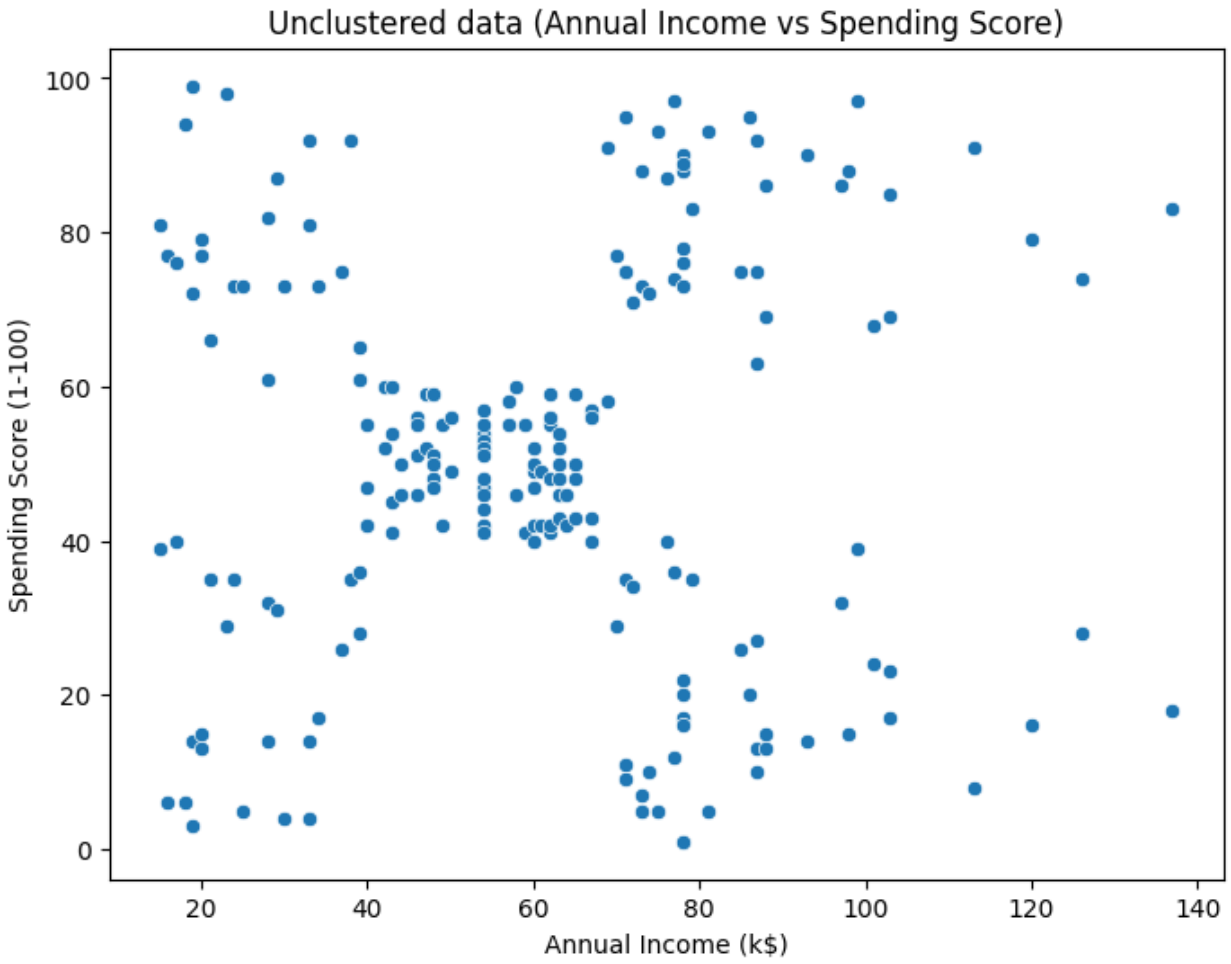
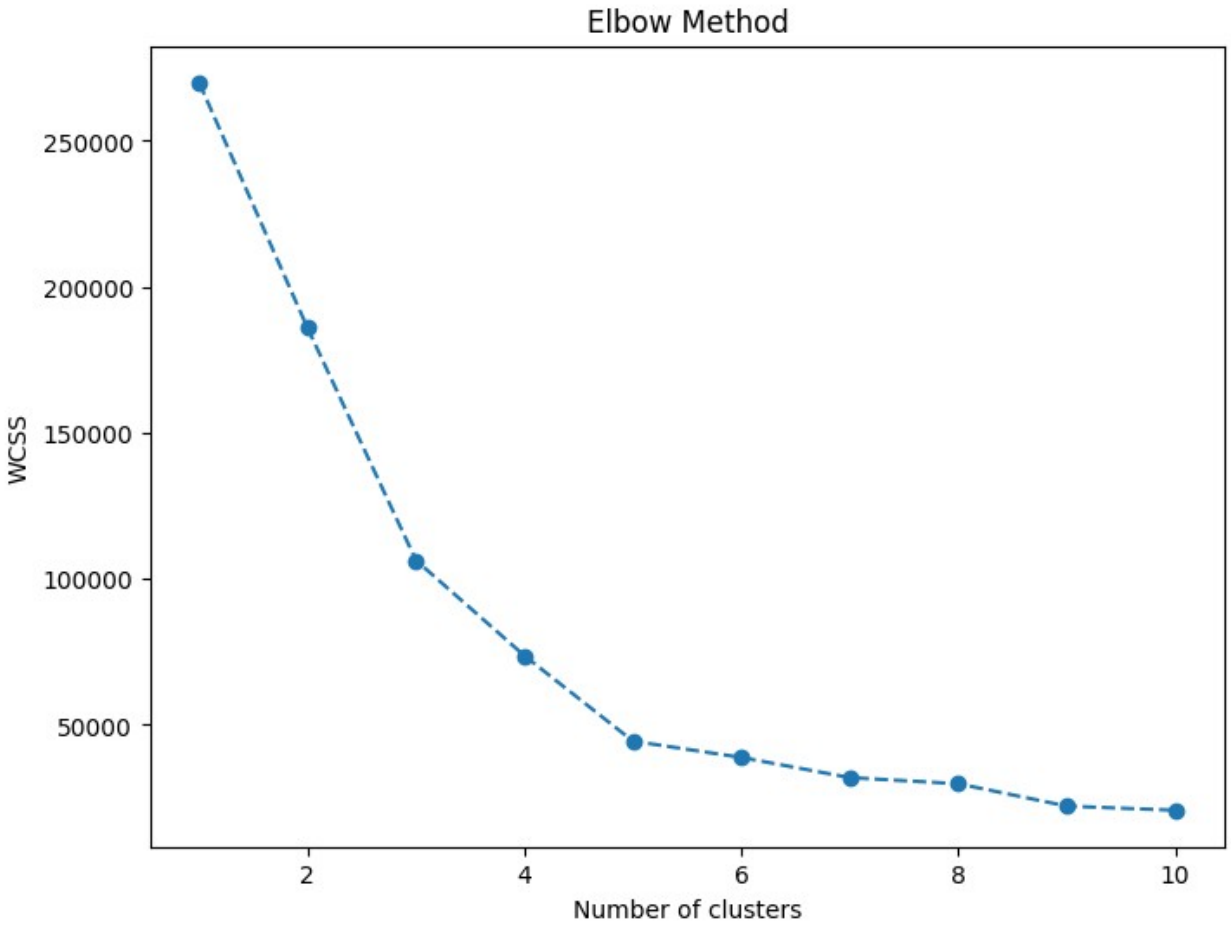
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('/content/Mall_Customers.csv')

data.describe()
data.columns
data.head(5)
data.isnull().sum()
```

```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
plt.figure(figsize=(8, 6))
plt.title('Unclustered data (Annual Income vs Spending Score)')
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)',
data=data)
plt.show()
```




```
X = data[['Annual Income (k$)', 'Spending Score (1-100)']]

from sklearn.cluster import KMeans

# Elbow Method
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

# Ploting the Elbow graph
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```




```
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(X)

data['Cluster'] = kmeans.labels_

plt.figure(figsize=(10, 6))
plt.title('Clustered data (Annual Income vs Spending Score)')
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)',
hue='Cluster', palette='rainbow', data=data)

centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 0], centroids[:, 1], s=300, c='black',
marker='.', label='Centroids')
plt.legend()
plt.show()
```

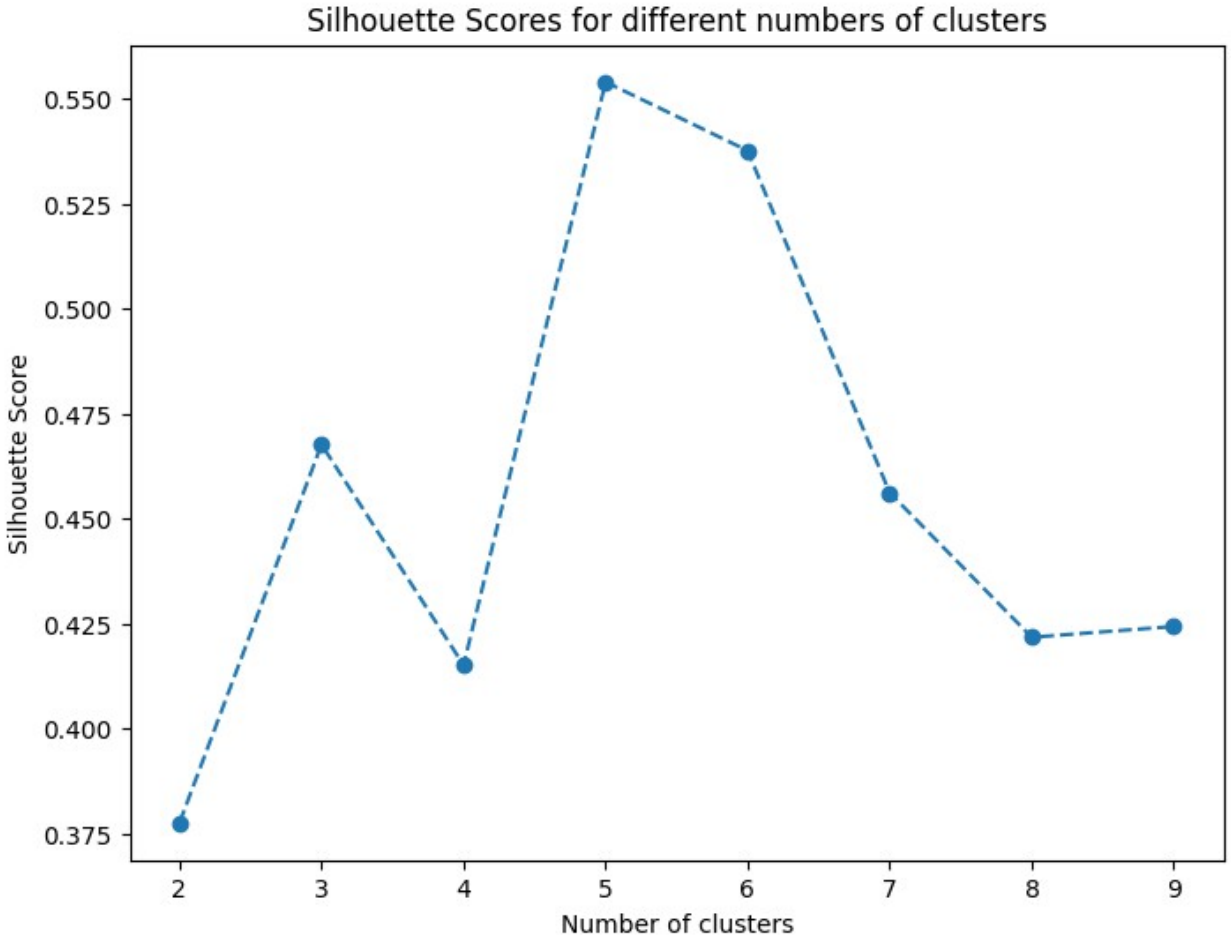

```
from sklearn.metrics import silhouette_score

silhouette= silhouette_score(X, kmeans.labels_)
print(f'Silhouette Score: {silhouette}')
```

```
Silhouette Score: 0.553931997444648
```

```
#.....................................................................
.........................

cluster_range = range(2, 10)
silhouette_scores = []

for n_clusters in cluster_range:
    kmeans = KMeans(n_clusters=n_clusters, random_state=30)
    kmeans.fit(X)


    labels = kmeans.labels_


    silhouette_avg = silhouette_score(X, labels)
    silhouette_scores.append(silhouette_avg)
    print(f"Number of clusters: {n_clusters}, Silhouette Score:
{silhouette_avg}")
```

```
plt.figure(figsize=(8, 6))
plt.plot(cluster_range, silhouette_scores, marker='o', linestyle='--')
plt.title('Silhouette Scores for different numbers of clusters')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Score')
plt.show()
```

```
Number of clusters: 2, Silhouette Score: 0.3774913479961559
Number of clusters: 3, Silhouette Score: 0.46761358158775435
Number of clusters: 4, Silhouette Score: 0.41508307009338524
Number of clusters: 5, Silhouette Score: 0.553931997444648
Number of clusters: 6, Silhouette Score: 0.5375812731362782
Number of clusters: 7, Silhouette Score: 0.45611379440879946
Number of clusters: 8, Silhouette Score: 0.42171151217040276
Number of clusters: 9, Silhouette Score: 0.4242794142604815
```




```
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage
```

```
plt.figure(figsize=(10, 8))
linked = linkage(X, method='ward')

dendrogram(linked,
           orientation='top',
           labels=data['CustomerID'].astype(str).values,
           distance_sort='descending',
           show_leaf_counts=True)
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()
```