Lab 10

In [12]:

```
import pandas as pd
import numpy as np
```

In [13]:

```
Mydata=pd.read_csv("iris.csv") #iris is csv file
```

In [14]:

```
print(Mydata)
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
132           6.4          2.8           5.6          2.2  virginica
133           6.3          2.8           5.1          1.5  virginica
134           6.1          2.6           5.6          1.4  virginica
135           7.7          3.0           6.1          2.3  virginica
136           6.3          3.4           5.6          2.4  virginica

[137 rows x 5 columns]
```

In [15]:

```
print(Mydata.ffill())
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
132           6.4          2.8           5.6          2.2  virginica
133           6.3          2.8           5.1          1.5  virginica
134           6.1          2.6           5.6          1.4  virginica
135           7.7          3.0           6.1          2.3  virginica
136           6.3          3.4           5.6          2.4  virginica

[137 rows x 5 columns]
```

In [16]:

```
drop=Mydata.dropna(axis=1)
drop
```

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
132	6.4	2.8	5.6	2.2	virginica
133	6.3	2.8	5.1	1.5	virginica
134	6.1	2.6	5.6	1.4	virginica
135	7.7	3.0	6.1	2.3	virginica
136	6.3	3.4	5.6	2.4	virginica

137 rows × 5 columns

In [17]:

```
drop1=Mydata.dropna()
drop1
```

Out[17]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
132	6.4	2.8	5.6	2.2	virginica
133	6.3	2.8	5.1	1.5	virginica
134	6.1	2.6	5.6	1.4	virginica
135	7.7	3.0	6.1	2.3	virginica
136	6.3	3.4	5.6	2.4	virginica

137 rows × 5 columns

In [18]:

```
print(Mydata.bfill())
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
132           6.4          2.8           5.6          2.2  virginica
133           6.3          2.8           5.1          1.5  virginica
134           6.1          2.6           5.6          1.4  virginica
135           7.7          3.0           6.1          2.3  virginica
136           6.3          3.4           5.6          2.4  virginica

[137 rows x 5 columns]
```

In [19]:

```
print(Mydata.mean())
```

```
sepal_length    5.786131
sepal_width     3.055474
petal_length    3.611679
petal_width     1.112409
dtype: float64

C:\Users\kisho\AppData\Local\Temp\ipykernel_21232\342992872.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_onl
y=None') is deprecated; in a future version this will raise TypeError.  Sele
ct only valid columns before calling the reduction.
  print(Mydata.mean())
```

In [20]:

```
print(Mydata.median())
```

```
sepal_length    5.7
sepal_width     3.0
petal_length    4.1
petal_width     1.3
dtype: float64

C:\Users\kisho\AppData\Local\Temp\ipykernel_21232\3455496640.py:1: FutureWar
ning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_on
ly=None') is deprecated; in a future version this will raise TypeError.  Sel
ect only valid columns before calling the reduction.
  print(Mydata.median())
```

In [21]:

```
print(Mydata.mode())
```

```
   sepal_length  sepal_width  petal_length  petal_width     species
0           5.0          3.0           1.5          0.2      setosa
1           NaN          NaN           NaN          NaN  versicolor
```

In [22]:

```
print(Mydata.max())
```

```
sepal_length          7.9
sepal_width           4.4
petal_length          6.9
petal_width           2.5
species         virginica
dtype: object
```

In [23]:

```
print(Mydata.min())
```

```
sepal_length       4.3
sepal_width        2.0
petal_length       1.0
petal_width        0.1
species         setosa
dtype: object
```

In [24]:

```
print(Mydata.count())
```

```
sepal_length    137
sepal_width     137
petal_length    137
petal_width     137
species         137
dtype: int64
```

Lab 9

In [25]:

```
import numpy as np
```

In [26]:

```
m = np.array([[34, 12, 25], [22, 18, 45], [41, 27, 33]])

print("Original Matrix =")
print(m)
```

```
Original Matrix =
[[34 12 25]
 [22 18 45]
 [41 27 33]]
```

In [27]:

```
# Get the second row
second_row = m[1, :]
print("\nSecond Row of Matrix=")
print(second_row)
```

```

Second Row of Matrix=
[22 18 45]
```

In [28]:

```
# Get the second column
second_column = m[:, 1]
print("\nSecond Column of Matrix=")
print(second_column)
```

```

Second Column of Matrix=
[12 18 27]
```

In [29]:

```
#Sort the matrix by values in the first column
sort = m[m[:, 0].argsort()]
print("\nMatrix sorted by first column=")
print(sort)
```

```

Matrix sorted by first column=
[[22 18 45]
 [34 12 25]
 [41 27 33]]
```

In [30]:

```
# Sort each row in ascending order
sort1 = np.sort(m, axis=1)
print("\nMatrix with each row sorted=")
print(sort1)
```

```

Matrix with each row sorted=
[[12 25 34]
 [18 22 45]
 [27 33 41]]
```

In [31]:

```
#Sort the entire matrix in ascending order
sort2 = np.sort(m, axis=None).reshape(m.shape)
print("\nMatrix fully sorted in ascending order=")
print(sort2)
```

```

Matrix fully sorted in ascending order=
[[12 18 22]
 [25 27 33]
 [34 41 45]]
```

In [ ]:

```

```