

# Day 17 & 18: Introduction to Pandas with GDP Dataset

---

## What is Pandas?

Pandas is a powerful Python library for data analysis and manipulation. It provides two primary data structures:

- `Series` : One-dimensional labeled array
- `DataFrame` : Two-dimensional labeled table (like Excel or SQL)

## Why Use Pandas?

- Easy reading/writing from CSV, Excel, SQL
- Powerful data filtering, transformation, and aggregation
- Integrated with NumPy, Matplotlib, Seaborn, etc.

## Understanding Pandas + Its Role in the Data Science Ecosystem

---

### How Pandas Connects with Other Libraries

Pandas is a **core component** of the Python data science stack. Here's how it integrates with the most used tools:

Library	Role	Connection with Pandas
<b>NumPy</b>	Numerical computation	Pandas is built on top of NumPy arrays internally
<b>Matplotlib</b>	Basic plotting and visualization	<code>DataFrame.plot()</code> uses Matplotlib under the hood
<b>Seaborn</b>	Statistical visualizations	Works directly with Pandas DataFrames
<b>Scikit-learn</b>	Machine learning	Accepts Pandas DataFrames as input features

**In short:** Pandas prepares, transforms, and structures the data for these libraries.

---

### Understanding Rows & Columns in Pandas

At the core of Pandas is the `DataFrame`, a 2D table-like structure made up of:

- **Rows:** Individual records (e.g., one country's GDP info)
- **Columns:** Attributes or features (e.g., country, GDP, population)

Think of a `DataFrame` like a spreadsheet or a SQL table.

#### Example Structure:

Country	GDP (in Trillions)	Population (Millions)
USA	23.3	331
India	3.7	1391

Country	GDP (in Trillions)	Population (Millions)
China	17.7	1444

- `df.shape` : tells you the number of (rows, columns)
  - `df.columns` : lists all column names
  - `df.loc[]` : lets you select rows/columns by label
  - `df.iloc[]` : selects by position
- 

## Summary of Common Pandas Operations

Action	Command
Read CSV	<code>pd.read_csv('file.csv')</code>
Show first few rows	<code>df.head()</code>
Get column names	<code>df.columns</code>
Select one column	<code>df['Column']</code>
Filter rows	<code>df[df['GDP'] &gt; 5]</code>
Sort values	<code>df.sort_values('GDP')</code>
Check for nulls	<code>df.isnull().sum()</code>
Summary statistics	<code>df.describe()</code>
Grouping and aggregation	<code>df.groupby('Region').mean()</code>
Export to CSV	<code>df.to_csv('output.csv')</code>

---

## When to Use Pandas

Use Pandas when:

- You're cleaning or exploring tabular datasets (CSV, Excel, SQL)
- You want to quickly filter, sort, or transform columns
- You need to merge multiple datasets (joins)
- You want to prepare data for visualization or ML

Not ideal when:

- You're working with unstructured data (images, audio, text)
- You need distributed processing (use **Dask** or **PySpark** instead)

```
In [1]: import pandas as pd
In [2]: pd.__version__
Out[2]: '2.2.3'
In [3]: df = pd.read_csv(r"C:\Users\Arman\Downloads\dataset\data.csv")
In [4]: df
```

```
Out[4]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

195 rows × 5 columns

```
In [5]: id(df)
```

```
Out[5]: 2485111046896
```

```
In [6]: len(df)
```

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

```
In [7]: df.columns
```

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

```
In [8]: len(df.columns)
```

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

```
In [9]: df.isnull()
```

```
Out[9]:   CountryName  CountryCode  BirthRate  InternetUsers  IncomeGroup
```

0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
190	False	False	False	False	False
191	False	False	False	False	False
192	False	False	False	False	False
193	False	False	False	False	False
194	False	False	False	False	False

195 rows × 5 columns

```
In [10]: df.isna()
```

```
Out[10]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
190	False	False	False	False	False
191	False	False	False	False	False
192	False	False	False	False	False
193	False	False	False	False	False
194	False	False	False	False	False

195 rows × 5 columns

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

```
Out[11]:
```

CountryName	0
CountryCode	0
BirthRate	0
InternetUsers	0
IncomeGroup	0
dtype: int64	

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

```
Out[12]:
```

CountryName	0
CountryCode	0
BirthRate	0
InternetUsers	0
IncomeGroup	0
dtype: int64	

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

```
Out[13]:
```

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

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

Out[14]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

In [15]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   CountryName    195 non-null   object  
 1   CountryCode     195 non-null   object  
 2   BirthRate       195 non-null   float64 
 3   InternetUsers  195 non-null   float64 
 4   IncomeGroup     195 non-null   object  
dtypes: float64(2), object(3)
memory usage: 7.7+ KB
```

In [16]: `df[:]`

Out[16]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
191	South Africa	ZAF	20.850	46.5	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
193	Zambia	ZMB	40.471	15.4	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5	Low income

195 rows × 5 columns

In [17]: `df[1:]`

Out[17]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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
5	Argentina	ARG	17.716	59.9	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

194 rows × 5 columns

In [18]: df[1:11]

Out[18]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
1	Afghanistan	AFG	35.253	5.9000	Low income
2	Angola	AGO	45.985	19.1000	Upper middle income
3	Albania	ALB	12.877	57.2000	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0000	High income
5	Argentina	ARG	17.716	59.9000	High income
6	Armenia	ARM	13.308	41.9000	Lower middle income
7	Antigua and Barbuda	ATG	16.447	63.4000	High income
8	Australia	AUS	13.200	83.0000	High income
9	Austria	AUT	9.400	80.6188	High income
10	Azerbaijan	AZE	18.300	58.7000	Upper middle income

In [19]: df[:::-1]

Out[19]:

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

Out[20]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
1	Afghanistan	AFG	35.253	5.9000	Low income
11	Burundi	BDI	44.151	1.3000	Low income
21	Belize	BLZ	23.092	33.6000	Upper middle income
31	Switzerland	CHE	10.200	86.3400	High income
41	Cuba	CUB	10.400	27.9300	Upper middle income
51	Egypt, Arab Rep.	EGY	28.032	29.4000	Lower middle income
61	United Kingdom	GBR	12.200	89.8441	High income
71	Guatemala	GTM	27.465	19.7000	Lower middle income
81	Ireland	IRL	15.000	78.2477	High income
91	Kenya	KEN	35.194	39.0000	Lower middle income

In [21]: `df.describe()`

Out[21]:

	BirthRate	InternetUsers
count	195.000000	195.000000
mean	21.469928	42.076471
std	10.605467	29.030788
min	7.900000	0.900000
25%	12.120500	14.520000
50%	19.680000	41.000000
75%	29.759500	66.225000
max	49.661000	96.546800

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

```
Out[22]:   CountryName  CountryCode  BirthRate  InternetUsers  IncomeGroup
0          Aruba        ABW      10.244       78.9    High income
```

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

```
Out[23]: 0          Aruba
1          Afghanistan
2          Angola
3          Albania
4          United Arab Emirates
...
190         Yemen, Rep.
191         South Africa
192         Congo, Dem. Rep.
193         Zambia
194         Zimbabwe
Name: CountryName, Length: 195, dtype: object
```

```
In [24]: df['CountryCode']
```

```
Out[24]: 0      ABW
1      AFG
2      AGO
3      ALB
4      ARE
...
190     YEM
191     ZAF
192     COD
193     ZMB
194     ZWE
Name: CountryCode, Length: 195, dtype: object
```

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

```
Out[25]:   CountryName  CountryCode  IncomeGroup
0          Aruba        ABW    High income
1          Afghanistan  AFG    Low income
2          Angola        AGO  Upper middle income
3          Albania        ALB  Upper middle income
4          United Arab Emirates  ARE    High income
...
190         Yemen, Rep.  YEM  Lower middle income
191         South Africa  ZAF  Upper middle income
192         Congo, Dem. Rep.  COD    Low income
193         Zambia        ZMB  Lower middle income
194         Zimbabwe      ZWE    Low income
```

195 rows × 3 columns

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

```
Out[26]:
```

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

195 rows × 3 columns

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

5  
3

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

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

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

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

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

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

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

```
Out[31]: BirthRate  InternetUsers
```

	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 [32]: 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 [33]: df_cat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   CountryName    195 non-null   object  
 1   CountryCode     195 non-null   object  
 2   IncomeGroup     195 non-null   object  
dtypes: object(3)
memory usage: 4.7+ KB
```

```
In [34]: df_num.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   BirthRate     195 non-null   float64 
 1   InternetUsers 195 non-null   float64 
dtypes: float64(2)
memory usage: 3.2 KB
```

```
In [35]: df.describe()
```

```
Out[35]:
```

