

## Q1: Numpy array

1. Write a Python program to create a 5x5 array with random values and find the minimum and maximum values.
2. Write a Python program to create a random 10x4 array and extract the first five rows of the array and store them into a variable.
3. Write a Python program to create a random vector of size 10 and sort it.
4. Write a Python program to find the most frequent value in an one-dimensional array

In [ ]:

```
# Q1.1
import numpy as np
arr = np.random.rand(5, 5)

min_val = np.min(arr)
max_val = np.max(arr)
print("Array:")
print(arr)
print("Minimum value:", min_val)
print("Maximum value:", max_val)

#Q1.2
arr = np.random.rand(10, 4)
first_five_rows = arr[:5, :]
print("Array:")
print(arr)
print("First five rows:")
print(first_five_rows)

#Q1.3
vec = np.random.rand(10)
# Sort the vector
sorted_vec = np.sort(vec)
print("Vector:")
print(vec)
print("Sorted vector:")
print(sorted_vec)

#Q1.4
arr = np.array([1, 2, 3, 4, 1, 2, 3, 1, 2, 1])

# Find the most frequent value
most_freq_val = np.bincount(arr).argmax()

print("Array:")
print(arr)
print("Most frequent value:", most_freq_val)
```

Array:

```
[[0.65343859 0.40131901 0.80661201 0.90827084 0.97253937]
 [0.44195564 0.61107106 0.47778849 0.7143094 0.26731692]
 [0.89224649 0.45709448 0.13227858 0.51015184 0.09533194]
 [0.95958318 0.15137788 0.06036099 0.99900175 0.5688955 ]
 [0.15263539 0.096094 0.08682336 0.99681491 0.08379173]]
```

Minimum value: 0.060360990638355316

Maximum value: 0.9990017454256855

Array:

```
[[0.61534186 0.7844447 0.2236946 0.30965475]
 [0.26641308 0.1646026 0.93738371 0.01441015]
 [0.17661365 0.00721641 0.37762121 0.95725058]
 [0.74279005 0.71745329 0.30281841 0.25159635]
 [0.79785256 0.79049072 0.27525058 0.63279884]
 [0.80495615 0.41445176 0.44004392 0.54583671]
 [0.24307926 0.48612474 0.6985225 0.2237292 ]
 [0.73929606 0.59157316 0.78229287 0.94939461]
 [0.50962642 0.80929789 0.83278139 0.40127202]
 [0.98691956 0.05726697 0.81321345 0.20202481]]
```

First five rows:

```
[[0.61534186 0.7844447 0.2236946 0.30965475]
 [0.26641308 0.1646026 0.93738371 0.01441015]
 [0.17661365 0.00721641 0.37762121 0.95725058]
 [0.74279005 0.71745329 0.30281841 0.25159635]
 [0.79785256 0.79049072 0.27525058 0.63279884]]
```

Vector:

```
[0.86891569 0.95587306 0.65261503 0.42963711 0.73198988 0.77590815
 0.55669227 0.54507453 0.8306748 0.5986303 ]
```

Sorted vector:

```
[0.42963711 0.54507453 0.55669227 0.5986303 0.65261503 0.73198988
 0.77590815 0.8306748 0.86891569 0.95587306]
```

Array:

```
[1 2 3 4 1 2 3 1 2 1]
```

Most frequent value: 1

Q2. Linear Algebra (40pts) Bob invested 10000 dollars on two mutual funds one year ago. After one year, the overall yield rates for these two mutual funds are +2% and +4%. His yearend account showed \$10,250. How much did he initially put onto two mutual funds?

Let's represent the initial amount invested in mutual fund 1 by  $x$ , and the initial amount invested in mutual fund 2 by  $y$ . So we have the equation:  $x + y = 10000$  After one year, the overall yield rates for these two mutual funds are +2% and +4%. His yearend account showed \$10,250.

The amount he earned on the first mutual fund is  $0.02x$  and the amount he earned on the second mutual fund is  $0.04y$ . So, we have the equation:  $x + y + 0.02x + 0.04y = 10250$ ;  $1.02x + 1.04y = 10250$  We have this:  $x + y = 10000$   $1.02x + 1.04y = 10250$  let's multiplying the first equation by 1.02, we get:  $1.02x + 1.02y = 10200$ , then we get we get:  $0.02y = 50$ , thus we get  $y = 2500$  for the  $x$  we get:  $x + 2500 = 10000$ , so,  $x = 7500$ .

Therefore, Bob put 7500 on one mutual fund and \$2500 on the other mutual fund.

Q3. K-means Clustering Load the data from attached file "Assignment\_table.csv". • Create a histogram for the first and second column respectively (20 points) • Conduct a clustering algorithm on these two columns. Try out 3,4,5 clusters respectively. What do you think is the right number of clusters for this dataset? (40 points)

In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans

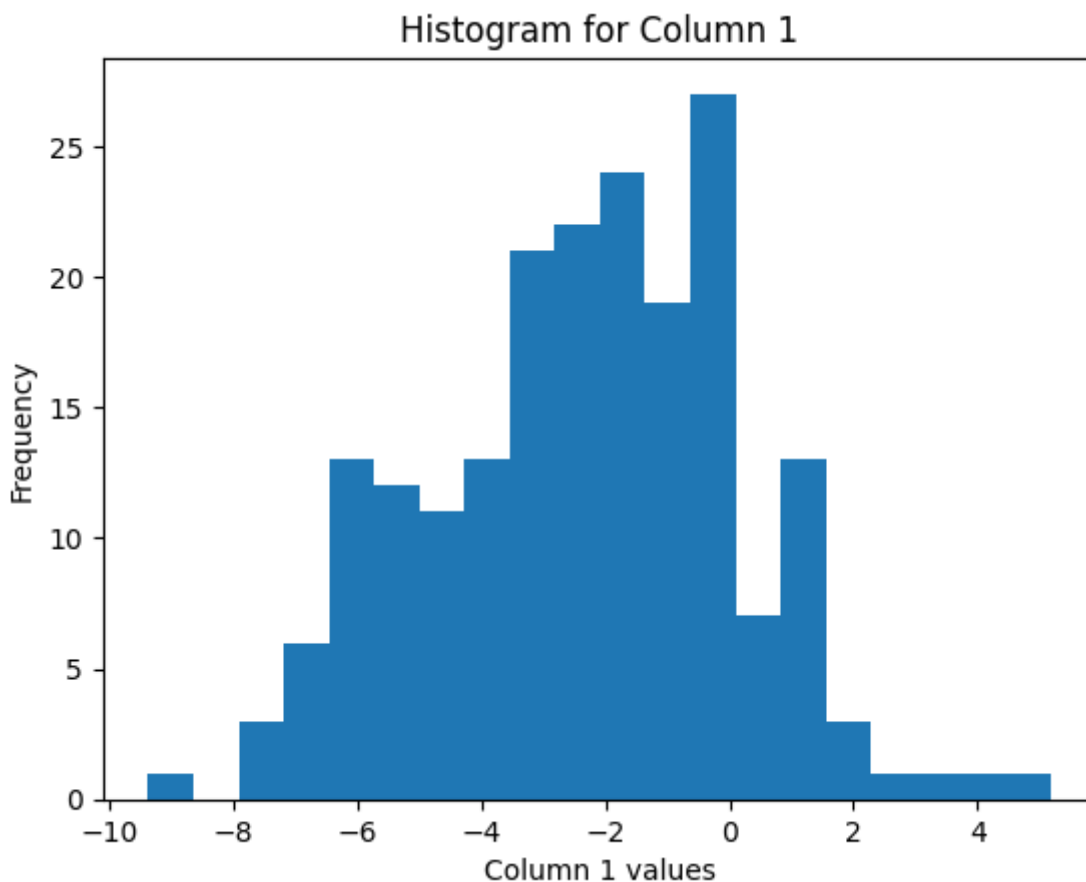
data = pd.read_csv("Assignment_table.csv")

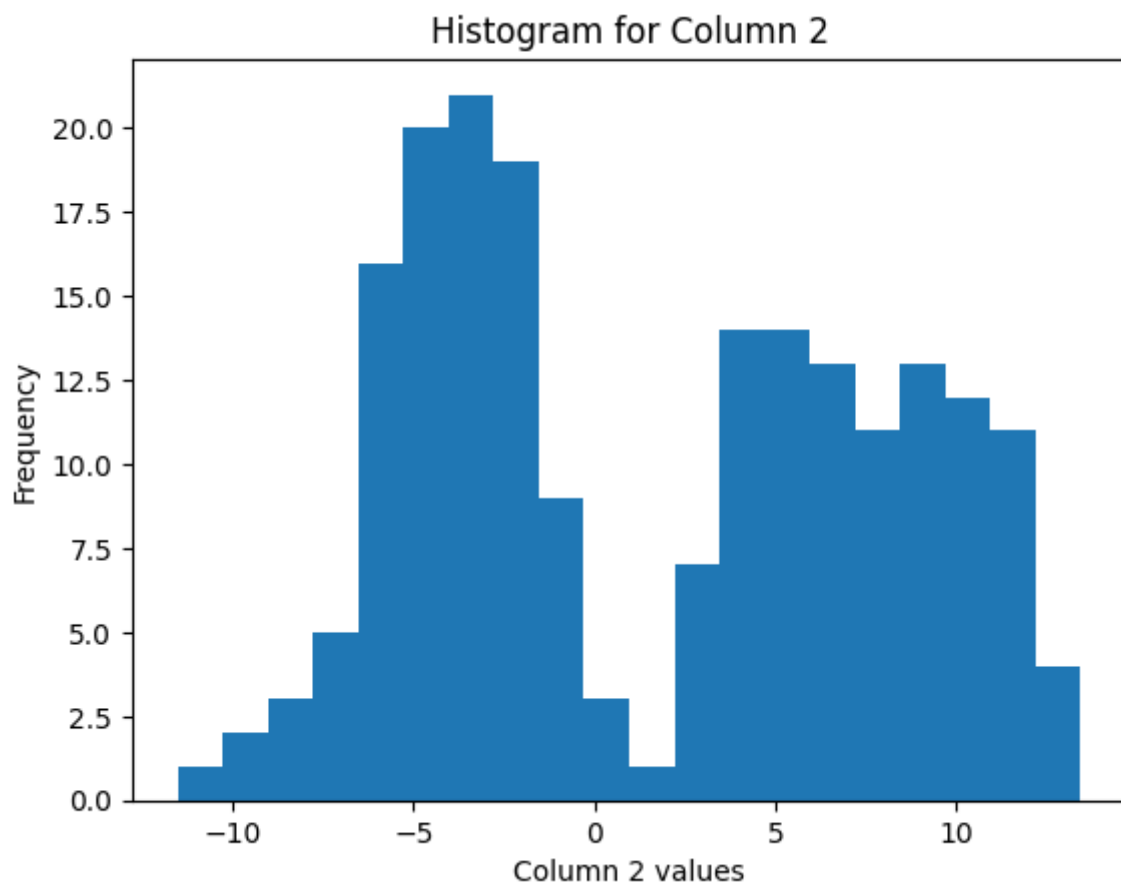
# Create histograms for first and second columns
```

```
plt.hist(data.iloc[:, 0], bins=20)
plt.title("Histogram for Column 1")
plt.xlabel("Column 1 values")
plt.ylabel("Frequency")
plt.show()
plt.hist(data.iloc[:, 1], bins=20)
plt.title("Histogram for Column 2")
plt.xlabel("Column 2 values")
plt.ylabel("Frequency")
plt.show()

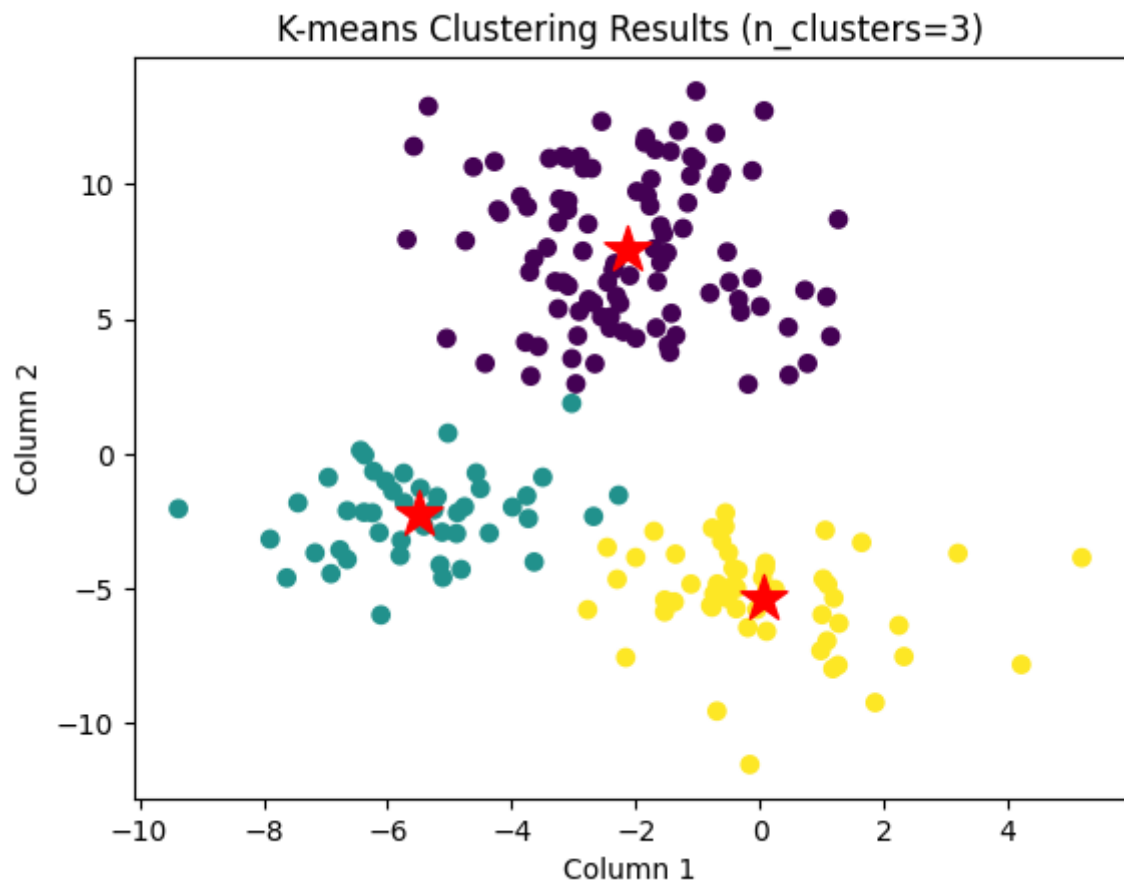
# do clustering
X = data.iloc[:, [0, 1]].values

for n_clusters in range(3, 6):
    kmeans = KMeans(n_clusters=n_clusters, random_state=0)
    kmeans.fit(X)
    labels = kmeans.labels_
    centroids = kmeans.cluster_centers_
    # show the results
    plt.scatter(X[:, 0], X[:, 1], c=labels)
    plt.scatter(centroids[:, 0], centroids[:, 1], marker='*', s=300, c='r')
    plt.title(f'K-means Clustering Results (n_clusters={n_clusters})')
    plt.xlabel('Column 1')
    plt.ylabel('Column 2')
    plt.show()
```

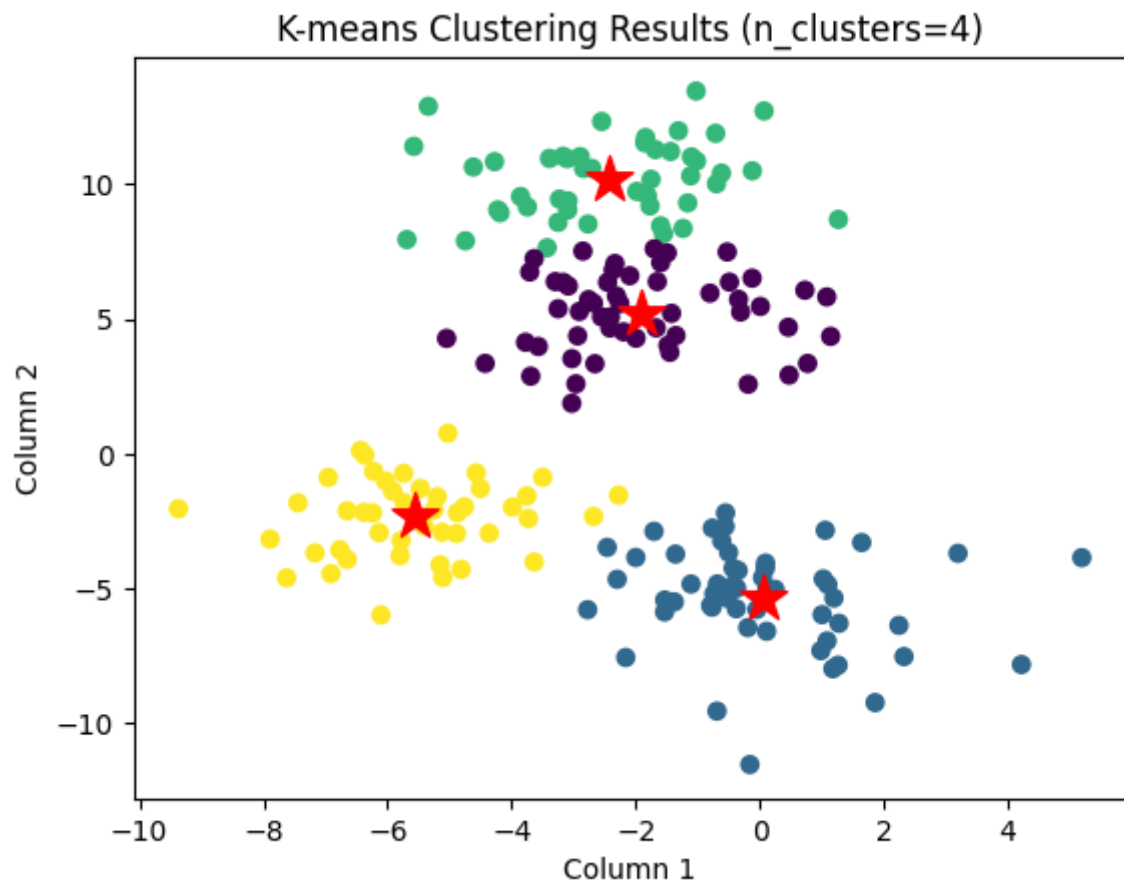




```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/  
sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil  
l change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr  
ess the warning  
    warnings.warn(  
        warnings.warn(  
            "The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning.",  
            FutureWarning,  
            stacklevel=2,  
        )  
    )
```



```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/  
sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil  
l change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr  
ess the warning  
    warnings.warn(
```



```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/  
sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` wil  
l change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppr  
ess the warning  
    warnings.warn(
```

