

Country GDP Analysis in Python

Mohd Sameer Hussain - mohdsameerhussain28@gmail.com

<https://github.com/SameerHussain128?tab=repositories>

```
# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Import dataset
df=pd.read_csv('data.csv')
```

```
# Gives Top 5 rows
df.head()
```

	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

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
# Gives bottom 5 rows
df.tail()
```

	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

```
# 2 in () gives only top 2 rows
df.head(2)
```

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

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

df

	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 x 5 columns

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
#see Number of columns
len(df)
```

195

```
df.shape
```

(195, 5)

```
#see columns
df.columns
```

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

```
#gives length of the column
len(df.columns)
```

5

```
#see type
type(df)
```

```
pandas.core.frame.DataFrame
def __init__(data=None, index: Axes | None=None, columns: Axes | None=None, dtype: Dtype |
None=None, copy: bool | None=None) -> None
```

[/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py](#)

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).

Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary

```
# Check information of the Dataset
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
```

✓ Checking missing values

```
df.isnull().sum()
```

```
0
CountryName  0
CountryCode  0
BirthRate    0
InternetUsers 0
IncomeGroup  0
```

✓ SLICING

```
# Gives Reverse of rows
```

```
df[::-1]
```

```

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 x 5 columns

```
# Gives 0 to 4 rows - 5 ROWS
```

```
df[: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
```

```
# Gives 6 to rest of rows
```

```
df[6:]
```

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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
...
190	Yemen, Rep.	YEM	32.947	20.0000	Lower middle income
191	South Africa	ZAF	20.850	46.5000	Upper middle income
192	Congo, Dem. Rep.	COD	42.394	2.2000	Low income
193	Zambia	ZMB	40.471	15.4000	Lower middle income
194	Zimbabwe	ZWE	35.715	18.5000	Low income

189 rows × 5 columns

Gives 0 to 199 rows with step of 10
df[0:200:10]

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.900000	High income
10	Azerbaijan	AZE	18.300	58.700000	Upper middle income
20	Belarus	BLR	12.500	54.170000	Upper middle income
30	Canada	CAN	10.900	85.800000	High income
40	Costa Rica	CRI	15.022	45.960000	Upper middle income
50	Ecuador	ECU	21.070	40.353684	Upper middle income
60	Gabon	GAB	30.555	9.200000	Upper middle income
70	Greenland	GRL	14.500	65.800000	High income
80	India	IND	20.291	15.100000	Lower middle income
90	Kazakhstan	KAZ	22.730	54.000000	Upper middle income
100	Libya	LBY	21.425	16.500000	Upper middle income
110	Moldova	MDA	12.141	45.000000	Lower middle income
120	Mozambique	MOZ	39.705	5.400000	Low income
130	Netherlands	NLD	10.200	93.956400	High income
140	Poland	POL	9.600	62.849200	High income
150	Sudan	SDN	33.477	22.700000	Lower middle income
160	Suriname	SUR	18.455	37.400000	Upper middle income
170	Tajikistan	TJK	30.792	16.000000	Lower middle income
180	Uruguay	URY	14.374	57.690000	High income
190	Yemen, Rep.	YEM	32.947	20.000000	Lower middle income

df

	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 x 5 columns

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
# descriptive statistics , describe only numerical info
df.describe()
```

	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

```
# Transpose convert column into rows
df.describe().transpose()
```

	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

df.columns

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

df[0:5]

	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

df[:10]

	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

```
df[['CountryName', 'CountryCode', 'BirthRate']]
```

	CountryName	CountryCode	BirthRate
0	Aruba	ABW	10.244
1	Afghanistan	AFG	35.253
2	Angola	AGO	45.985
3	Albania	ALB	12.877
4	United Arab Emirates	ARE	11.044
...
190	Yemen, Rep.	YEM	32.947
191	South Africa	ZAF	20.850
192	Congo, Dem. Rep.	COD	42.394
193	Zambia	ZMB	40.471
194	Zimbabwe	ZWE	35.715

195 rows x 3 columns

```
df.dtypes
```

CountryName	object
CountryCode	object
BirthRate	float64
InternetUsers	float64
IncomeGroup	object

```
df['BirthRate']
```

**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



df['CountryName']

**CountryName**

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

195 rows × 1 columns



Combain the two

df[4:8][['CountryName','BirthRate']]

**CountryName BirthRate**

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



df[['CountryName','BirthRate']][4:10]

**CountryName BirthRate**

4	United Arab Emirates	11.044
5	Argentina	17.716
6	Armenia	13.308
7	Antigua and Barbuda	16.447
8	Australia	13.200
9	Austria	9.400



df.columns

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

```
df_categorical=df[['CountryName', 'CountryCode', 'IncomeGroup']]
df_categorical.head()
```

	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

Next steps:

[Generate code with df_categorical](#)[View recommended plots](#)[New interactive sheet](#)

df.describe()

	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

df_categorical

	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 x 3 columns

Next steps:

[Generate code with df_categorical](#)[View recommended plots](#)[New interactive sheet](#)

df_categorical.describe()

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

```
# Mathematical Operations
df.BirthRate * df.InternetUsers
```



```
0
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
195 rows x 1 columns

dtype: float64
```

```
# Add a column
df['MyCal']=df.BirthRate * df.InternetUsers
```

df

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	MyCal
0	Aruba	ABW	10.244	78.9	High income	808.2516
1	Afghanistan	AFG	35.253	5.9	Low income	207.9927
2	Angola	AGO	45.985	19.1	Upper middle income	878.3135
3	Albania	ALB	12.877	57.2	Upper middle income	736.5644
4	United Arab Emirates	ARE	11.044	88.0	High income	971.8720
...
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income	658.9400
191	South Africa	ZAF	20.850	46.5	Upper middle income	969.5250
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income	93.2668
193	Zambia	ZMB	40.471	15.4	Lower middle income	623.2534
194	Zimbabwe	ZWE	35.715	18.5	Low income	660.7275

195 rows x 6 columns

Next steps:

Generate code with df

View recommended plots

New interactive sheet

df.head()

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	MyCal
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

Next steps:

Generate code with df

View recommended plots

New interactive sheet

```
# Delete column
df.drop('MyCal',axis=1)
```

	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

```
df=df.drop('MyCal',axis=1)
```

```
df.head()
```

	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

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
df.InternetUsers < 2 # we are checking given condition if its correct true or false
```

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

195 rows × 1 columns

```
Filter = df.InternetUsers < 2
Filter
```

**InternetUsers**

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

195 rows × 1 columns

df[5:10]



	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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

df[40:50]




	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
40	Costa Rica	CRI	15.022	45.9600	Upper middle income
41	Cuba	CUB	10.400	27.9300	Upper middle income
42	Cayman Islands	CYM	12.500	74.1000	High income
43	Cyprus	CYP	11.436	65.4548	High income
44	Czech Republic	CZE	10.200	74.1104	High income
45	Germany	DEU	8.500	84.1700	High income
46	Djibouti	DJI	25.486	9.5000	Lower middle income
47	Denmark	DNK	10.000	94.6297	High income
48	Dominican Republic	DOM	21.198	45.9000	Upper middle income
49	Algeria	DZA	24.738	16.5000	Upper middle income

df[Filter] # it will take that row which are false - Internetusers < 2



	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


df.BirthRate > 40



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

195 rows × 1 columns


```
Filter2= df.BirthRate > 40
Filter2
```



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


195 rows × 1 columns

```
df[Filter2]
```



	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


```
#filter and filter2
Filter & Filter2
```



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


195 rows × 1 columns

```
df[Filter & Filter2]
```




	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

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



	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

```
df.head()
```



	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

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
df[df.IncomeGroup == 'Low income'] # Displays rows who has Low Income
```

	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

```
# Display unique categories of Particular Column
df.IncomeGroup.unique()
```

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

```
df.BirthRate.unique()
```

```
array([10.244, 35.253, 45.985, 12.877, 11.044, 17.716, 13.308, 16.447,
      13.2 , 9.4 , 18.3 , 44.151, 11.2 , 36.44 , 40.551, 20.142,
      9.2 , 15.04 , 15.339, 9.062, 12.5 , 23.092, 10.4 , 24.236,
      14.931, 12.188, 16.405, 18.134, 25.267, 34.076, 10.9 , 10.2 ,
      13.385, 12.1 , 37.32 , 37.236, 37.011, 16.076, 34.326, 21.625,
      15.022, 11.436, 8.5 , 25.486, 10. , 21.198, 24.738, 21.07 ,
      28.032, 34.8 , 9.1 , 10.3 , 32.925, 10.7 , 20.463, 12.3 ,
      23.511, 30.555, 12.2 , 13.332, 33.131, 37.337, 42.525, 37.503,
      35.362, 19.334, 14.5 , 27.465, 17.389, 18.885, 7.9 , 21.593,
      25.345, 20.297, 20.291, 15. , 17.9 , 31.093, 13.4 , 21.3 ,
      13.54 , 27.046, 8.2 , 22.73 , 35.194, 27.2 , 24.462, 29.044,
      8.6 , 20.575, 27.051, 13.426, 35.521, 21.425, 15.43 , 17.863,
      28.738, 10.1 , 11.3 , 11.256, 21.023, 12.141, 34.686, 21.447,
      19.104, 11.222, 44.138, 9.5 , 18.119, 11.616, 24.275, 39.705,
      33.801, 39.459, 16.805, 29.937, 17. , 49.661, 40.045, 20.788,
      11.6 , 20.923, 13.12 , 20.419, 29.582, 19.68 , 20.198, 23.79 ,
      28.899, 9.6 , 10.8 , 21.588, 16.393, 11.94 , 8.8 , 32.689,
      20.576, 33.477, 38.533, 9.3 , 30.578, 36.729, 17.476, 43.891,
      37.126, 34.537, 18.455, 11.8 , 30.093, 18.6 , 24.043, 45.745,
      36.08 , 11.041, 30.792, 21.322, 35.755, 25.409, 14.59 , 19.8 ,
      16.836, 39.518, 43.474, 11.1 , 14.374, 22.5 , 16.306, 19.842,
      15.537, 26.739, 30.394, 26.172, 32.947, 20.85 , 42.394, 40.471,
      35.715])
```

```
df.InternetUsers.unique()
```

```
array([[78.9, 5.9, 19.1, 57.2, 88.],
       [59.9, 41.9, 63.4, 83., 80.6188],
       [58.7, 1.3, 82.1702, 4.9, 9.1],
       [6.63, 53.0615, 90.000397, 72., 57.79],
       [54.17, 33.6, 95.3, 36.94, 51.04],
       [73., 64.5, 29.9, 15., 3.5],
       [85.8, 86.34, 66.5, 45.8, 8.4],
       [6.4, 6.6, 51.7, 6.5, 37.5],
       [45.96, 27.93, 74.1, 65.4548, 74.1104],
       [84.17, 9.5, 94.6297, 45.9, 16.5],
       [40.35368423, 29.4, 0.9, 71.635, 79.4],
       [1.9, 91.5144, 37.1, 81.9198, 27.8],
       [9.2, 89.8441, 43.3, 12.3, 1.6],
       [14., 3.1, 16.4, 59.8663, 35.],
       [65.8, 19.7, 65.4, 74.2, 17.8],
       [66.7476, 10.6, 72.6439, 14.94, 15.1],
       [78.2477, 29.95, 96.5468, 70.8, 58.4593],
       [41., 89.71, 54., 39., 23.],
       [6.8, 11.5, 84.77, 75.46, 12.5],
       [70.5, 3.2, 46.2, 93.8, 21.9],
       [5., 68.4529, 93.7765, 75.2344, 56.],
       [45., 3., 44.1, 43.46, 65.24],
       [68.9138, 60.31, 20., 5.4, 6.2],
       [5.05, 66.97, 13.9, 66., 1.7],
       [38., 15.5, 93.9564, 95.0534, 13.3],
       [82.78, 66.45, 10.9, 44.03, 39.2],
       [37., 62.8492, 73.9, 62.0956, 36.9],
       [56.8, 85.3, 49.7645, 67.97, 9.],
       [60.5, 22.7, 13.1, 81., 8.],
       [23.1093, 1.5, 51.5, 14.1, 37.4],
       [77.8826, 72.6756, 94.7836, 24.7, 50.4],
       [26.2, 2.3, 4.5, 28.94, 16.],
       [9.6, 1.1, 63.8, 43.8, 46.25],
       [4.4, 16.2, 57.69, 84.2, 38.2],
       [52., 54.9, 45.3, 43.9, 11.3],
       [46.6, 15.3, 46.5, 2.2, 15.4],
       [18.5]])
```

```
# Introduction to seaborn # seaborn is very powerfull visualizatio(STATISTIC VISULAIZATION) pkg in python
```

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

```
# seaborn are used for advance visualization e.x --> distribution plot, line plot
```

```
%matplotlib inline
plt.rcParams['figure.figsize'] = 6,2
```

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

```
df.head()
```

```

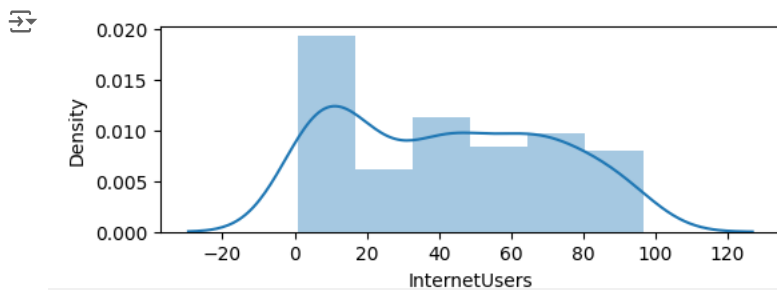
CountryName CountryCode BirthRate InternetUsers IncomeGroup
0          Aruba        ABW   10.244           78.9    High income
1    Afghanistan        AFG   35.253           5.9    Low income
2          Angola        AGO   45.985          19.1  Upper middle income
3        Albania        ALB   12.877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

Next steps:

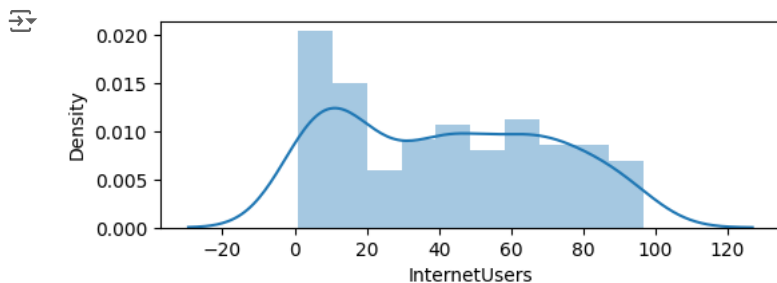
[Generate code with df](#)
[View recommended plots](#)
[New interactive sheet](#)

✓ VISUALIZATION

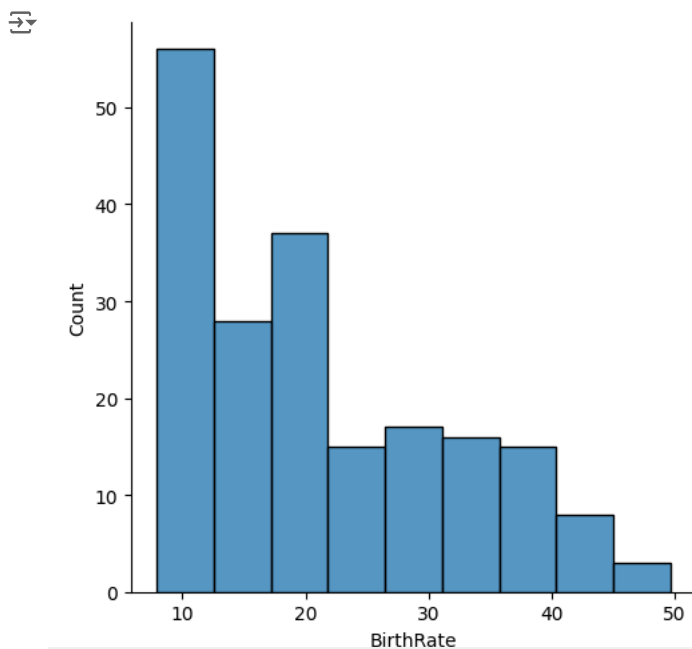
```
# Distributions :
vis1 = sns.distplot(df["InternetUsers"]) # Univariate - Plot the graph using one variable
```



```
vis1 = sns.distplot(df["InternetUsers"],bins=10)
```



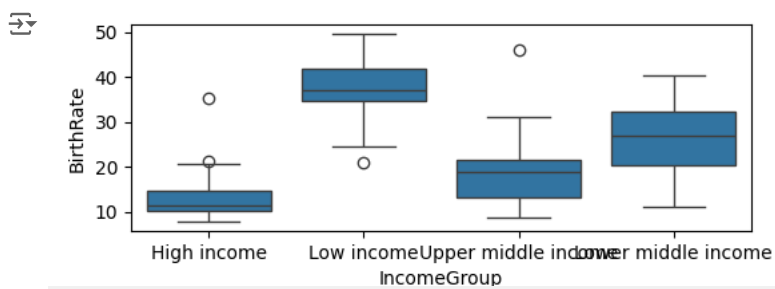
```
vis2 = sns.distplot(df["BirthRate"])
```



```
# Box plots
```

```
vis3 = sns.boxplot(data=df,x="IncomeGroup",y="BirthRate") #bivariate analysis -- plot the graph using two variables
```

```
# small dots are called outliers == anomaly detections -- Statistic outliers is the datapoint which is very far from the other observati
```



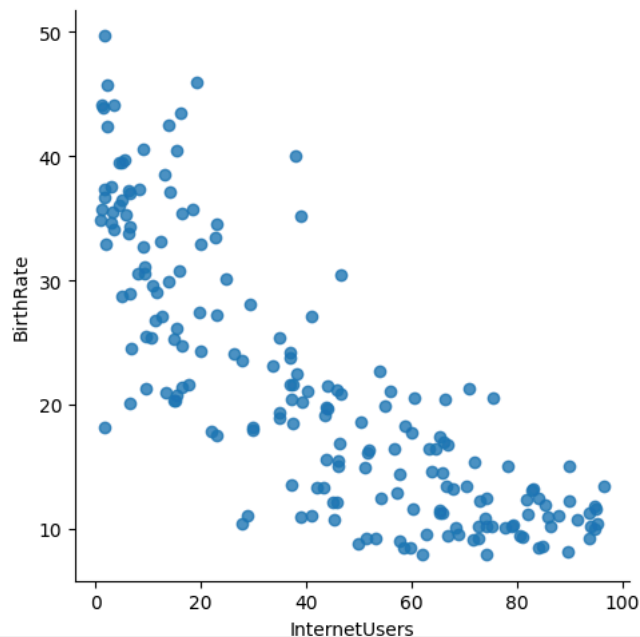
```
df["IncomeGroup"].value_counts()
```



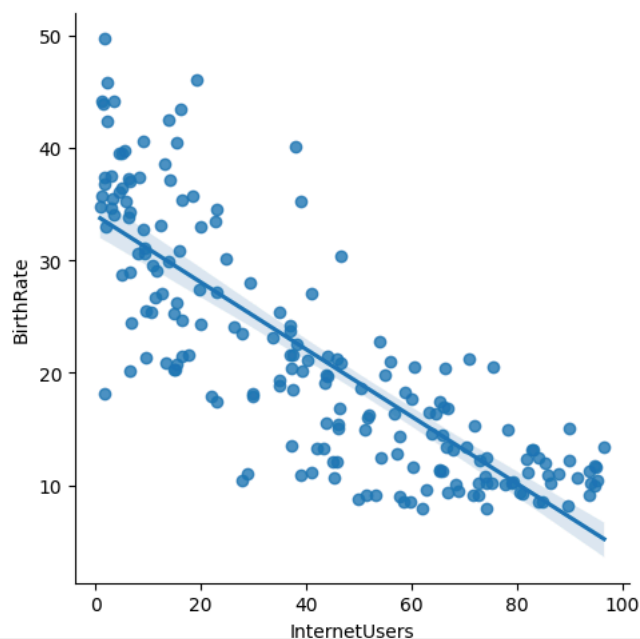

	count
IncomeGroup	
High income	67
Lower middle income	50
Upper middle income	48
Low income	30

```
# refer to seaborn gallery
# visualizatuon with seaborn
```

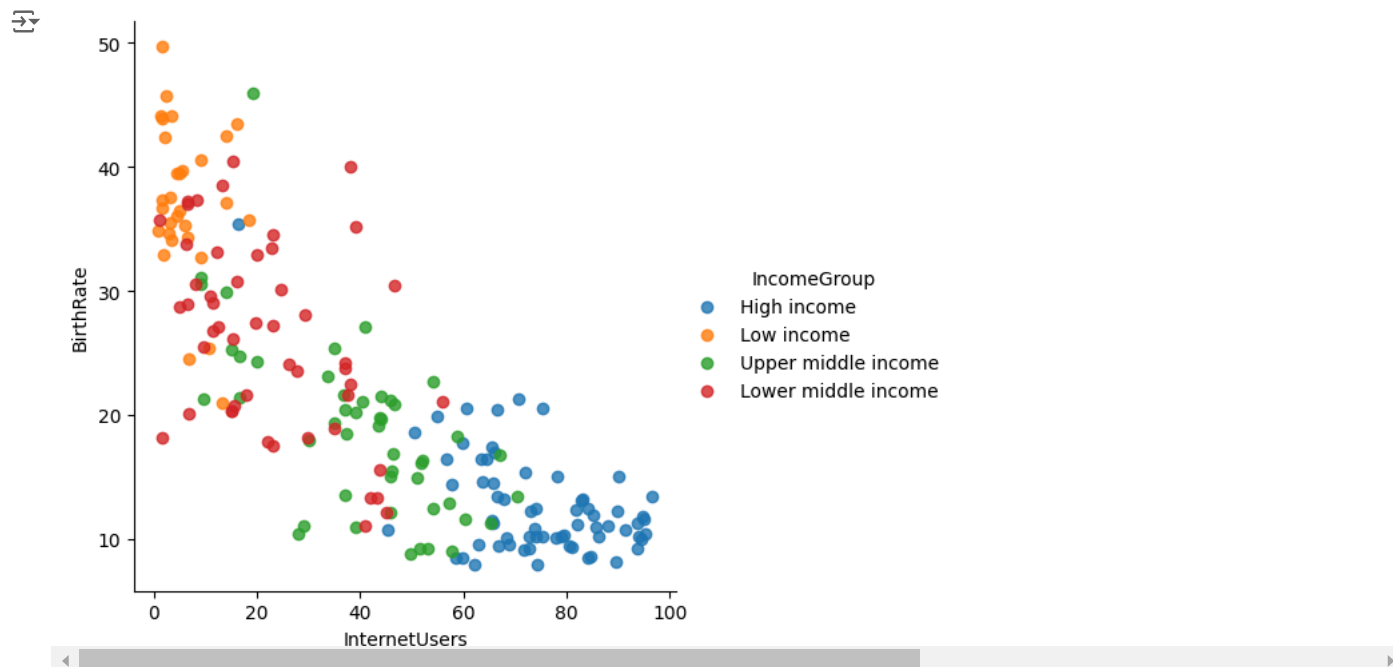
```
vis4= sns.lmplot(data=df,x="InternetUsers",y="BirthRate",fit_reg=False)
#lm -- linear model
```



```
vis5=sns.lmplot(data=df,x="InternetUsers",y="BirthRate")
```



```
vis6 = sns.lmplot(data=df,x="InternetUsers",y="BirthRate",fit_reg=False,hue="IncomeGroup")
# hue -- parameter for color
```



df

	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 x 5 columns

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
vis7=sns.lmplot(data=df,x="InternetUsers",y="BirthRate",fit_reg=False,hue="CountryName")
```



CountryName

- Aruba
- Afghanistan
- Angola
- Albania
- United Arab Emirates
- Argentina
- Armenia
- Antigua and Barbuda
- Australia
- Austria
- Azerbaijan
- Burundi
- Belgium
- Benin
- Burkina Faso
- Bangladesh
- Bulgaria
- Bahrain
- Bahamas, The
- Bosnia and Herzegovina
- Belarus
- Belize
- Bermuda
- Bolivia
- Brazil
- Barbados
- Brunei Darussalam
- Bhutan
- Botswana
- Central African Republic
- Canada
- Switzerland
- Chile
- China
- Cote d'Ivoire
- Cameroon
- Congo, Rep.
- Colombia
- Comoros
- Cabo Verde
- Costa Rica
- Cuba
- Cayman Islands
- Cyprus
- Czech Republic
- Germany
- Djibouti
- Denmark
- Dominican Republic
- Algeria
- Ecuador
- Egypt, Arab Rep.
- Eritrea
- Spain
- Estonia
- Ethiopia
- Finland
- Fiji
- France
- Micronesia, Fed. Sts.
- Gabon
- United Kingdom
- Georgia
- Ghana
- Guinea
- Gambia, The
- Guinea-Bissau
- Equatorial Guinea
- Greece
- Grenada
- Greenland
- Guatemala
- Guam