	<b>BirthRate</b>	<b>InternetUsers</b>
<b>count</b>	195.000000	195.000000
<b>mean</b>	21.469928	42.076471
<b>std</b>	10.605467	29.030788
<b>min</b>	7.900000	0.900000
<b>25%</b>	12.120500	14.520000
<b>50%</b>	19.680000	41.000000
<b>75%</b>	29.759500	66.225000
<b>max</b>	49.661000	96.546800

```
In [36]: df.describe().transpose()
```

```
Out[36]:
```

	<b>count</b>	<b>mean</b>	<b>std</b>	<b>min</b>	<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>max</b>
<b>BirthRate</b>	195.0	21.469928	10.605467	7.9	12.1205	19.68	29.7595	49.6610
<b>InternetUsers</b>	195.0	42.076471	29.030788	0.9	14.5200	41.00	66.2250	96.5468

```
In [37]: df.describe().T
```

```
Out[37]:
```

	<b>count</b>	<b>mean</b>	<b>std</b>	<b>min</b>	<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>max</b>
<b>BirthRate</b>	195.0	21.469928	10.605467	7.9	12.1205	19.68	29.7595	49.6610
<b>InternetUsers</b>	195.0	42.076471	29.030788	0.9	14.5200	41.00	66.2250	96.5468

```
In [38]: df.columns
```

```
Out[38]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
       dtype='object')
```

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

```
In [40]: df.columns
```

```
Out[40]: Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

```
In [41]: df.head()
```

```
Out[41]:
```

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>
<b>0</b>	Aruba	ABW	10.244	78.9	High income
<b>1</b>	Afghanistan	AFG	35.253	5.9	Low income
<b>2</b>	Angola	AGO	45.985	19.1	Upper middle income
<b>3</b>	Albania	ALB	12.877	57.2	Upper middle income
<b>4</b>	United Arab Emirates	ARE	11.044	88.0	High income

```
In [42]: df.head(2)
```

```
Out[42]:
```

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>
<b>0</b>	Aruba	ABW	10.244	78.9	High income
<b>1</b>	Afghanistan	AFG	35.253	5.9	Low income

```
In [43]: df.columns=['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers','IncomeGroup']
```

```
In [44]: df.head()
```

```
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

```
In [45]: df[['CountryName', 'BirthRate', 'InternetUsers']][4:8] #subset
```

```
Out[45]:
```

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

```
In [46]: df[4:8][['CountryCode', 'BirthRate', 'InternetUsers']]
```

```
Out[46]:
```

	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 [47]: df.columns
```

```
Out[47]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',  
               'IncomeGroup'],  
               dtype='object')
```

```
In [48]: df.BirthRate * df.InternetUsers
```

```
Out[48]: 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 [49]: df.head(2)
```

```
Out[49]:
```

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

```
In [50]: df['newcolumn'] = df.BirthRate * df.InternetUsers
```

```
In [51]: df.head(5)
```

```
Out[51]:
```

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

```
In [52]: len(df.columns)
```

```
Out[52]: 6
```

```
In [53]: df = df.drop('newcolumn', axis = 1)
```

```
In [54]: df.head(1)
```

```
Out[54]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
<b>0</b>	Aruba	ABW	10.244	78.9	High income

```
In [55]: df
```

```
Out[55]:
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
<b>0</b>	Aruba	ABW	10.244	78.9	High income
<b>1</b>	Afghanistan	AFG	35.253	5.9	Low income
<b>2</b>	Angola	AGO	45.985	19.1	Upper middle income
<b>3</b>	Albania	ALB	12.877	57.2	Upper middle income
<b>4</b>	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
<b>190</b>	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
<b>191</b>	South Africa	ZAF	20.850	46.5	Upper middle income
<b>192</b>	Congo, Dem. Rep.	COD	42.394	2.2	Low income
<b>193</b>	Zambia	ZMB	40.471	15.4	Lower middle income
<b>194</b>	Zimbabwe	ZWE	35.715	18.5	Low income

195 rows × 5 columns

```
In [56]: df.InternetUsers<2
```

```
Out[56]: 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 [57]: df[df.InternetUsers<2]
```

```
Out[57]:   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 [58]: len(df[df.InternetUsers<2])
```

```
Out[58]: 9
```

```
In [59]: df.BirthRate>40
```

```
Out[59]: 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 [60]: df[df.BirthRate>40]
```

Out[60]:

	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 [61]: `Filter = df.InternetUsers < 2`

In [62]: `Filter2 = df.BirthRate > 40`

In [63]: `df[Filter & Filter2]`

Out[63]:

	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 [64]: `df[df.IncomeGroup == 'High income']`

Out[64]:

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

67 rows × 5 columns

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

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

```
In [67]: df.IncomeGroup.nunique()
```

```
Out[67]: 4
```

```
In [68]: import matplotlib.pyplot as plt #Visualization
import seaborn as sns #Statstic Visualization, Advanced Visualization

%matplotlib inline
```

```
plt.rcParams['figure.figsize'] = 6,2 #rcparam comes from plt library

import warnings
warnings.filterwarnings('ignore') #ignore the os errors
```

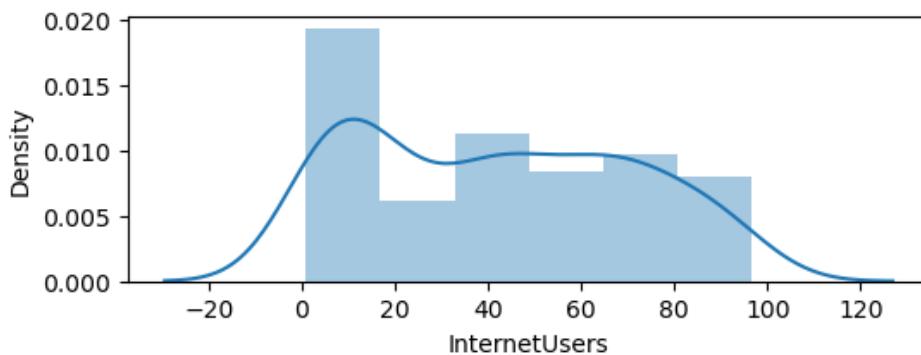
```
In [69]: df.columns
```

```
Out[69]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

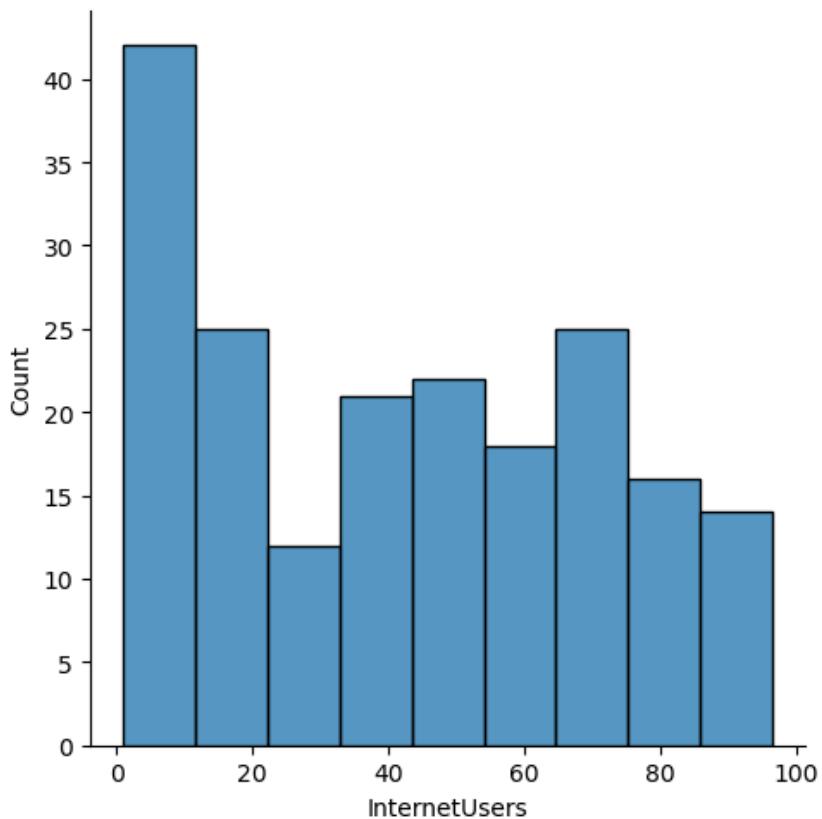
```
In [70]: df['InternetUsers']
```

```
Out[70]: 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
```

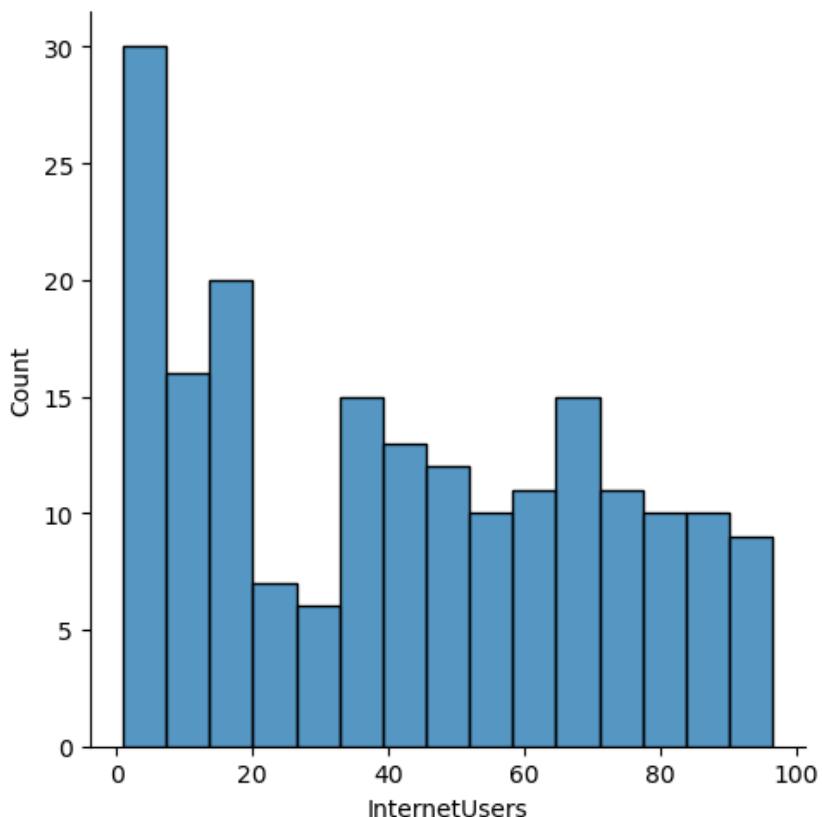
```
In [71]: vis1 = sns.distplot(df['InternetUsers'])
plt.show(vis1)
```



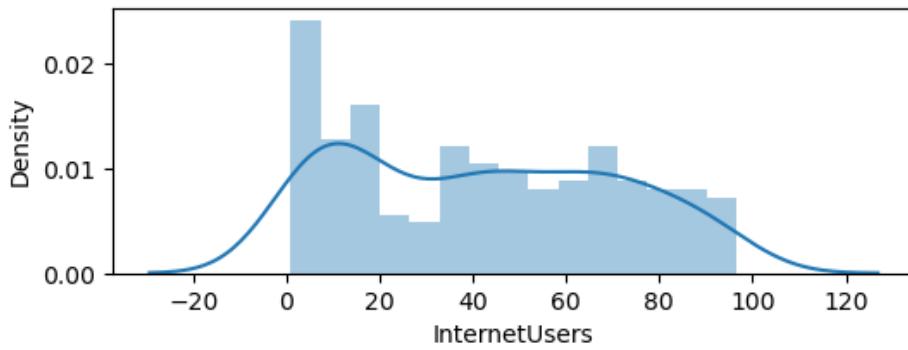
```
In [72]: vis2 = sns.distplot(df['InternetUsers'])
plt.show(vis2)
```



```
In [73]: vis3 = sns.distplot(df['InternetUsers'], bins=15)
plt.show(vis3)
```

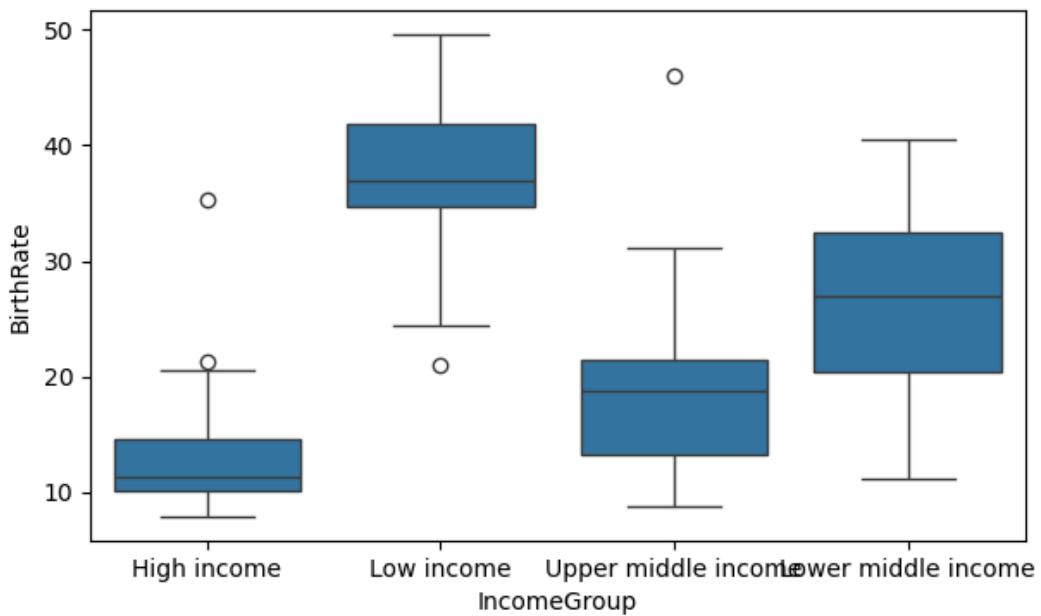


```
In [74]: vis3 = sns.distplot(df['InternetUsers'], bins=15)
plt.show(vis3)
```

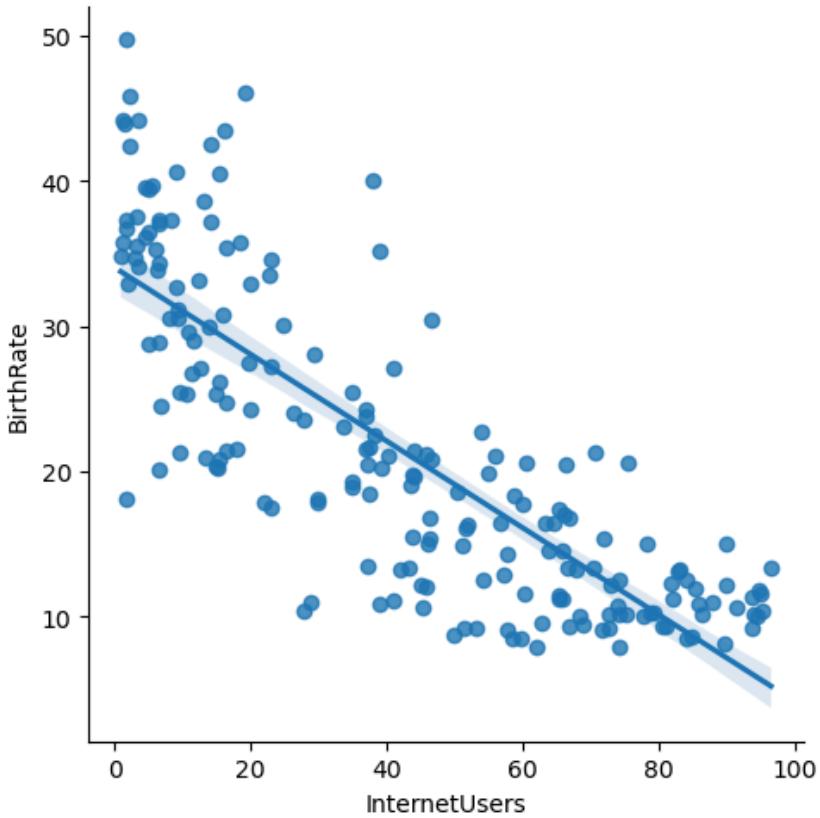


```
In [75]: plt.rcParams['figure.figsize'] = 7,4
```

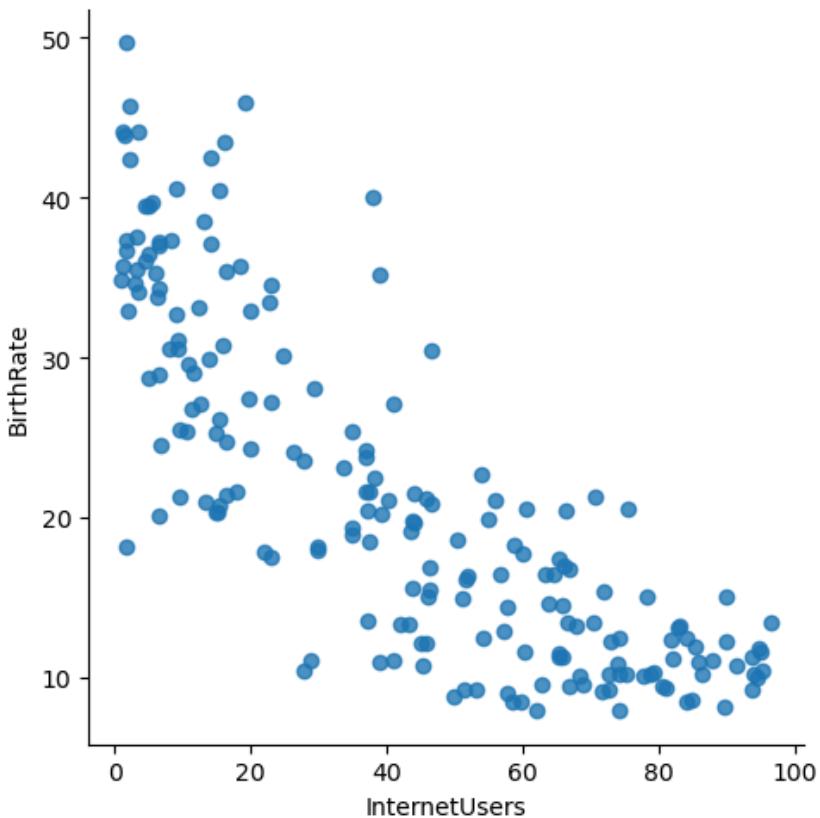
```
In [76]: vis4 = sns.boxplot(data = df ,x= 'IncomeGroup',y = 'BirthRate')
plt.show(vis4)
```



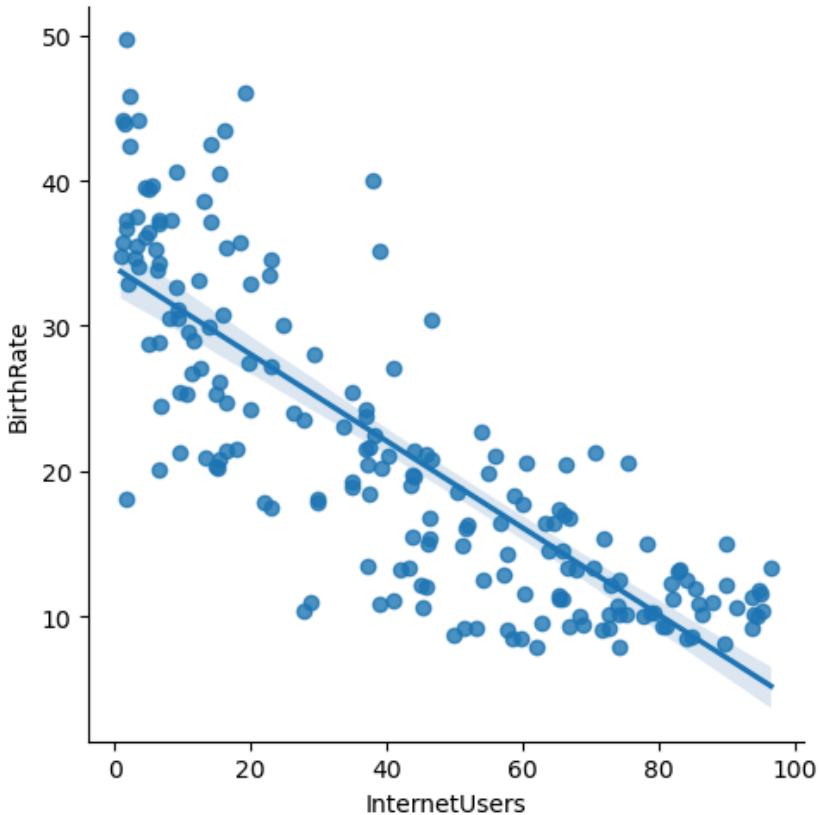
```
In [77]: vis5 = sns.lmplot(data = df ,x= 'InternetUsers',y = 'BirthRate')
plt.show(vis5)
```



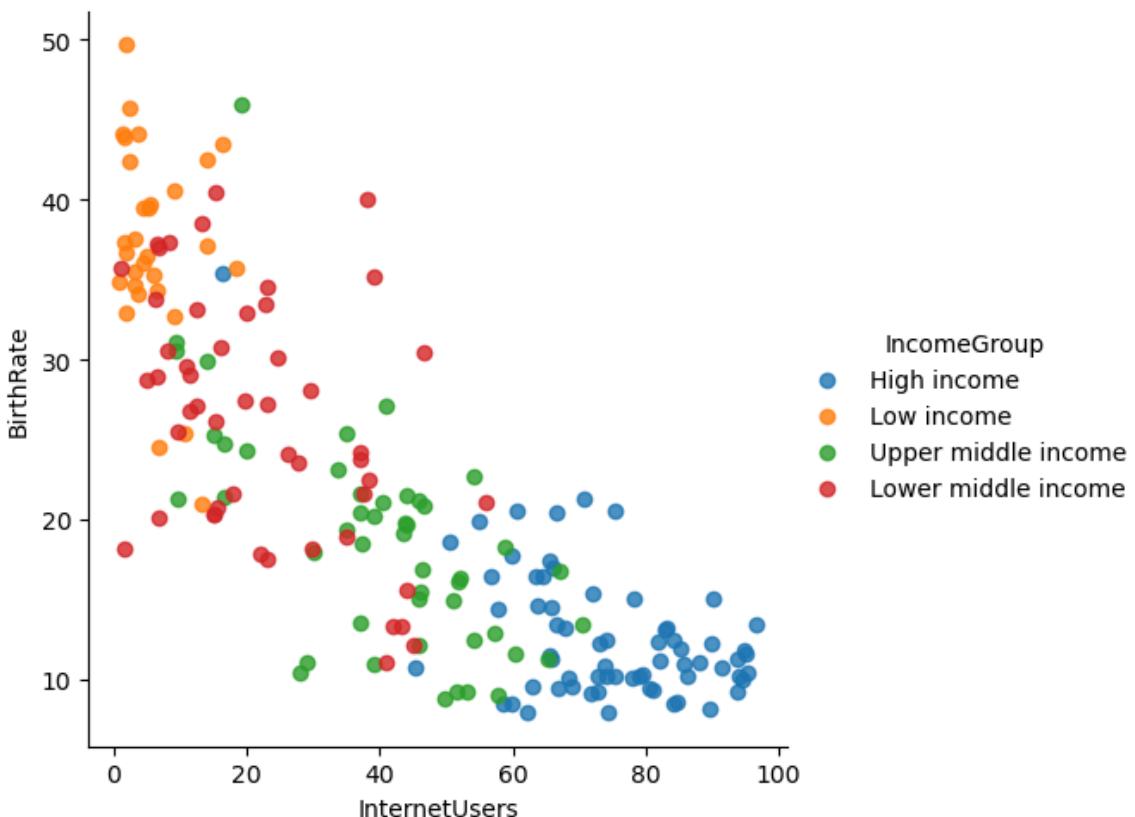
```
In [78]: vis6 = sns.lmplot(data = df ,x= 'InternetUsers',y = 'BirthRate',fit_reg=False)
plt.show(vis6)
```



```
In [79]: vis7 = sns.lmplot(data = df ,x= 'InternetUsers',y = 'BirthRate',fit_reg=True)
plt.show(vis7)
```



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



```
In [81]: vis8=sns.lmplot(data=df,x="InternetUsers",y="BirthRate",fit_reg=True,hue="IncomeGroup")
plt.show(vis8)
```

