

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

```
In [2]: df = pd.read_csv(r'E:\FSDS With GEN AI_NIT\1st Feb - 28th Feb 2025\data.csv')
df
```

Out[2]:

	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 [3]: len(df)
```

```
Out[3]: 195
```

```
In [4]: df.shape
```

```
Out[4]: (195, 5)
```

```
In [5]: df.columns
```

```
Out[5]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
               'IncomeGroup'],
               dtype='object')
```

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

```
Out[6]: pandas.core.frame.DataFrame
```

```
In [9]: pd.__version__
```

```
Out[9]: '2.1.4'
```

```
In [10]: 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 [11]: `len(df.columns)`

Out[11]: 5

In [13]: `df.head() #print top 5 rows`

	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()`

	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`

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
...
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 [16]: df.head(2)

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

In [17]: df[:::-1]

Out[17]:

	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 [18]: df[:5]

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

In [19]: df[6:]

Out[19]:

	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

In [20]:

df[0:200:10]

Out[20]:

	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

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.describe().transpose()`

Out[22]:

	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 [23]:

`df.columns`

Out[23]:

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

In [26]:

`df.columns = ['a', 'b', 'c', 'd', 'e'] #rename columns`

In [25]:

`df`

Out[25]:

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

195 rows × 5 columns

In [27]: df.columns = ['a', 'b', 'c', 'd', 'e', 'f']

```

-----
ValueError                                Traceback (most recent call last)

Cell In[27], line 1
----> 1 df.columns = ['a', 'b','c','d','e','f']

File ~\.conda\envs\mytensor_env\lib\site-packages\pandas\core\generic.py:6218, in ND
Frame.__setattribute__(self, name, value)
6216     try:
6217         object.__getattribute__(self, name)
--> 6218     return object.__setattribute__(self, name, value)
6219 except AttributeError:
6220     pass

File properties.pyx:69, in pandas._libs.properties.AxisProperty.__set__()

File ~\.conda\envs\mytensor_env\lib\site-packages\pandas\core\generic.py:767, in NDF
rame._set_axis(self, axis, labels)
762 """
763 This is called from the cython code when we set the `index` attribute
764 directly, e.g. `series.index = [1, 2, 3]`.
765 """
766 labels = ensure_index(labels)
--> 767 self._mgr.set_axis(axis, labels)
768 self._clear_item_cache()

File ~\.conda\envs\mytensor_env\lib\site-packages\pandas\core\internals\managers.py:
227, in BaseBlockManager.set_axis(self, axis, new_labels)
228     def set_axis(self, axis: AxisInt, new_labels: Index) -> None:
229         # Caller is responsible for ensuring we have an Index object.
--> 230         self._validate_set_axis(axis, new_labels)
231         self.axes[axis] = new_labels

File ~\.conda\envs\mytensor_env\lib\site-packages\pandas\core\internals\base.py:85,
in DataManager._validate_set_axis(self, axis, new_labels)
82     pass
83 elif new_len != old_len:
--> 84     raise ValueError(
85         f"Length mismatch: Expected axis has {old_len} elements, new "
86         f"values have {new_len} elements"
87     )

ValueError: Length mismatch: Expected axis has 5 elements, new values have 6 element
s

```

In [28]: `df.columns`

Out[28]: `Index(['a', 'b', 'c', 'd', 'e'], dtype='object')`

In [29]: `df.columns = ['CountryName', 'CountryCode', 'BirthRate','InternetUsers','IncomeGrou`

In [30]: `df`

Out[30]:

	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 [31]:

df

Out[31]:

	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 [32]:

df.columns

```
Out[32]: Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
       dtype='object')
```

```
In [33]: df[['CountryName', 'CountryCode']]
```

```
Out[33]:
```

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

195 rows × 2 columns

```
In [34]: df.isnull()
```

Out[34]:

	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 [35]: `df.isnull().sum()`

Out[35]:

In [36]: `df.dtypes`

Out[36]:

In [37]: `df.columns`

Out[37]:

In [38]: `df_categorical = df[['CountryName', 'CountryCode', 'IncomeGroup']]`
`df_categorical`

Out[38]:

	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 [40]:

```
df_numerical = df[['BirthRate', 'InternetUsers']]
df_numerical
```

Out[40]:

	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 [41]:

```
df_categorical.describe() #we can apply describe fun to categorical data as well bu
```

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

```
In [43]: df_numerical.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

In []:

In []: