SOLUTIONS:

https://eduxir.com/study-material/ncert-solutions/class-11/information-practices/introduction-to-numpy/case-study-solutions/

#Mounting Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

#Code

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

```
#1.
iris = np.genfromtxt('/content/drive/MyDrive/STUDY2/DATA ANALYSIS LAB/LABCYCLE/IRIS_PROBLEM/iris1.csv', skip_header=1, delimiter=',', dtype=float
iris
```

```
array([[4.9, 3. , 1.4, 0.2, nan],
       [4.7, 3.2, 1.3, 0.2, nan],
       [4.6, 3.1, 1.5, 0.2, nan],
       [5. , 3.6, 1.4, 0.2, nan],
       [5.4, 3.9, 1.7, 0.4, nan],
       [4.6, 3.4, 1.4, 0.3, nan],
       [5. , 3.4, 1.5, 0.2, nan],
       [4.4, 2.9, 1.4, 0.2, nan],
       [4.9, 3.1, 1.5, 0.1, nan],
       [5.4, 3.7, 1.5, 0.2, nan],
       [4.8, 3.4, 1.6, 0.2, nan],
       [4.8, 3. , 1.4, 0.1, nan],
       [4.3, 3. , 1.1, 0.1, nan],
       [5.8, 4. , 1.2, 0.2, nan],
       [5.7, 4.4, 1.5, 0.4, nan],
       [5.4, 3.9, 1.3, 0.4, nan],
       [5.1, 3.5, 1.4, 0.3, nan],
       [5.7, 3.8, 1.7, 0.3, nan],
       [5.1, 3.8, 1.5, 0.3, nan],
       [5.4, 3.4, 1.7, 0.2, nan],
       [5.1, 3.7, 1.5, 0.4, nan],
       [4.6, 3.6, 1. , 0.2, nan],
       [5.1, 3.3, 1.7, 0.5, nan],
       [4.8, 3.4, 1.9, 0.2, nan],
       [5. , 3. , 1.6, 0.2, nan],
       [5. , 3.4, 1.6, 0.4, nan],
       [5.2, 3.5, 1.5, 0.2, nan],
       [5.2, 3.4, 1.4, 0.2, nan],
       [4.7, 3.2, 1.6, 0.2, nan],
       [4.8, 3.1, 1.6, 0.2, nan],
       [5.4, 3.4, 1.5, 0.4, nan],
       [5.2, 4.1, 1.5, 0.1, nan],
       [5.5, 4.2, 1.4, 0.2, nan],
       [4.9, 3.1, 1.5, 0.1, nan],
       [5. , 3.2, 1.2, 0.2, nan],
       [5.5, 3.5, 1.3, 0.2, nan],
       [4.9, 3.1, 1.5, 0.1, nan],
       [4.4, 3. , 1.3, 0.2, nan],
       [5.1, 3.4, 1.5, 0.2, nan],
       [5. , 3.5, 1.3, 0.3, nan],
       [4.5, 2.3, 1.3, 0.3, nan],
       [4.4, 3.2, 1.3, 0.2, nan],
       [5. , 3.5, 1.6, 0.6, nan],
       [5.1, 3.8, 1.9, 0.4, nan],
       [4.8, 3. , 1.4, 0.3, nan],
       [5.1, 3.8, 1.6, 0.2, nan],
       [4.6, 3.2, 1.4, 0.2, nan],
       [5.3, 3.7, 1.5, 0.2, nan],
       [5. , 3.3, 1.4, 0.2, nan],
       [7. , 3.2, 4.7, 1.4, nan],
       [6.4, 3.2, 4.5, 1.5, nan],
       [6.9, 3.1, 4.9, 1.5, nan],
       [5.5, 2.3, 4. , 1.3, nan],
       [6.5, 2.8, 4.6, 1.5, nan],
       [5.7, 2.8, 4.5, 1.3, nan],
       [6.3, 3.3, 4.7, 1.6, nan],
       [4.9, 2.4, 3.3, 1. , nan],
       [6.6, 2.9, 4.6, 1.3, nan],
       [5.2, 2.7, 3.9, 1.4, nan],
       [5. , 2. , 3.5, 1. , nan],
       [5.9, 3. , 4.2, 1.5, nan],
       [6. , 2.2, 4. , 1. , nan],
       [6.1, 2.9, 4.7, 1.4, nan],
       [5.6, 2.9, 3.6, 1.3, nan],
       [6.7, 3.1, 4.4, 1.4, nan],
       [5.6, 3. , 4.5, 1.5, nan],
```

```
       [5.8, 2.7, 4.1, 1. , nan],
       [6.2, 2.2, 4.5, 1.5, nan],
       [5.6, 2.5, 3.9, 1.1, nan],
       [5.9, 3.2, 4.8, 1.8, nan],
       [6.1, 2.8, 4. , 1.3, nan],
       [6.3, 2.5, 4.9, 1.5, nan],
       [6.1, 2.8, 4.7, 1.2, nan],
       [6.4, 2.9, 4.3, 1.3, nan],
       [6.6, 3. , 4.4, 1.4, nan],
       [6.8, 2.8, 4.8, 1.4, nan],
       [6.7, 3. , 5. , 1.7, nan],
       [6. , 2.9, 4.5, 1.5, nan],
       [5.7, 2.6, 3.5, 1. , nan],
       [5.5, 2.4, 3.8, 1.1, nan],
       [5.5, 2.4, 3.7, 1. , nan],
       [5.8, 2.7, 3.9, 1.2, nan],
       [6. , 2.7, 5.1, 1.6, nan],
       [5.4, 3. , 4.5, 1.5, nan],
       [6. , 3.4, 4.5, 1.6, nan],
       [6.7, 3.1, 4.7, 1.5, nan],
       [6.3, 2.3, 4.4, 1.3, nan],
       [5.6, 3. , 4.1, 1.3, nan],
       [5.5, 2.5, 4. , 1.3, nan],
       [5.5, 2.6, 4.4, 1.2, nan],
       [6.1, 3. , 4.6, 1.4, nan],
       [5.8, 2.6, 4. , 1.2, nan],
       [5. , 2.3, 3.3, 1. , nan],
       [5.6, 2.7, 4.2, 1.3, nan],
       [5.7, 3. , 4.2, 1.2, nan],
       [5.7, 2.9, 4.2, 1.3, nan],
       [6.2, 2.9, 4.3, 1.3, nan],
       [5.1, 2.5, 3. , 1.1, nan],
       [5.7, 2.8, 4.1, 1.3, nan],
       [6.3, 3.3, 6. , 2.5, nan],
       [5.8, 2.7, 5.1, 1.9, nan],
       [7.1, 3. , 5.9, 2.1, nan],
       [6.3, 2.9, 5.6, 1.8, nan],
       [6.5, 3. , 5.8, 2.2, nan],
       [7.6, 3. , 6.6, 2.1, nan],
       [4.9, 2.5, 4.5, 1.7, nan],
       [7.3, 2.9, 6.3, 1.8, nan],
       [6.7, 2.5, 5.8, 1.8, nan],
       [7.2, 3.6, 6.1, 2.5, nan],
       [6.5, 3.2, 5.1, 2. , nan],
       [6.4, 2.7, 5.3, 1.9, nan],
       [6.8, 3. , 5.5, 2.1, nan],
       [5.7, 2.5, 5. , 2. , nan],
       [5.8, 2.8, 5.1, 2.4, nan],
       [6.4, 3.2, 5.3, 2.3, nan],
       [6.5, 3. , 5.5, 1.8, nan],
       [7.7, 3.8, 6.7, 2.2, nan],
       [7.7, 2.6, 6.9, 2.3, nan],
       [6. , 2.2, 5. , 1.5, nan],
       [6.9, 3.2, 5.7, 2.3, nan],
       [5.6, 2.8, 4.9, 2. , nan],
       [7.7, 2.8, 6.7, 2. , nan],
       [6.3, 2.7, 4.9, 1.8, nan],
       [6.7, 3.3, 5.7, 2.1, nan],
       [7.2, 3.2, 6. , 1.8, nan],
       [6.2, 2.8, 4.8, 1.8, nan],
       [6.1, 3. , 4.9, 1.8, nan],
       [6.4, 2.8, 5.6, 2.1, nan],
       [7.2, 3. , 5.8, 1.6, nan],
       [7.4, 2.8, 6.1, 1.9, nan],
       [7.9, 3.8, 6.4, 2. , nan],
       [6.4, 2.8, 5.6, 2.2, nan],
       [6.3, 2.8, 5.1, 1.5, nan],
       [6.1, 2.6, 5.6, 1.4, nan],
       [7.7, 3. , 6.1, 2.3, nan],
       [6.3, 3.4, 5.6, 2.4, nan],
       [6.4, 3.1, 5.5, 1.8, nan],
       [6. , 3. , 4.8, 1.8, nan],
       [6.9, 3.1, 5.4, 2.1, nan],
       [6.7, 3.1, 5.6, 2.4, nan],
       [6.9, 3.1, 5.1, 2.3, nan],
       [5.8, 2.7, 5.1, 1.9, nan],
       [6.8, 3.2, 5.9, 2.3, nan],
       [6.7, 3.3, 5.7, 2.5, nan],
       [6.7, 3. , 5.2, 2.3, nan],
       [6.3, 2.5, 5. , 1.9, nan],
       [6.5, 3. , 5.2, 2. , nan],
       [6.2, 3.4, 5.4, 2.3, nan],
       [5.9, 3. , 5.1, 1.8, nan]])
```

]:

```
#2. Dropping column whose index=4 from array iris
iris = np.delete(iris, 4, axis=1)
iris
```

```
]: array([[4.9, 3. , 1.4, 0.2],
          [4.7, 3.2, 1.3, 0.2],
          [4.6, 3.1, 1.5, 0.2],
          [5. , 3.6, 1.4, 0.2],
          [5.4, 3.9, 1.7, 0.4],
          [4.6, 3.4, 1.4, 0.3],
          [5. , 3.4, 1.5, 0.2],
          [4.4, 2.9, 1.4, 0.2],
          [4.9, 3.1, 1.5, 0.1],
          [5.4, 3.7, 1.5, 0.2],
          [4.8, 3.4, 1.6, 0.2],
          [4.8, 3. , 1.4, 0.1],
          [4.3, 3. , 1.1, 0.1],
          [5.8, 4. , 1.2, 0.2],
          [5.7, 4.4, 1.5, 0.4],
          [5.4, 3.9, 1.3, 0.4],
          [5.1, 3.5, 1.4, 0.3],
          [5.7, 3.8, 1.7, 0.3],
          [5.1, 3.8, 1.5, 0.3],
          [5.4, 3.4, 1.7, 0.2],
          [5.1, 3.7, 1.5, 0.4],
          [4.6, 3.6, 1. , 0.2],
          [5.1, 3.3, 1.7, 0.5],
          [4.8, 3.4, 1.9, 0.2],
          [5. , 3. , 1.6, 0.2],
          [5. , 3.4, 1.6, 0.4],
          [5.2, 3.5, 1.5, 0.2],
          [5.2, 3.4, 1.4, 0.2],
          [4.7, 3.2, 1.6, 0.2],
          [4.8, 3.1, 1.6, 0.2],
          [5.4, 3.4, 1.5, 0.4],
          [5.2, 4.1, 1.5, 0.1],
          [5.5, 4.2, 1.4, 0.2],
          [4.9, 3.1, 1.5, 0.1],
          [5. , 3.2, 1.2, 0.2],
          [5.5, 3.5, 1.3, 0.2],
          [4.9, 3.1, 1.5, 0.1],
          [4.4, 3. , 1.3, 0.2],
          [5.1, 3.4, 1.5, 0.2],
          [5. , 3.5, 1.3, 0.3],
          [4.5, 2.3, 1.3, 0.3],
          [4.4, 3.2, 1.3, 0.2],
          [5. , 3.5, 1.6, 0.6],
          [5.1, 3.8, 1.9, 0.4],
          [4.8, 3. , 1.4, 0.3],
          [5.1, 3.8, 1.6, 0.2],
          [4.6, 3.2, 1.4, 0.2],
          [5.3, 3.7, 1.5, 0.2],
          [5. , 3.3, 1.4, 0.2],
          [7. , 3.2, 4.7, 1.4],
          [6.4, 3.2, 4.5, 1.5],
          [6.9, 3.1, 4.9, 1.5],
          [5.5, 2.3, 4. , 1.3],
          [6.5, 2.8, 4.6, 1.5],
          [5.7, 2.8, 4.5, 1.3],
          [6.3, 3.3, 4.7, 1.6],
          [4.9, 2.4, 3.3, 1. ],
          [6.6, 2.9, 4.6, 1.3],
          [5.2, 2.7, 3.9, 1.4],
          [5. , 2. , 3.5, 1. ],
          [5.9, 3. , 4.2, 1.5],
          [6. , 2.2, 4. , 1. ],
          [6.1, 2.9, 4.7, 1.4],
          [5.6, 2.9, 3.6, 1.3],
          [6.7, 3.1, 4.4, 1.4],
          [5.6, 3. , 4.5, 1.5],
          [5.8, 2.7, 4.1, 1. ],
          [6.2, 2.2, 4.5, 1.5],
          [5.6, 2.5, 3.9, 1.1],
          [5.9, 3.2, 4.8, 1.8],
          [6.1, 2.8, 4. , 1.3],
          [6.3, 2.5, 4.9, 1.5],
          [6.1, 2.8, 4.7, 1.2],
          [6.4, 2.9, 4.3, 1.3],
          [6.6, 3. , 4.4, 1.4],
          [6.8, 2.8, 4.8, 1.4],
          [6.7, 3. , 5. , 1.7],
          [6. , 2.9, 4.5, 1.5],
          [5.7, 2.6, 3.5, 1. ],
          [5.5, 2.4, 3.8, 1.1],
          [5.5, 2.4, 3.7, 1. ],
          [5.8, 2.7, 3.9, 1.2],
          [6. , 2.7, 5.1, 1.6],
          [5.4, 3. , 4.5, 1.5],
          [6. , 3.4, 4.5, 1.6],
          [6.7, 3.1, 4.7, 1.5],
          [6.3, 2.3, 4.4, 1.3],
          [5.6, 3. , 4.1, 1.3],
```

```
       [5.5, 2.5, 4. , 1.3],
       [5.5, 2.6, 4.4, 1.2],
       [6.1, 3. , 4.6, 1.4],
       [5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
       [6.5, 3. , 5.8, 2.2],
       [7.6, 3. , 6.6, 2.1],
       [4.9, 2.5, 4.5, 1.7],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 2.5, 5.8, 1.8],
       [7.2, 3.6, 6.1, 2.5],
       [6.5, 3.2, 5.1, 2. ],
       [6.4, 2.7, 5.3, 1.9],
       [6.8, 3. , 5.5, 2.1],
       [5.7, 2.5, 5. , 2. ],
       [5.8, 2.8, 5.1, 2.4],
       [6.4, 3.2, 5.3, 2.3],
       [6.5, 3. , 5.5, 1.8],
       [7.7, 3.8, 6.7, 2.2],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.2, 5. , 1.5],
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]])
```

]:
```
#3.print shape, dimensions and size
print(iris.shape)
print(iris.size)
print(iris.ndim)
```

]:
```
(149, 4)
596
2
```

]:
```
#4. Splitting iris into 2d arrays
iriss = np.genfromtxt('/content/drive/MyDrive/STUDY2/DATA ANALYSIS LAB/LABCYCLE/IRIS_PROBLEM/iris1.csv', skip_header=1, delimiter=',', dtype='str
species = iriss[:, 4]
iris1 = iriss[species == 'Iris-setosa',:4].astype(float)
iris2 = iriss[species == 'Iris-versicolor',:4].astype(float)
iris3 = iriss[species == 'Iris-virginica',:4].astype(float)

#5. Print the arrays
print('IRIS1:\n',iris1,'\n\n','IRIS2:\n',iris2,'\n\n','IRIS3:\n',iris3)
```

]:
```
IRIS1:
 [[4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
```

```
[5.4 3.9 1.7 0.4]
[4.6 3.4 1.4 0.3]
[5.  3.4 1.5 0.2]
[4.4 2.9 1.4 0.2]
[4.9 3.1 1.5 0.1]
[5.4 3.7 1.5 0.2]
[4.8 3.4 1.6 0.2]
[4.8 3.  1.4 0.1]
[4.3 3.  1.1 0.1]
[5.8 4.  1.2 0.2]
[5.7 4.4 1.5 0.4]
[5.4 3.9 1.3 0.4]
[5.1 3.5 1.4 0.3]
[5.7 3.8 1.7 0.3]
[5.1 3.8 1.5 0.3]
[5.4 3.4 1.7 0.2]
[5.1 3.7 1.5 0.4]
[4.6 3.6 1.  0.2]
[5.1 3.3 1.7 0.5]
[4.8 3.4 1.9 0.2]
[5.  3.  1.6 0.2]
[5.  3.4 1.6 0.4]
[5.2 3.5 1.5 0.2]
[5.2 3.4 1.4 0.2]
[4.7 3.2 1.6 0.2]
[4.8 3.1 1.6 0.2]
[5.4 3.4 1.5 0.4]
[5.2 4.1 1.5 0.1]
[5.5 4.2 1.4 0.2]
[4.9 3.1 1.5 0.1]
[5.  3.2 1.2 0.2]
[5.5 3.5 1.3 0.2]
[4.9 3.1 1.5 0.1]
[4.4 3.  1.3 0.2]
[5.1 3.4 1.5 0.2]
[5.  3.5 1.3 0.3]
[4.5 2.3 1.3 0.3]
[4.4 3.2 1.3 0.2]
[5.  3.5 1.6 0.6]
[5.1 3.8 1.9 0.4]
[4.8 3.  1.4 0.3]
[5.1 3.8 1.6 0.2]
[4.6 3.2 1.4 0.2]
[5.3 3.7 1.5 0.2]
[5.  3.3 1.4 0.2]]

IRIS2:
[[7.  3.2 4.7 1.4]
[6.4 3.2 4.5 1.5]
[6.9 3.1 4.9 1.5]
[5.5 2.3 4.  1.3]
[6.5 2.8 4.6 1.5]
[5.7 2.8 4.5 1.3]
[6.3 3.3 4.7 1.6]
[4.9 2.4 3.3 1. ]
[6.6 2.9 4.6 1.3]
[5.2 2.7 3.9 1.4]
[5.  2.  3.5 1. ]
[5.9 3.  4.2 1.5]
[6.  2.2 4.  1. ]
[6.1 2.9 4.7 1.4]
[5.6 2.9 3.6 1.3]
[6.7 3.1 4.4 1.4]
[5.6 3.  4.5 1.5]
[5.8 2.7 4.1 1. ]
[6.2 2.2 4.5 1.5]
[5.6 2.5 3.9 1.1]
[5.9 3.2 4.8 1.8]
[6.1 2.8 4.  1.3]
[6.3 2.5 4.9 1.5]
[6.1 2.8 4.7 1.2]
[6.4 2.9 4.3 1.3]
[6.6 3.  4.4 1.4]
[6.8 2.8 4.8 1.4]
[6.7 3.  5.  1.7]
[6.  2.9 4.5 1.5]
[5.7 2.6 3.5 1. ]
[5.5 2.4 3.8 1.1]
[5.5 2.4 3.7 1. ]
[5.8 2.7 3.9 1.2]
[6.  2.7 5.1 1.6]
[5.4 3.  4.5 1.5]
[6.  3.4 4.5 1.6]
[6.7 3.1 4.7 1.5]
[6.3 2.3 4.4 1.3]
[5.6 3.  4.1 1.3]
[5.5 2.5 4.  1.3]
[5.5 2.6 4.4 1.2]
```

```
[6.1 3.  4.6 1.4]
[5.8 2.6 4.  1.2]
[5.  2.3 3.3 1. ]
[5.6 2.7 4.2 1.3]
[5.7 3.  4.2 1.2]
[5.7 2.9 4.2 1.3]
[6.2 2.9 4.3 1.3]
[5.1 2.5 3.  1.1]
[5.7 2.8 4.1 1.3]]

IRIS3:
[[6.3 3.3 6.  2.5]
[5.8 2.7 5.1 1.9]
[7.1 3.  5.9 2.1]
[6.3 2.9 5.6 1.8]
[6.5 3.  5.8 2.2]
[7.6 3.  6.6 2.1]
[4.9 2.5 4.5 1.7]
[7.3 2.9 6.3 1.8]
[6.7 2.5 5.8 1.8]
[7.2 3.6 6.1 2.5]
[6.5 3.2 5.1 2. ]
[6.4 2.7 5.3 1.9]
[6.8 3.  5.5 2.1]
[5.7 2.5 5.  2. ]
[5.8 2.8 5.1 2.4]
[6.4 3.2 5.3 2.3]
[6.5 3.  5.5 1.8]
[7.7 3.8 6.7 2.2]
[7.7 2.6 6.9 2.3]
[6.  2.2 5.  1.5]
[6.9 3.2 5.7 2.3]
[5.6 2.8 4.9 2. ]
[7.7 2.8 6.7 2. ]
[6.3 2.7 4.9 1.8]
[6.7 3.3 5.7 2.1]
[7.2 3.2 6.  1.8]
[6.2 2.8 4.8 1.8]
[6.1 3.  4.9 1.8]
[6.4 2.8 5.6 2.1]
[7.2 3.  5.8 1.6]
[7.4 2.8 6.1 1.9]
[7.9 3.8 6.4 2. ]
[6.4 2.8 5.6 2.2]
[6.3 2.8 5.1 1.5]
[6.1 2.6 5.6 1.4]
[7.7 3.  6.1 2.3]
[6.3 3.4 5.6 2.4]
[6.4 3.1 5.5 1.8]
[6.  3.  4.8 1.8]
[6.9 3.1 5.4 2.1]
[6.7 3.1 5.6 2.4]
[6.9 3.1 5.1 2.3]
[5.8 2.7 5.1 1.9]
[6.8 3.2 5.9 2.3]
[6.7 3.3 5.7 2.5]
[6.7 3.  5.2 2.3]
[6.3 2.5 5.  1.9]
[6.5 3.  5.2 2. ]
[6.2 3.4 5.4 2.3]
[5.9 3.  5.1 1.8]]
```

]:
```
#6.
header=np.array(["sepal length","sepal width","petal length","petal width","species No"])
#7.
print(header)
```

]:
```
['sepal length' 'sepal width' 'petal length' 'petal width' 'species No']
```

]:
```
#8.
iris_max=iris.max(axis=0).round(2)
iris_min=iris.min(axis=0).round(2)
iris_avg=iris.mean(axis=0).round(2)
iris_std=iris.std(axis=0).round(2)
iris_var=iris.var(axis=0).round(2)

print(iris_max,iris_min,iris_avg,iris_std,iris_var)
```

]:
```
[7.9 4.4 6.9 2.5] [4.3 2.  1.  0.1] [5.85 3.05 3.77 1.21] [0.83 0.43 1.75 0.76] [0.68 0.19 3.08 0.58]
```

]:
```
#9.
iris1_max=iris1.max(axis=0).round(2)
iris1_min=iris1.min(axis=0).round(2)
iris1_avg=iris1.mean(axis=0).round(2)
iris1_std=iris1.std(axis=0).round(2)
```

```
iris1_var=iris1.var(axis=0).round(2)

iris2_max=iris2.max(axis=0).round(2)
iris2_min=iris2.min(axis=0).round(2)
iris2_avg=iris2.mean(axis=0).round(2)
iris2_std=iris2.std(axis=0).round(2)
iris2_var=iris2.var(axis=0).round(2)

iris3_max=iris3.max(axis=0).round(2)
iris3_min=iris3.min(axis=0).round(2)
iris3_avg=iris3.mean(axis=0).round(2)
iris3_std=iris3.std(axis=0).round(2)
iris3_var=iris3.var(axis=0).round(2)

print(iris1_max,iris1_min,iris1_avg,iris1_std,iris1_var)
print(iris2_max,iris2_min,iris2_avg,iris2_std,iris2_var)
print(iris3_max,iris3_min,iris3_avg,iris3_std,iris3_var)
```

]:
```
[5.8 4.4 1.9 0.6] [4.3 2.3 1.  0.1] [5.   3.42 1.47 0.24] [0.35 0.38 0.17 0.11] [0.12 0.15 0.03 0.01]
[7.  3.4 5.1 1.8] [4.9 2.  3.  1. ] [5.94 2.77 4.26 1.33] [0.51 0.31 0.47 0.2 ] [0.26 0.1  0.22 0.04]
[7.9 3.8 6.9 2.5] [4.9 2.2 4.5 1.4] [6.59 2.97 5.55 2.03] [0.63 0.32 0.55 0.27] [0.4  0.1  0.3  0.07]
```

]:
```
#10.
columns = ["Iris setosa", "Iris virginica", "Iris versicolor"]
rows = ["sepal length", "sepal width", "petal length", "petal width"]

df = pd.DataFrame(columns=columns, index=rows)

df.loc["sepal length","Iris setosa"]=iris1_min[0] > iris_min[0]
df.loc["sepal length","Iris virginica"]=iris2_min[0] > iris_min[0]
df.loc["sepal length","Iris versicolor"]=iris3_min[0] > iris_min[0]

df.loc["sepal width","Iris setosa"]=iris1_min[1] > iris_min[1]
df.loc["sepal width","Iris virginica"]=iris2_min[1] > iris_min[1]
df.loc["sepal width","Iris versicolor"]=iris3_min[1] > iris_min[1]

df.loc["petal length","Iris setosa"]=iris1_min[2] > iris_min[2]
df.loc["petal length","Iris virginica"]=iris2_min[2] > iris_min[2]
df.loc["petal length","Iris versicolor"]=iris3_min[2] > iris_min[2]

df.loc["petal width","Iris setosa"]=iris1_min[3] > iris_min[3]
df.loc["petal width","Iris virginica"]=iris2_min[3] > iris_min[3]
df.loc["petal width","Iris versicolor"]=iris3_min[3] > iris_min[3]


df
'''
#or

# Loop through rows and columns to fill the DataFrame
for row in rows:
    iris_min_value = globals()["iris_min"][rows.index(row)]  # Get the corresponding iris_min value

    for column in columns:
        iris_min_column = globals()["iris{}_min".format(columns.index(column) + 1)][rows.index(row)]
        condition = iris_min_column > iris_min_value
        df.loc[row, column] = condition

df
'''
```

]:

	Iris setosa	Iris virginica	Iris versicolor
sepal length	False	True	True
sepal width	True	False	True
petal length	False	True	True
petal width	False	True	True

]:
```
#11, 12, 13.

iris_avg=np.array([(iris1_avg[1]>iris2_avg[1]),
                   (iris1_avg[2]>iris2_avg[2]),
                   (iris1_avg[3]>iris2_avg[3])])

print(iris_avg)
```

]:
```
[ True False False]
```

]:
```
#14.
np.savetxt('/content/drive/MyDrive/STUDY2/DATA ANALYSIS LAB/LABCYCLE/IRIS_PROBLEM/IrisMeanValues.txt',iris_avg,delimiter=',',fmt='%.2f')
```

]:
```
#15.
# Find the maximum number of columns among the arrays
max_columns = max(iris_max.shape[0], iris_avg.shape[0], iris_min.shape[0])
```

```
# Pad the shorter arrays with NaN values to match the maximum number of columns
iris_max_padded = np.pad(iris_max, (0, max_columns - iris_max.shape[0]), constant_values=np.nan)
iris_avg_padded = np.pad(iris_avg, (0, max_columns - iris_avg.shape[0]), constant_values=np.nan)
iris_min_padded = np.pad(iris_min, (0, max_columns - iris_min.shape[0]), constant_values=np.nan)

# Stack the padded arrays vertically to create a 2D array
iris_stat = np.vstack((iris_max_padded, iris_avg_padded, iris_min_padded))

fmt = '%.2f'

np.savetxt('/content/drive/MyDrive/STUDY2/DATA ANALYSIS LAB/LABCYCLE/IRIS_PROBLEM/IrisStat.txt',
           iris_stat, delimiter=',', fmt=fmt)
```