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

In [3]:
```
df = pd.read_csv('IRIS.csv')
```

In [4]:
```
df
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Heading = Columns in file.to get heads we use head() method

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

```
   sepal_length  sepal_width  petal_length  petal_width      species
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa
```

to get no of rows and column we use shape .

In [7]:
```
print(df.shape)
```

```
(150, 5)
```

To get names of columns we use columns.

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

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

to get DATA TYPE of each column we use dtypes.

In [9]:

```
print(df.dtypes)
```

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

To get more information about column datatype we use info() method

In [10]:

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

To save a column in a variable

In [12]:

```
species_df=df['species']
```

to get first five observations of a column we use head()

In [13]:

```
print(species_df.head())
```

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: species, dtype: object
```

to get last five observation of a column we use tail()

In [14]:

```
print(species_df.tail())
```

```
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: species, dtype: object
```

to operate on multiple column

In [16]:

```
subset =df[['sepal_length','petal_length','species']]
print(subset.head())
```

```
   sepal_length  petal_length      species
0           5.1           1.4  Iris-setosa
1           4.9           1.4  Iris-setosa
2           4.7           1.3  Iris-setosa
3           4.6           1.5  Iris-setosa
4           5.0           1.4  Iris-setosa
```

In [17]:

```
print(subset.tail())
```

```
     sepal_length  petal_length         species
145           6.7           5.2  Iris-virginica
146           6.3           5.0  Iris-virginica
147           6.5           5.2  Iris-virginica
148           6.2           5.4  Iris-virginica
149           5.9           5.1  Iris-virginica
```

In [18]:

```
print(df.loc[0])
```

```
sepal_length            5.1
sepal_width             3.5
petal_length            1.4
petal_width             0.2
species         Iris-setosa
Name: 0, dtype: object
```

In [21]:

```
print(df.loc[100])
```

```
sepal_length               6.3
sepal_width                3.3
petal_length               6.0
petal_width                2.5
species         Iris-virginica
Name: 100, dtype: object
```

In [22]:

```
print(df.size)
```

```
750
```

In [9]:

```
import pandas as pd
df2=pd.read_csv("StudentsPerformance.csv")
```

In [10]:

```
df2
```

Out[10]:

	Name	Roll no	gender	Nationality	test preparation course	math score	reading score	writing score	Semester
0	Yash	223101	male	indian	none	72.0	72.0	74.0	7
1	Prit	223102	female	indian	completed	NaN	90.0	88.0	7
2	Meet	223103	female	indian	NaN	90.0	95.0	93.0	7
3	Drashti	223104	female	indian	none	47.0	57.0	44.0	7
4	Saloni	223105	female	indian	none	76.0	78.0	NaN	7
...	...	...	...	...	...	...	...	...	...
64	Mital	223165	female	indian	none	59.0	58.0	59.0	7
65	Nevil	223166	male	indian	none	67.0	64.0	61.0	7
66	Krishna	223167	male	indian	none	45.0	37.0	37.0	7
67	Krishna	223168	NaN	indian	none	60.0	72.0	74.0	7
68	Dhavni	223169	female	indian	none	61.0	58.0	56.0	7

69 rows × 9 columns

In [11]:

```
df2.head()
```

Out[11]:

	Name	Roll no	gender	Nationality	test preparation course	math score	reading score	writing score	Semester
0	Yash	223101	male	indian	none	72.0	72.0	74.0	7
1	Prit	223102	female	indian	completed	NaN	90.0	88.0	7
2	Meet	223103	female	indian	NaN	90.0	95.0	93.0	7
3	Drashti	223104	female	indian	none	47.0	57.0	44.0	7
4	Saloni	223105	female	indian	none	76.0	78.0	NaN	7

In [12]:

```
df2.isnull()
```

Out[12]:

	Name	Roll no	gender	Nationality	test preparation course	math score	reading score	writing score	Semester
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	True	False	False	False
2	False	False	False	False	True	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...
64	False	False	False	False	False	False	False	False	False
65	False	False	False	False	False	False	False	False	False
66	False	False	False	False	False	False	False	False	False
67	False	False	True	False	False	False	False	False	False
68	False	False	False	False	False	False	False	False	False

69 rows × 9 columns

In [13]:
```
df2.mean()
```

```
C:\Users\student\AppData\Local\Temp\ipykernel_4672\3587575296.py:1: FutureWarning: Dr
opping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depr
ecated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  df2.mean()
```

Out[13]:
```
Roll no          223135.000000
math score           61.446154
reading score        65.031250
writing score        63.734375
Semester              7.000000
dtype: float64
```

In [14]:
```
df2.min()
```

```
C:\Users\student\AppData\Local\Temp\ipykernel_4672\802159762.py:1: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depre
cated; in a future version this will raise TypeError.  Select only valid columns befo
re calling the reduction.
  df2.min()
```

Out[14]:
```
Name               Abhi
Roll no          223101
math score          0.0
reading score      17.0
writing score      10.0
Semester              7
dtype: object
```

In [15]:
```
df2.mode()
```

Out[15]:

	Name	Roll no	gender	Nationality	test preparation course	math score	reading score	writing score	Semester
0	Akshay	223101	female	indian	none	69.0	58.0	61.0	7.0
1	Bhavika	223102	NaN	NaN	NaN	NaN	72.0	65.0	NaN
2	Krishna	223103	NaN	NaN	NaN	NaN	74.0	74.0	NaN
3	NaN	223104	NaN	NaN	NaN	NaN	NaN	75.0	NaN
4	NaN	223105	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
64	NaN	223165	NaN	NaN	NaN	NaN	NaN	NaN	NaN
65	NaN	223166	NaN	NaN	NaN	NaN	NaN	NaN	NaN
66	NaN	223167	NaN	NaN	NaN	NaN	NaN	NaN	NaN
67	NaN	223168	NaN	NaN	NaN	NaN	NaN	NaN	NaN
68	NaN	223169	NaN	NaN	NaN	NaN	NaN	NaN	NaN

69 rows × 9 columns

In [22]:

```
pd.get_dummies(df2['gender'])
```

Out[22]:

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	1	0
...	...	...
64	1	0
65	0	1
66	0	1
67	0	0
68	1	0

69 rows × 2 columns

In [ ]: