi	BDI	44.151	1.30	Low income	57.39630
13	Benin	BEN	36.440	4.90	Low income	178.55600
14	Burkina Faso	BFA	40.551	9.10	Low income	369.01410
29	Central African Republic	CAF	34.076	3.50	Low income	119.26600
38	Comoros	COM	34.326	6.50	Low income	223.11900
52	Eritrea	ERI	34.800	0.90	Low income	31.32000
55	Ethiopia	ETH	32.925	1.90	Low income	62.55750
64	Guinea	GIN	37.337	1.60	Low income	59.73920
65	Gambia, The	GMB	42.525	14.00	Low income	595.35000
66	Guinea-Bissau	GNB	37.503	3.10	Low income	116.25930
77	Haiti	HTI	25.345	10.60	Low income	268.65700
93	Cambodia	KHM	24.462	6.80	Low income	166.34160
99	Liberia	LBR	35.521	3.20	Low income	113.66720
111	Madagascar	MDG	34.686	3.00	Low income	104.05800
115	Mali	MLI	44.138	3.50	Low income	154.48300
120	Mozambique	MOZ	39.705	5.40	Low income	214.40700
123	Malawi	MWI	39.459	5.05	Low income	199.26795
127	Niger	NER	49.661	1.70	Low income	84.42370
132	Nepal	NPL	20.923	13.30	Low income	278.27590
148	Rwanda	RWA	32.689	9.00	Low income	294.20100
154	Sierra Leone	SLE	36.729	1.70	Low income	62.43930
156	Somalia	SOM	43.891	1.50	Low income	65.83650
158	South Sudan	SSD	37.126	14.10	Low income	523.47660
167	Chad	TCD	45.745	2.30	Low income	105.21350
168	Togo	TGO	36.080	4.50	Low income	162.36000
177	Tanzania	TZA	39.518	4.40	Low income	173.87920
178	Uganda	UGA	43.474	16.20	Low income	704.27880
192	Congo, Dem. Rep.	COD	42.394	2.20	Low income	93.26680
194	Zimbabwe	ZWE	35.715	18.50	Low income	660.72750

In [209…
```
df[df.IncomeGroup == 'High income']
```

Out[209…

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
0	Aruba	ABW	10.244	78.90	High income	808.25160
4	United Arab Emirates	ARE	11.044	88.00	High income	971.87200
5	Argentina	ARG	17.716	59.90	High income	1061.18840
7	Antigua and Barbuda	ATG	16.447	63.40	High income	1042.73980
8	Australia	AUS	13.200	83.00	High income	1095.60000
...	...	...	...	...	...	...
174	Trinidad and Tobago	TTO	14.590	63.80	High income	930.84200
180	Uruguay	URY	14.374	57.69	High income	829.23606
181	United States	USA	12.500	84.20	High income	1052.50000
184	Venezuela, RB	VEN	19.842	54.90	High income	1089.32580
185	Virgin Islands (U.S.)	VIR	10.700	45.30	High income	484.71000

67 rows × 6 columns

In [227…
```
# HOW TO DIVIDE CATEGORICAL VALUES FROM NUMERICAL VALUES
```

In [219…
```
df_c = df[['CountryName', 'CountryCode','IncomeGroup']]
df_n = df[[ 'BirthRate', 'InternetUsers']]
```

In [223…
```
df_c
```

Out[223…

	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 [225…
```
df_n
```

Out[225…

	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 [229…
```
# HOW TO GET THE UNIQUE CATEGORIES IN A COLUMN
```

In [231…
```
df.IncomeGroup.unique()
```

Out[231…
```
array(['High income', 'Low income', 'Upper middle income',
       'Lower middle income'], dtype=object)
```

In [233…
```
# HOW TO RETURN THE NUMBER OF UNIQUE CATEGORIES IN A COLUMN
```

In [235…
```
df.IncomeGroup.nunique()
```

Out[235… 4

VISUALIZATION OF THE DATASET

In [242…
```
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
plt.rcParams['figure.figsize'] = 6,4

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

In [244…
```
df.head()
```

Out[244…

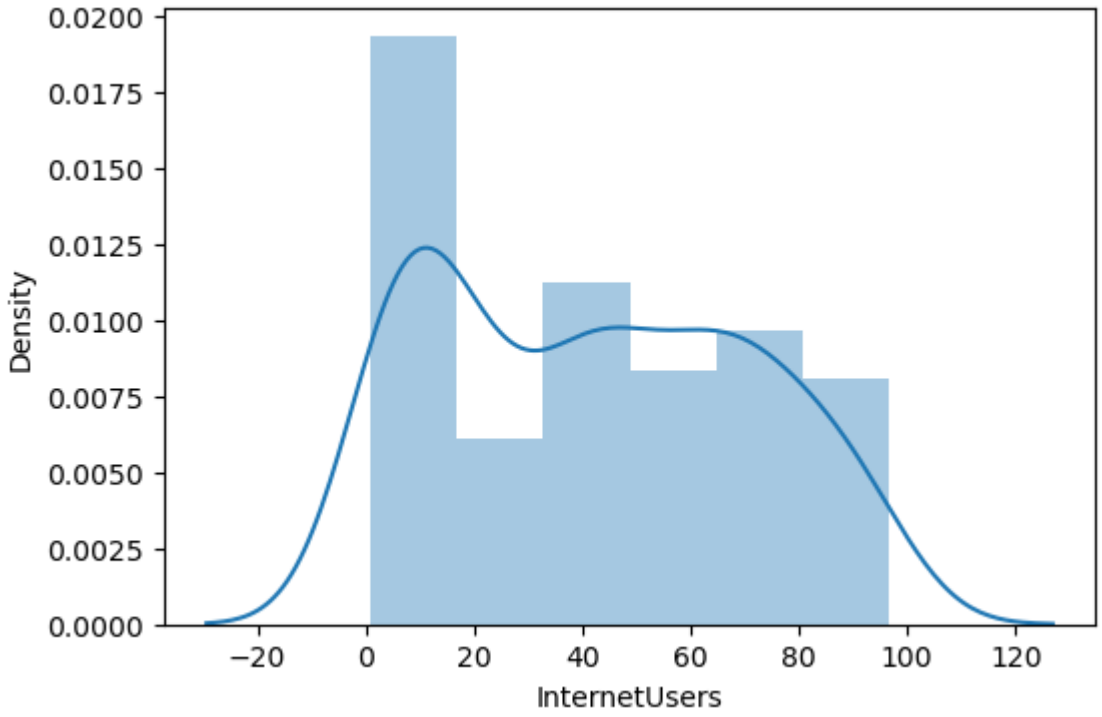
	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	mycalc
0	Aruba	ABW	10.244	78.9	High income	808.2516
1	Afghanistan	AFG	35.253	5.9	Low income	207.9927
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
3	Albania	ALB	12.877	57.2	Upper middle income	736.5644
4	United Arab Emirates	ARE	11.044	88.0	High income	971.8720

In [248…
```
df['InternetUsers']
```

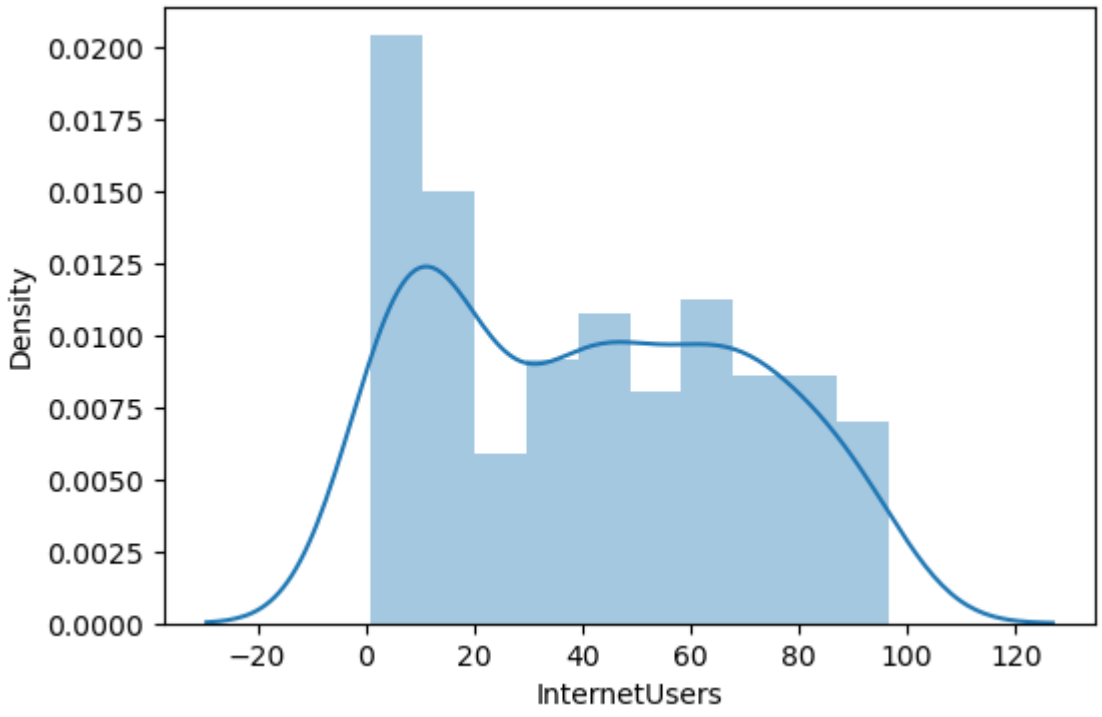
Out[248…
```
0      78.9
1       5.9
2      19.1
3      57.2
4      88.0
       ...
190    20.0
191    46.5
192     2.2
193    15.4
194    18.5
Name: InternetUsers, Length: 195, dtype: float64
```

In [252…
```
# HOW TO RETURN THE DISTPLOT OF A COLUMN
```

In [250…
```
vis1 = sns.distplot(df['InternetUsers'])
plt.show()
```

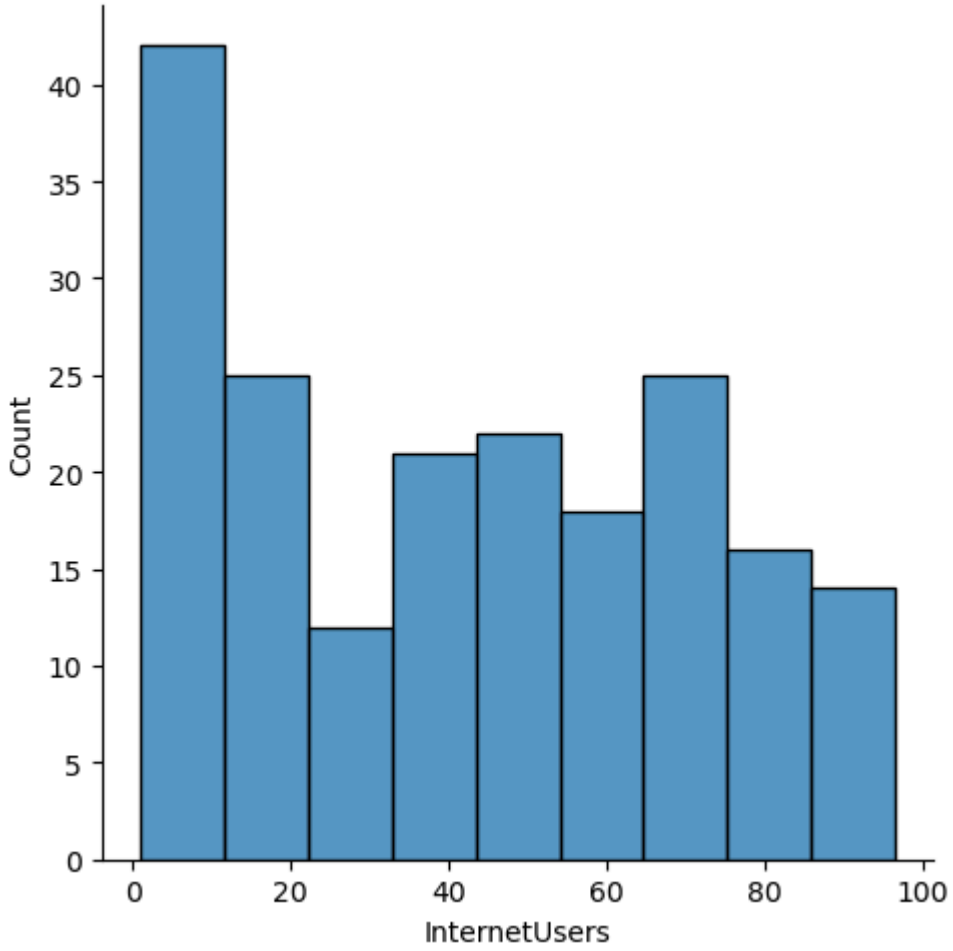


In [266…
```
vis2 = sns.distplot(df['InternetUsers'], bins = 10)
plt.show()
```



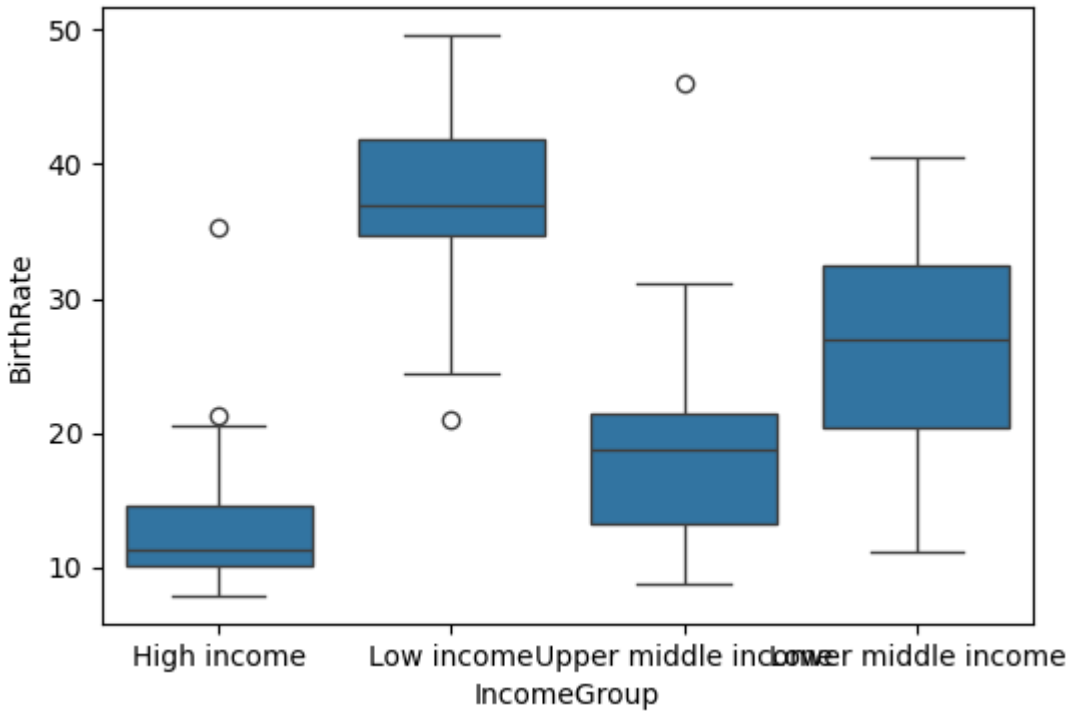
In [254…
```
# HOW TO RETURN THE DISPLOT OF A COLUMN
```

In [268…
```
vis3 = sns.displot(df['InternetUsers'])
plt.show()
```



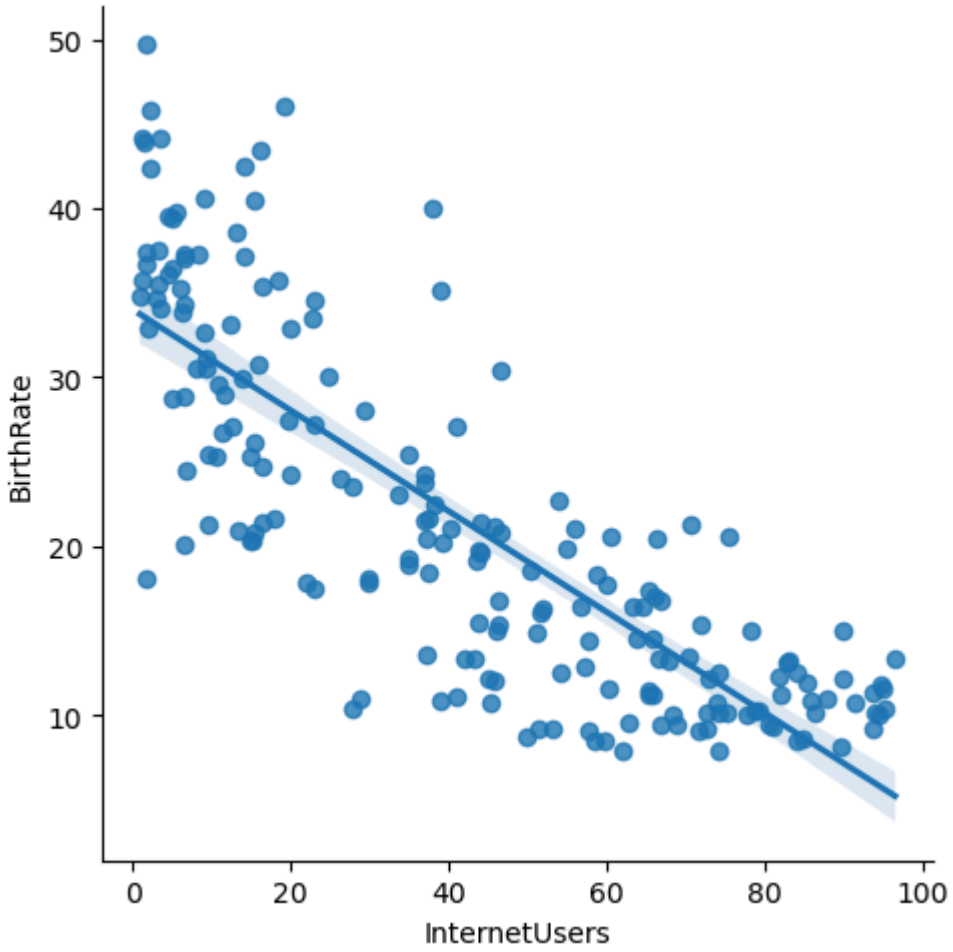
In [264…
```
# HOW TO RETURN THE BOXPLOT OF A DATASET
```

In [282…
```
vis4 = sns.boxplot(data = df, x = 'IncomeGroup', y = 'BirthRate')
plt.show()
```

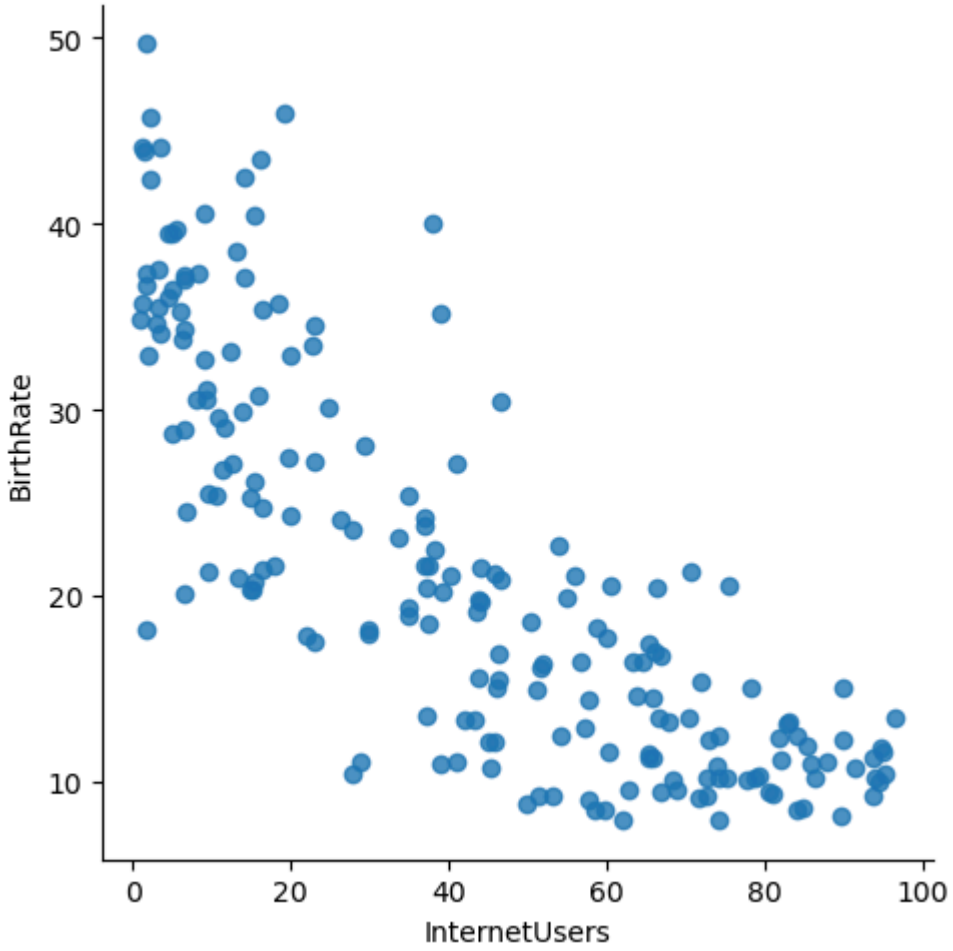


In [284…
```
# HOW TO RETURN THE LMPLOT OF A DATASET
```

In [290…
```
vis5 = sns.lmplot(data = df, x = 'InternetUsers', y = 'BirthRate')
plt.show()
```

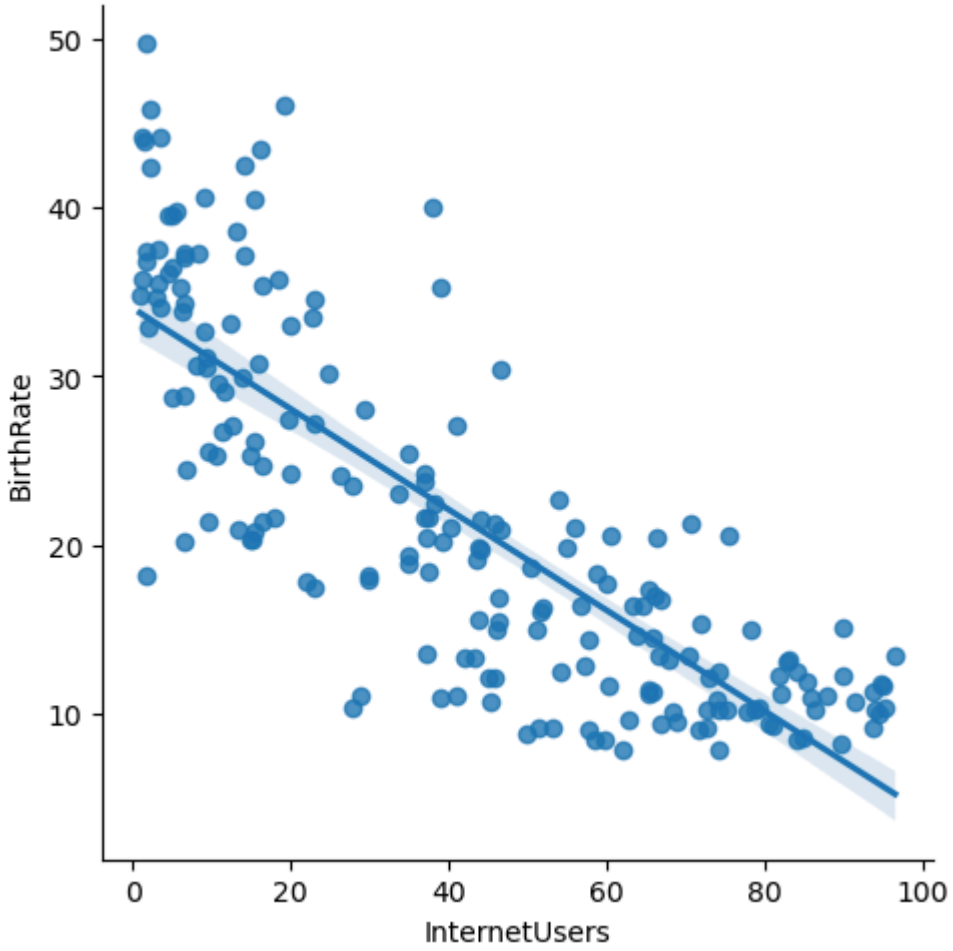


In [292…
```
vis6 = sns.lmplot(data = df, x = 'InternetUsers', y = 'BirthRate', fit_reg = Fal
plt.show() # this removes the line
```



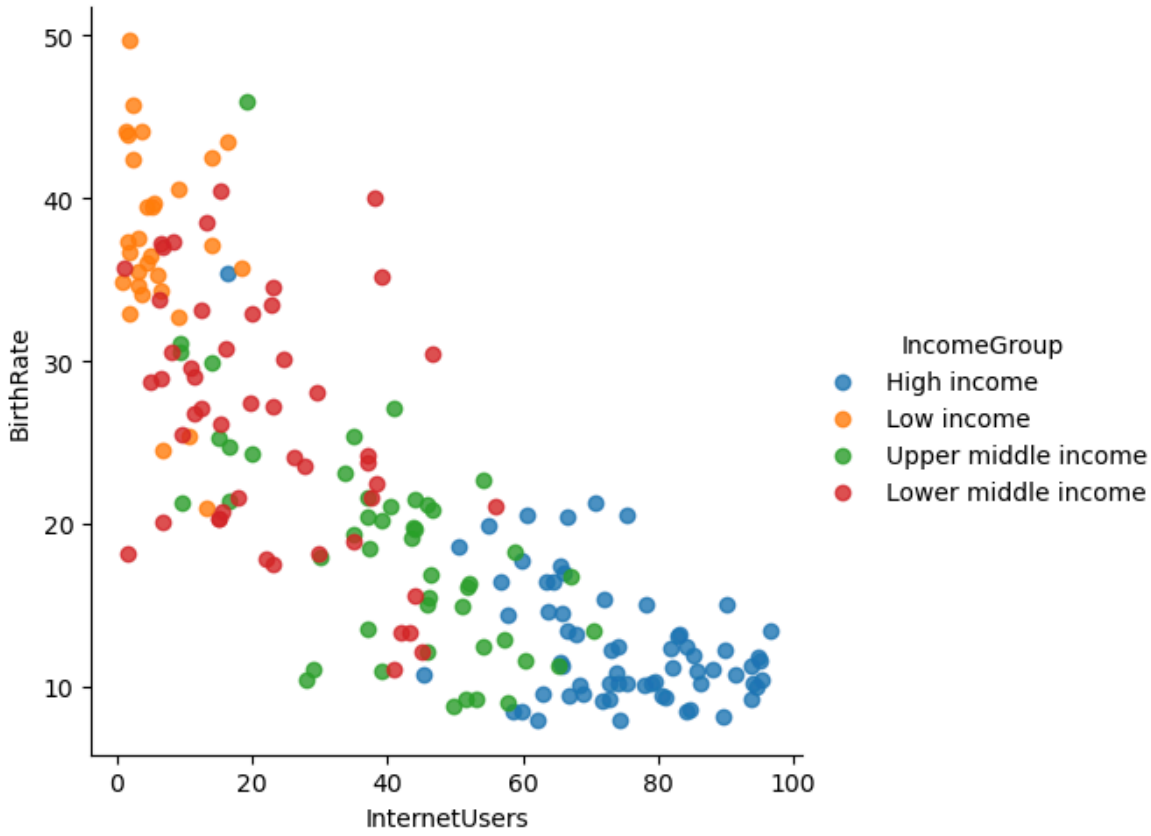
In [296…

```
vis7 = sns.lmplot(data = df, x = 'InternetUsers', y = 'BirthRate', fit_reg = Tru
plt.show() # this brings back the line
```



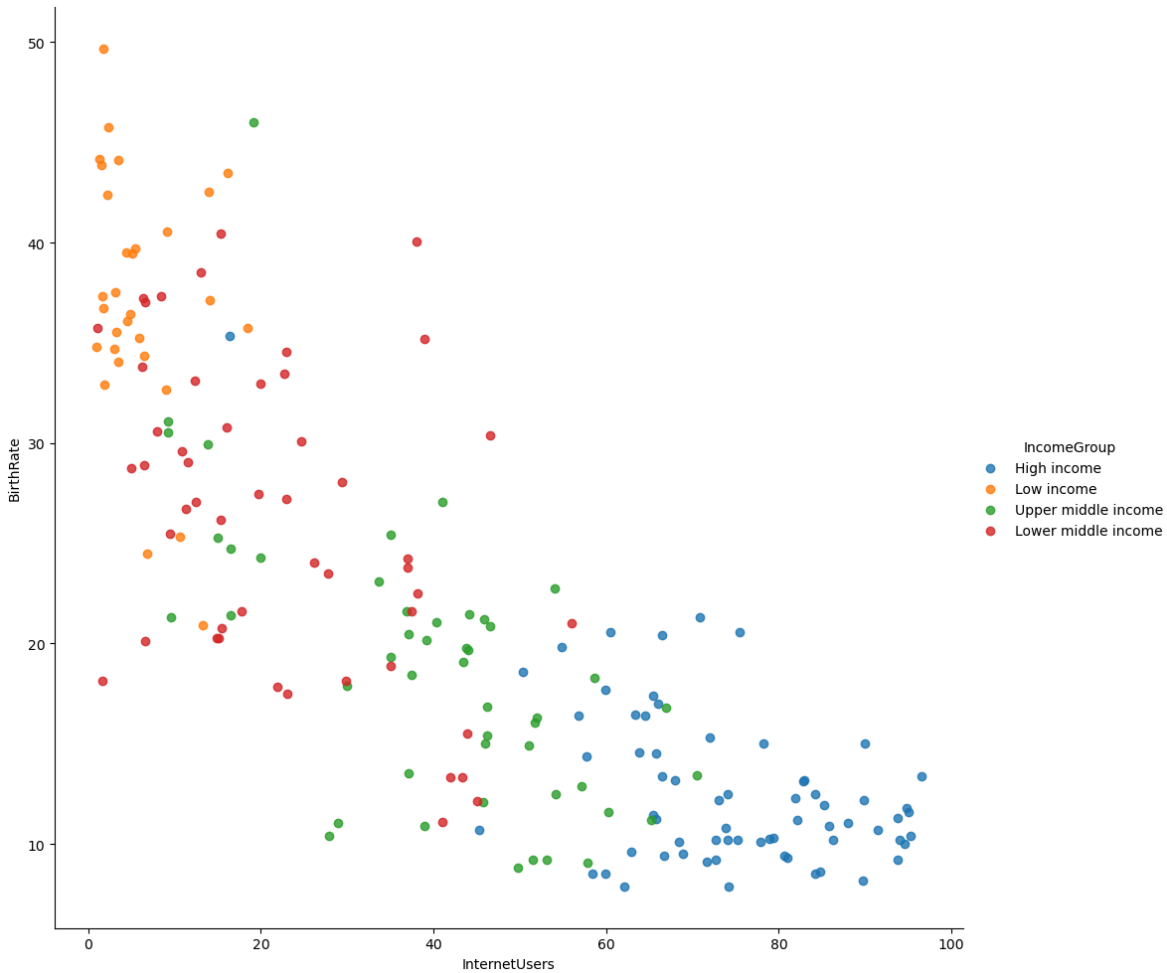
In [298…
```
# HOW TO GIVE MORE INSIGHTS
```

In [302…
```
vis6 = sns.lmplot(data = df, x = 'InternetUsers', y = 'BirthRate', fit_reg = Fal
plt.show()
```



In [320…

```
vis6 = sns.lmplot(data = df, x = 'InternetUsers', y = 'BirthRate', fit_reg = Fal
plt.show()
```



In [ ]: