In [2]:
```
import pandas as pd
```

In [3]:
```
pd.__version__
```

Out[3]: '2.3.3'

In [5]:
```
import pandas as pd

df = pd.read_excel(r"D:\datascience\data.xlsx")
print(df)
```

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

             IncomeGroup
0            High income
1             Low income
2    Upper middle income
3    Upper middle income
4            High income
..                   ...
190  Lower middle income
191  Upper middle income
192           Low income
193  Lower middle income
194           Low income

[195 rows x 5 columns]
```

In [6]:
```
df.shape # give the dimension
```

Out[6]: (195, 5)

In [7]:
```
len(df) # give length
```

Out[7]: 195

In [8]:
```
df.isnull()
```

Out[8]:

	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 [9]:
```
df.isna()
```

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.isnull().sum() # say that any missing value in give data
```

Out[10]:
```
CountryName      0
CountryCode      0
BirthRate        0
InternetUsers    0
IncomeGroup      0
dtype: int64
```

In [11]:
```
id(df)
```

Out[11]:
```
2008378179072
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

In [13]:
```
len(df.columns)
```

Out[13]:
```
5
```

In [15]:
```
type(df)
```

Out[15]:
```
pandas.core.frame.DataFrame
```

In [14]:
```
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.head()
```

Out[17]:

	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 [18]:
```
df.tail()
```

Out[18]:

	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 [19]:
```
df.head(3)
```

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

In [20]:
```
df
```

Out[20]:

	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 [21]:
```
df.columns
```

Out[21]:
```
Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',
       'IncomeGroup'],
      dtype='object')
```

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', 'BirthRate']]
```

Out[25]:

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

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

In [31]:
```
df_cat = df[['CounrtyName', 'CountryCode', 'IncomeGroup']]
df_cat
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[31], line 1
----> 1 df_cat = df[[              ,              ,              ]]
      2 df_cat

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4119, in DataFrame.__getitem
__(self, key)
   4117     if is_iterator(key):
   4118         key = list(key)
-> 4119     indexer = self.columns._get_indexer_strict(key,          )[1]
   4121 # take() does not accept boolean indexers
   4122 if getattr(indexer, "dtype", None) == bool:

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6212, in Index._get_i
ndexer_strict(self, key, axis_name)
   6209 else:
   6210     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 6212 self._raise_if_missing(keyarr, indexer, axis_name)
   6214 keyarr = self.take(indexer)
   6215 if isinstance(key, Index):
   6216     # GH 42790 - Preserve name from an Index

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6264, in Index._raise
_if_missing(self, key, indexer, axis_name)
   6261     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
   6263 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 6264 raise KeyError(f"{not_found} not in index")

KeyError: "['CounrtyName'] not in index"
```

In [30]:
```
df_cat = df[['CountryName', 'CountryCode', 'IncomeGroup']]
df_cat
```

Out[30]:

	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 [32]:
```
df.describe()
```

Out[32]:

	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 [33]:
```
df_num.describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[33], line 1
----> 1 df_num.describe()

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

In [34]:
```
df_cat.describe()
```

Out[34]:

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

In [35]:
```
df
```

Out[35]:

	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

indexing and slicing in Pandas

In [36]:
```
df[:]
```

Out[36]:

	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 [37]:
```
df[4]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.get_loc(self, key)
   3811 try:
-> 3812     return self._engine.get_loc(casted_key)
   3813 except KeyError as err:

File pandas/_libs/index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:7096, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 4

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[37], line 1
----> 1 df[4]

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4113, in DataFrame.__getitem__(self, key)
   4111 if self.columns.nlevels > 1:
   4112     return self._getitem_multilevel(key)
-> 4113 indexer = self.columns.get_loc(key)
   4114 if is_integer(indexer):
   4115     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3819, in Index.get_loc(self, key)
   3814     if isinstance(casted_key, slice) or (
   3815         isinstance(casted_key, abc.Iterable)
   3816         and any(isinstance(x, slice) for x in casted_key)
   3817     ):
   3818         raise InvalidIndexError(key)
-> 3819     raise KeyError(key) from err
   3820 except TypeError:
   3821     # If we have a listlike key, _check_indexing_error will raise
   3822     #  InvalidIndexError. Otherwise we fall through and re-raise
   3823     #  the TypeError.
   3824     self._check_indexing_error(key)

KeyError: 4
```

In [38]:
```
df[4:10]
```

Out[38]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
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 [39]:
```
df[1:10:2]
```

Out[39]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
1	Afghanistan	AFG	35.253	5.9000	Low income
3	Albania	ALB	12.877	57.2000	Upper middle income
5	Argentina	ARG	17.716	59.9000	High income
7	Antigua and Barbuda	ATG	16.447	63.4000	High income
9	Austria	AUT	9.400	80.6188	High income

In [40]:
```
df[0:150:10]
```

Out[40]:

	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

In [41]:
```
df[::-1]
```

Out[41]:

	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 [42]:
```
df[::-2]
```

Out[42]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
194	Zimbabwe	ZWE	35.715	18.5	Low income
192	Congo, Dem. Rep.	COD	42.394	2.2	Low income
190	Yemen, Rep.	YEM	32.947	20.0	Lower middle income
188	West Bank and Gaza	PSE	30.394	46.6	Lower middle income
186	Vietnam	VNM	15.537	43.9	Lower middle income
...	...	...	...	...	...
8	Australia	AUS	13.200	83.0	High income
6	Armenia	ARM	13.308	41.9	Lower middle income
4	United Arab Emirates	ARE	11.044	88.0	High income
2	Angola	AGO	45.985	19.1	Upper middle income
0	Aruba	ABW	10.244	78.9	High income

98 rows × 5 columns

In [43]:
```
df[4,2] #Comma is not work  in pandas
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.get_loc(self, key)
   3811 try:
-> 3812     return self._engine.get_loc(casted_key)
   3813 except KeyError as err:

File pandas/_libs/index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:7096, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: (4, 2)

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[43], line 1
----> 1 df[4,2]

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4113, in DataFrame.__getitem__(self, key)
   4111 if self.columns.nlevels > 1:
   4112     return self._getitem_multilevel(key)
-> 4113 indexer = self.columns.get_loc(key)
   4114 if is_integer(indexer):
   4115     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3819, in Index.get_loc(self, key)
   3814     if isinstance(casted_key, slice) or (
   3815         isinstance(casted_key, abc.Iterable)
   3816         and any(isinstance(x, slice) for x in casted_key)
   3817     ):
   3818         raise InvalidIndexError(key)
-> 3819     raise KeyError(key) from err
   3820 except TypeError:
   3821     # If we have a listlike key, _check_indexing_error will raise
   3822     #  InvalidIndexError. Otherwise we fall through and re-raise
   3823     #  the TypeError.
   3824     self._check_indexing_error(key)

KeyError: (4, 2)
```

In [44]:
```
df[3:4]
```

Out[44]:

	CountryName	CountryCode	BirthRate	InternetUsers	IncomeGroup
3	Albania	ALB	12.877	57.2	Upper middle income

In [ ]: