1.Given dataset contains 200 records and five columns, two of which describe the customer's annual income and spending score. The latter is a value from 0 to 100. The higher the number, the more this customer has spent with the company in the past:

Functions to familiarize:

- The purpose of Kmeans.fit() is to train the model with data.
- The purpose of Kmeans.predict() is to apply a trained model to data

==========

Q. Using k means clustering create 6 clusters of customers based on their spending pattern.

- **Visualize the same in a scatter plot with each cluster in a different color scheme.**

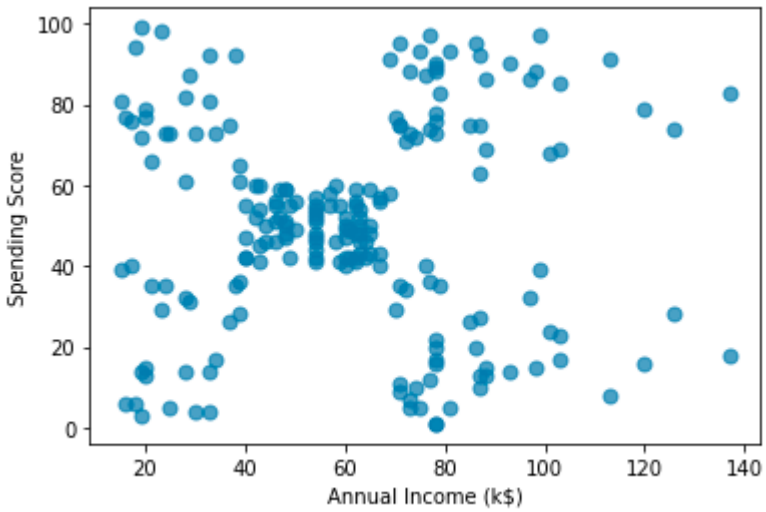
Code

```
from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

import pandas as pd

customers = pd.read_csv('customer_data.csv')

customers.head()
```

:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
points = customers.iloc[:, 3:5].values
x = points[:, 0]
y = points[:, 1]
plt.scatter(x, y, s=50, alpha=0.7)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score')
```



- **Display the cluster labels of each point.(print cluster indexes)**

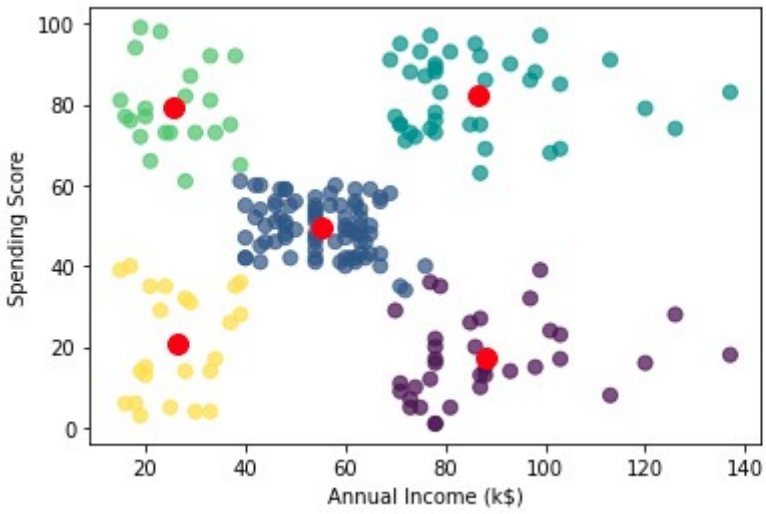
```
print (predicted_cluster_indexes)
```

```
print (predicted_cluster_indexes)
```

```
[4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
 3 4 3 4 3 4 1 4 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 2 0 2 1 2 0 2 0 2 1 2 0 2 0 2 0 2 1 2 0 2 0 2
 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0
 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2]
```

- **Display the cluster centers.**

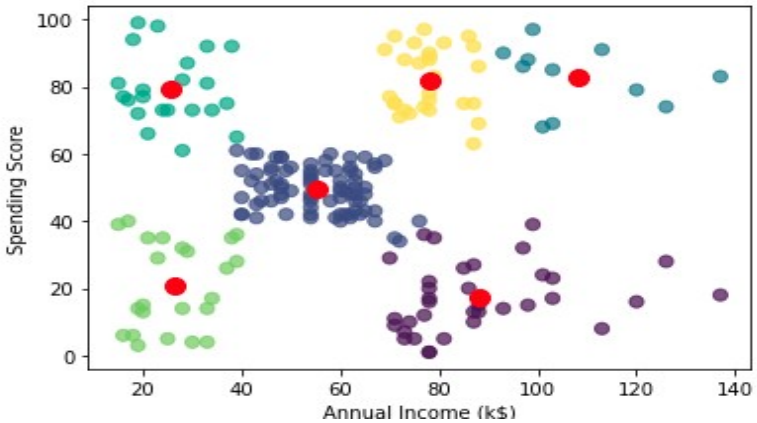
```
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(points)
predicted_cluster_indexes = kmeans.predict(points)
plt.scatter(x, y, c=predicted_cluster_indexes, s=50, alpha=0.7, cmap='viridis')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=100)
```



- **Use different values of K and visualize the same using scatter plot**

```
kmeans = KMeans(n_clusters=6, random_state=0)
kmeans.fit(points)
predicted_cluster_indexes = kmeans.predict(points)
```

```
plt.scatter(x, y, c=predicted_cluster_indexes, s=50, alpha=0.7,
cmap='viridis')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score')

centers = kmeans.cluster_centers_

plt.scatter(centers[:, 0], centers[:, 1], c='red', s=100)
```



References:

https://www.atmosera.com/blog/unsupervised-learning-with-k-means-clustering-part-ii/

https://nickmccullum.com/python-machine-learning/k-means-clustering-python/