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

In [2]:
```
pd.__version__
```

Out[2]:
```
'2.2.2'
```

In [3]:
```
df = pd.read_csv(r"C:\Users\Vansh\Downloads\data.csv")
df
```

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]:
```
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]:

```
# Explore data in python
#1. Full dataframe
#2. How many rows & columns. you have to chk the row becuase the no. of raw shou

len(df) #195 rows imported (this is for tracking later part )
```

Out[5]:

```
195
```

In [6]:

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

Out[6]:

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

In [7]:

```
#4.Number of columns

len(df.columns)
```

Out[7]:

```
5
```

In [8]:

```
#5.top rows

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

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]:

```
df.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]:

```
#6. Bottom rows
df.tail() #last 5 rows
```

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]:

```
df.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]:

```
#7. information of the column

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

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

In [13]:

```
#8. get stats on the columns
df.describe() #it will work like a statistic fun
```

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]:

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

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]:

```
# Renaming columns of a dataframe

df.head()
```

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
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 [16]:

```
df.columns
```

Out[16]:

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

In [17]:

```
df.columns = ['a','b','c','d','e']
df.head()
```

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

In [18]:

```
df.columns = ['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers','Income
df
```

Out[18]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
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 [19]:

```
df.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 dataframes in pandas

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

In [21]:

```
# Rows:

df[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]:

```
df[:]
```

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]:

```
df[: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]:

```
df.head(10)
```

Out[24]:

	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 [25]:

```
# How to reverse the dataframe

df[ : : -1]
```

Out[25]:

	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 [26]:

```
df
```

Out[26]:

	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 [27]:

```
# get only every 20th row
df[: : 20]
```

Out[27]:

	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 [28]:

```
# COLUMNS:
df.columns
```

Out[28]:

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

In [29]:

```
df.head()
```

Out[29]:

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

In [30]:

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

Out[30]:

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

In [31]:

```
['CountryName','BirthRate']
```

Out[31]:

```
['CountryName', 'BirthRate']
```

In [32]:

```
df[['CountryName','BirthRate']].head()
```

Out[32]:

	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 [33]:

```
df.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]:

```
df['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

df[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]:

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

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 = df[['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 = df[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]:

```
# Basic operation of dataframe
df.head()
```

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

In [40]:

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

Out[40]:

	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 [41]:

```
df.head()
```

Out[41]:

	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 [42]:

```
# Mathmetical operation =
df.BirthRate * df.InternetUsers
```

Out[42]:

```
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 [43]:

```
# Add a column
df['myCalc'] = df.BirthRate * df.InternetUsers
df.head()
```

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]: df.columns
```

```
Out[44]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
                'IncomeGroup', 'myCalc'],
               dtype='object')
```

```
In [45]: df['myCalc']
```

```
Out[45]: 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
         Name: myCalc, Length: 195, dtype: float64
```

```
In [46]: df.myCalc
```

```
Out[46]: 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
         Name: myCalc, Length: 195, dtype: float64
```

```
In [47]: df
```

Out[47]:

	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
...	...	...	...	...	...	...
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 × 6 columns

In [48]:

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

Out[48]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
0	Aruba	ABW	10.244	78.9	High income
1	Afghanistan	AFG	35.253	5.9	Low income
2	Angola	AGO	45.985	19.1	Upper middle income
3	Albania	ALB	12.877	57.2	Upper middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
...	...	...	...	...	...
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 [49]:

```
df.head()
```

Out[49]:

	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 [50]:

```
df.columns[2]
```

Out[50]:

```
'BirthRate'
```

In [51]:

```
df.InternetUsers<2 #we are checking if the given condition istrue or false
```

Out[51]:

```
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 [52]:

```
filter = df.InternetUsers<2
filter
```

Out[52]:

```
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 [53]:

```
df[3:7]
```

Out[53]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	myCalc
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
5	Argentina	ARG	17.716	59.9	High income	1061.1884
6	Armenia	ARM	13.308	41.9	Lower middle income	557.6052

In [54]:

```
df[30:40]
```

Out[54]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup	myCalc
30	Canada	CAN	10.900	85.80	High income	935.2200
31	Switzerland	CHE	10.200	86.34	High income	880.6680
32	Chile	CHL	13.385	66.50	High income	890.1025
33	China	CHN	12.100	45.80	Upper middle income	554.1800
34	Cote d'Ivoire	CIV	37.320	8.40	Lower middle income	313.4880
35	Cameroon	CMR	37.236	6.40	Lower middle income	238.3104
36	Congo, Rep.	COG	37.011	6.60	Lower middle income	244.2726
37	Colombia	COL	16.076	51.70	Upper middle income	831.1292
38	Comoros	COM	34.326	6.50	Low income	223.1190
39	Cabo Verde	CPV	21.625	37.50	Lower middle income	810.9375

In [55]:

```
df[filter] #it will take that row which are false
```

Out[55]:

	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

In [56]:

```
df.BirthRate>40
```

Out[56]:

```
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 [57]:

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

Out[57]:

```
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 [58]:

```
df[Filter2]
```

Out[58]:

	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

In [67]:
```
#filter & filter2
filter & Filter2
```

Out[67]:
```
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 [69]:
```
df[filter & filter2]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[69], line 1
----> 1 df[filter & filter2]

NameError: name 'filter2' is not defined
```

In [71]:
```
df[(df.BirthRate > 40) & (df.InternetUsers < 2)]
```

Out[71]:

	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 [73]:
```
df.head()
```

Out[73]:

	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 [75]:
```
df[df.IncomeGroup == 'low income']
df
```

Out[75]:

	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
...	...	...	...	...	...	...
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 × 6 columns

In [77]:

```
#how to get the unique categories
df.IncomeGroup.unique()
```

Out[77]:

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

In [79]:

```
#introduction to seaborn #seaborn is very powerful visualization (stastical visu
import matplotlib.pyplot as plt #visualization
import seaborn as sns #distribution visualization
%matplotlib inline
plt.rcParams['figure.figsize'] = 8,4
#import warnings
#warnings.filterwarnings('ignore')
```

In [81]:

```
df.head()
```

Out[81]:

	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 [83]:

```
#distributions:
vis1 = sns.distplot(df["InternetUsers"])
```

```
C:\Users\Vansh\AppData\Local\Temp\ipykernel_4660\1496636179.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

In [85]:

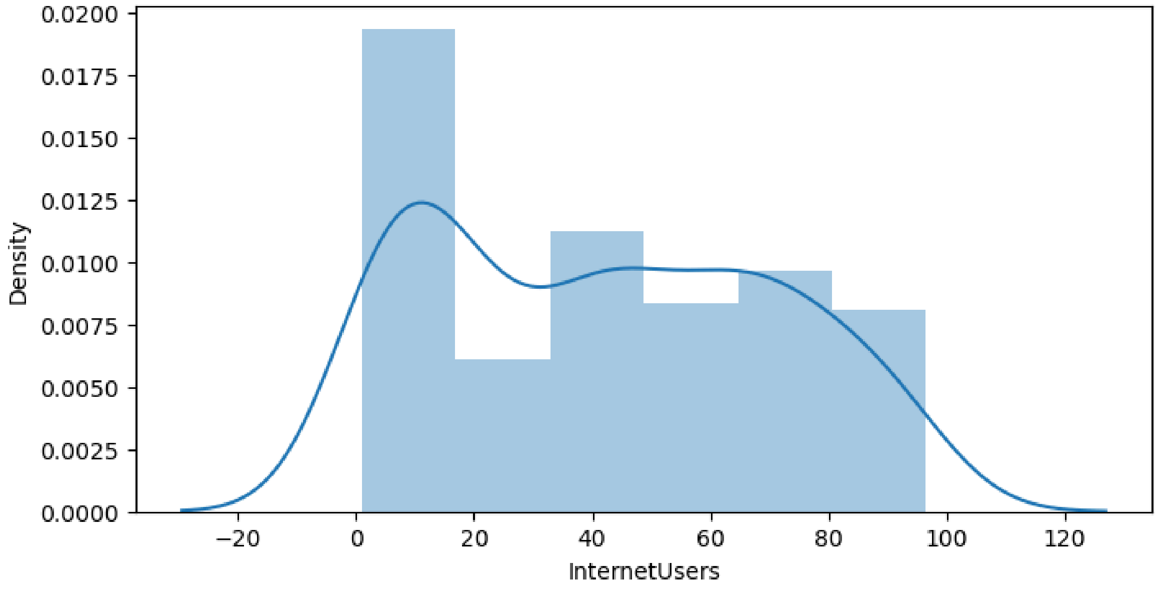
```
vis1
```

Out[85]:

```
<Axes: xlabel='InternetUsers', ylabel='Density'>
```

In [87]:

```
plt.show()
```

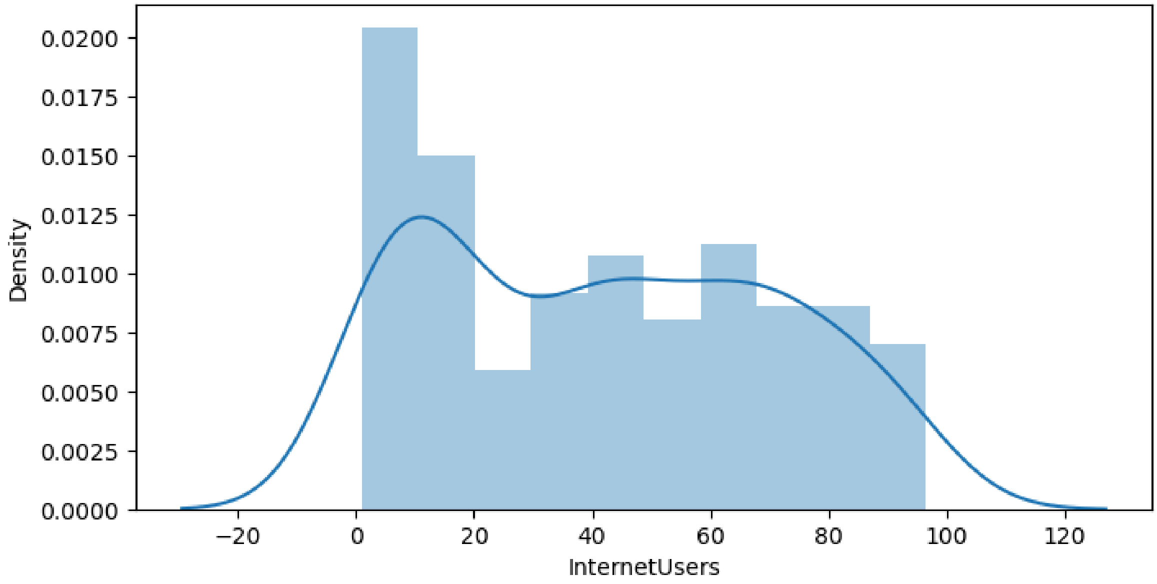


In [89]:

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

```
C:\Users\Vansh\AppData\Local\Temp\ipykernel_4660\2748777435.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

```
In [91]: plt.show()
```



```
In [93]: plt.boxplot()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[93], line 1
----> 1 plt.boxplot()

File ~\anaconda3\Lib\site-packages\matplotlib\_api\deprecation.py:300, in rename_
parameter.<locals>.wrapper(*args, **kwargs)
    295     warn_deprecated(
    296         since, message=f"The {old!r} parameter of {func.__name__}() "
    297         f"has been renamed {new!r} since Matplotlib {since}; support "
    298         f"for the old name will be dropped %(removal)s.")
    299     kwargs[new] = kwargs.pop(old)
--> 300 return func(*args, **kwargs)

TypeError: boxplot() missing 1 required positional argument: 'x'
```

```
In [97]: df
```

Out[97]:

	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
...	...	...	...	...	...	...
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 × 6 columns

In [105…

```
df.columns
```

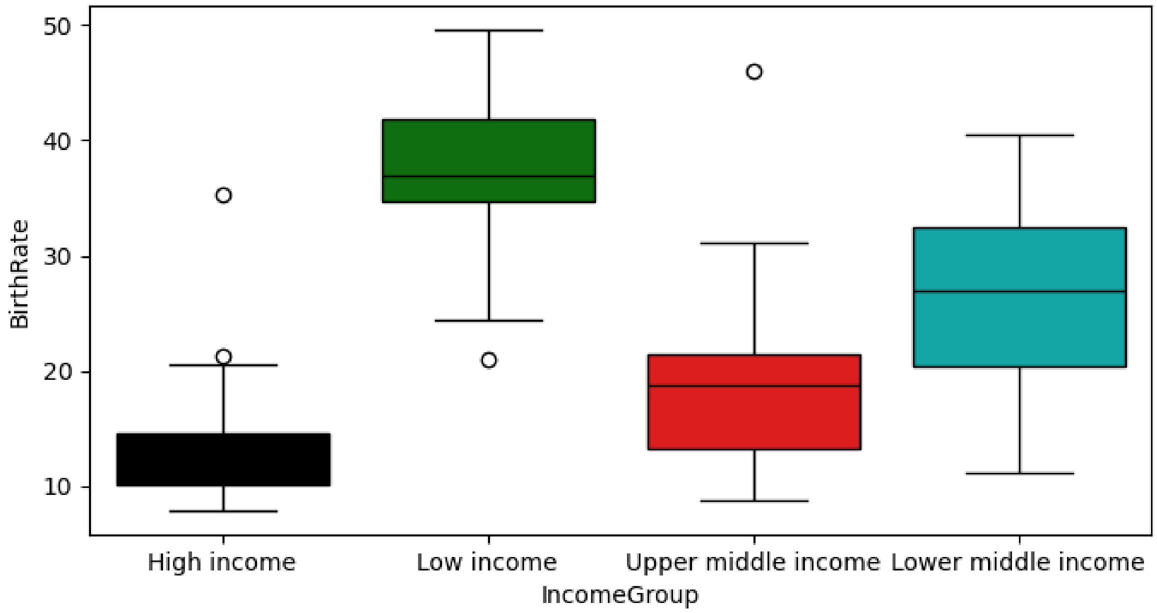
Out[105…

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

In [121…

```
#BOX PLOTS:
custom_colors = ['#FF6347', '#4CAF50', '#1E90FF', '#FFD700']
vis2 = sns.boxplot(data = df, x="IncomeGroup", y='BirthRate',palette= ['k','g','
plt.show()
```

```
C:\Users\Vansh\AppData\Local\Temp\ipykernel_4660\3405921935.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  vis2 = sns.boxplot(data = df, x="IncomeGroup", y='BirthRate',palette=
['k','g','r','c'])
```



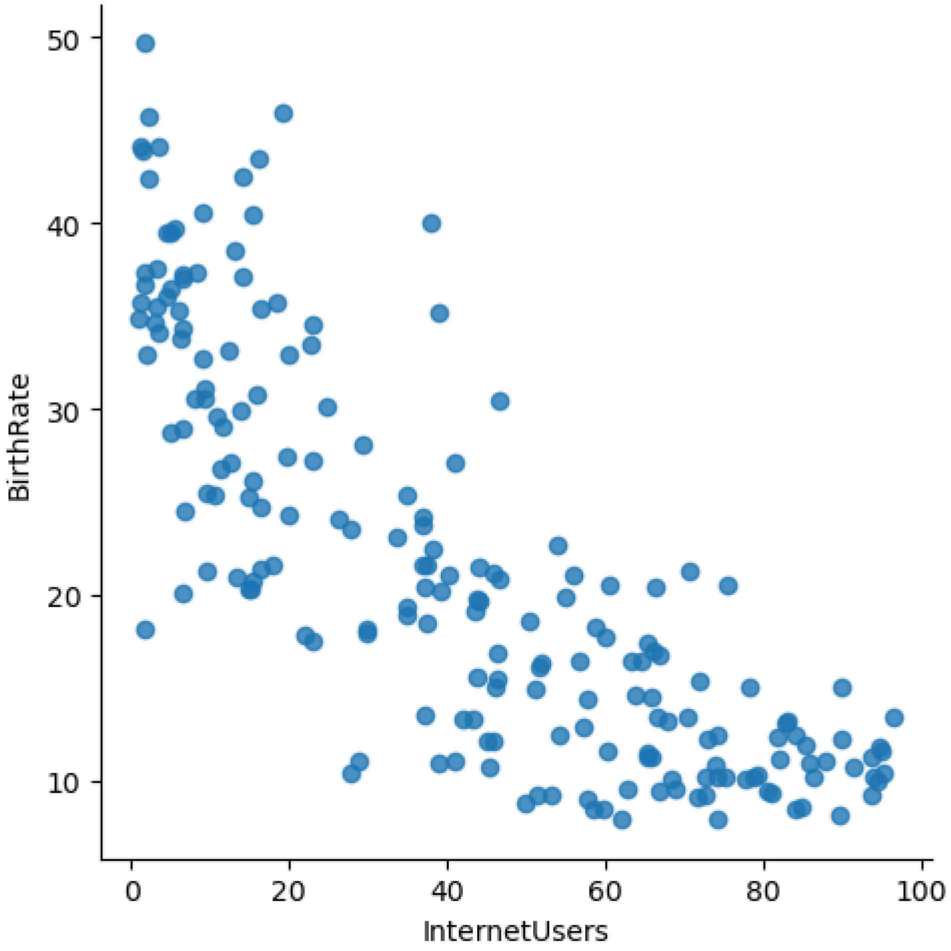
In [123…
```
plt.show()
```

In [125…
```
# Refer to seaborn gallary
```

In [127…
```
# visualizing with seaborn
```

In [129…
```
vis3 = sns.lmplot(data = df,x = 'InternetUsers', y = 'BirthRate', fit_reg = Fals
```

In [131…
```
plt.show()
```



In [ ]:

```
vis3 = sns.lmplot(data = df,x = 'InternetUsers', y = 'BirthRate', fit_reg = True
```

In [133…

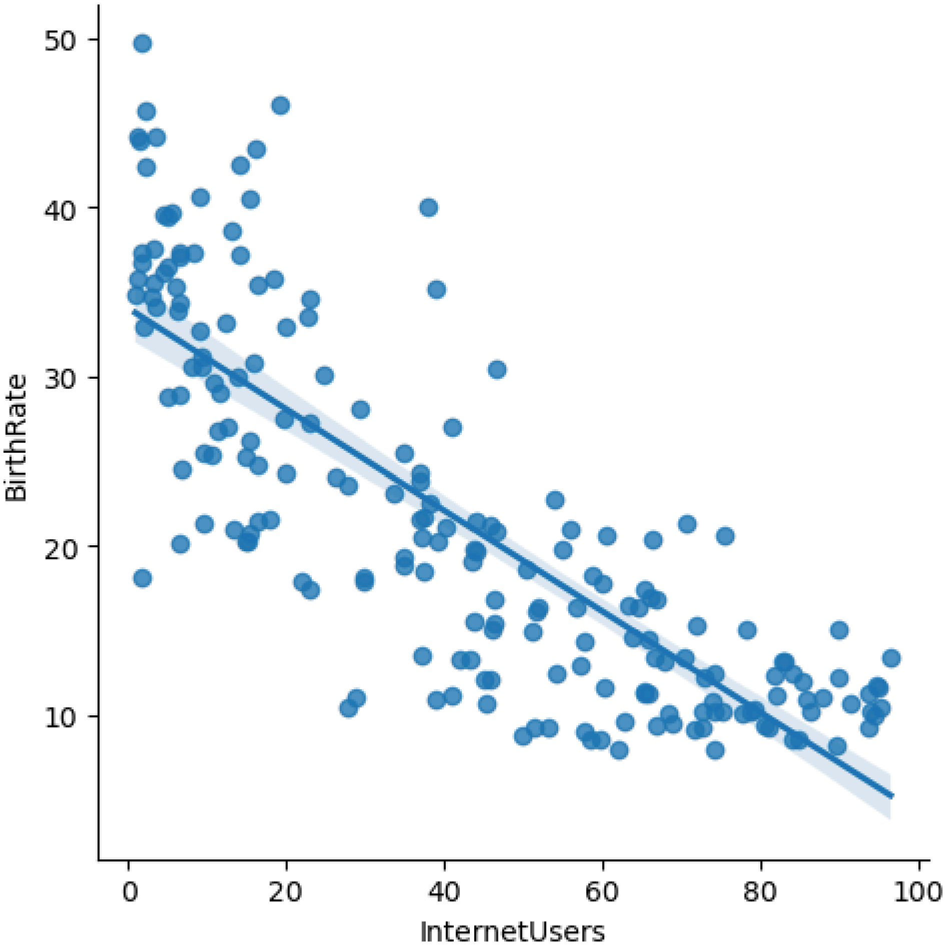
```
plt.show()
```

In [135…

```
vis4 = sns.lmplot(data = df,x = 'InternetUsers', y = 'BirthRate')
```

In [137…

```
plt.show()
```

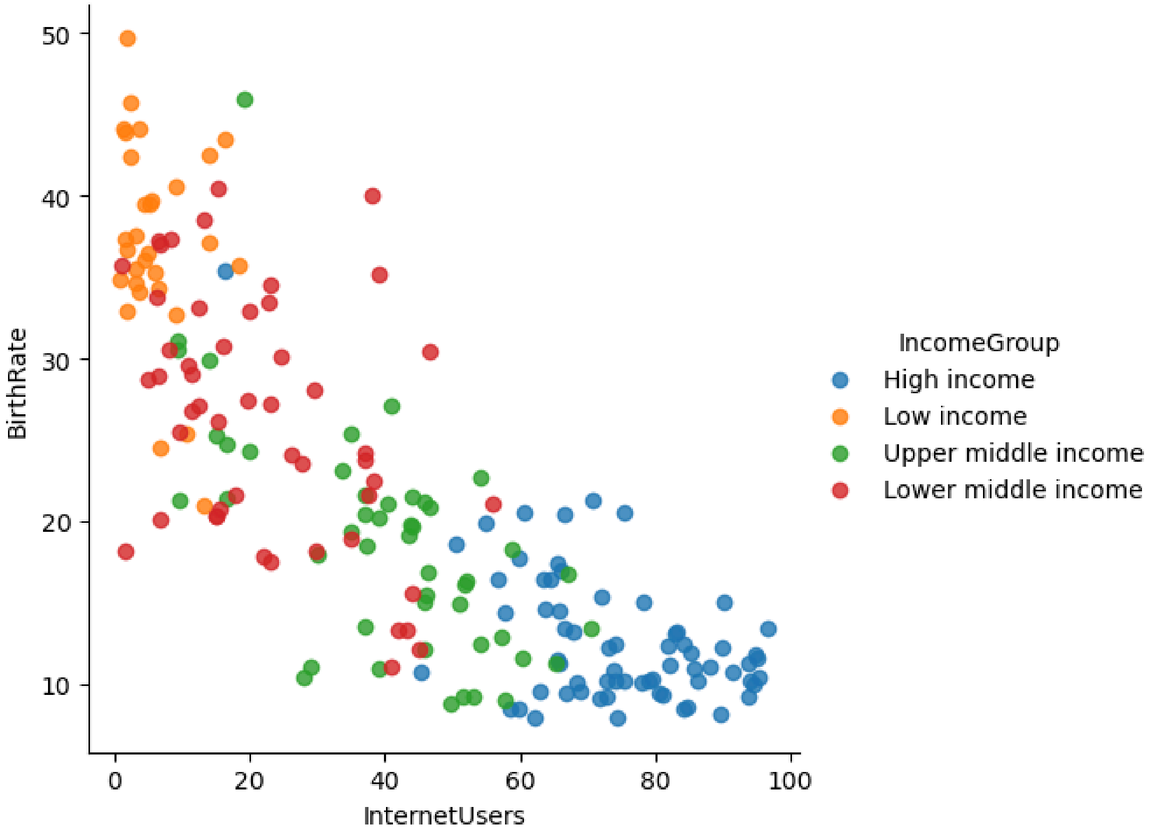


In [139…

```
vis5 = sns.lmplot(data = df,x = 'InternetUsers', y = 'BirthRate',
                  fit_reg = False,hue = 'IncomeGroup') #hue - parameter for colo
```

In [141…

```
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```