```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler


A=pd.read_csv("Mall_Customers.csv")


A.head()
```

	CustomerID	Genre	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

Next steps: Generate code with A | View recommended plots | New interactive sheet

```
print("Missing values:")
A.isnull().sum()
```

```
Missing values:
```

	0
CustomerID	0
Genre	0
Age	0
Annual Income (k$)	0
Spending Score (1-100)	0

dtype: int64

```
A.drop("CustomerID",axis=1,inplace=True)


from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()


A.Genre = le.fit_transform(A.Genre)


A.head()
```

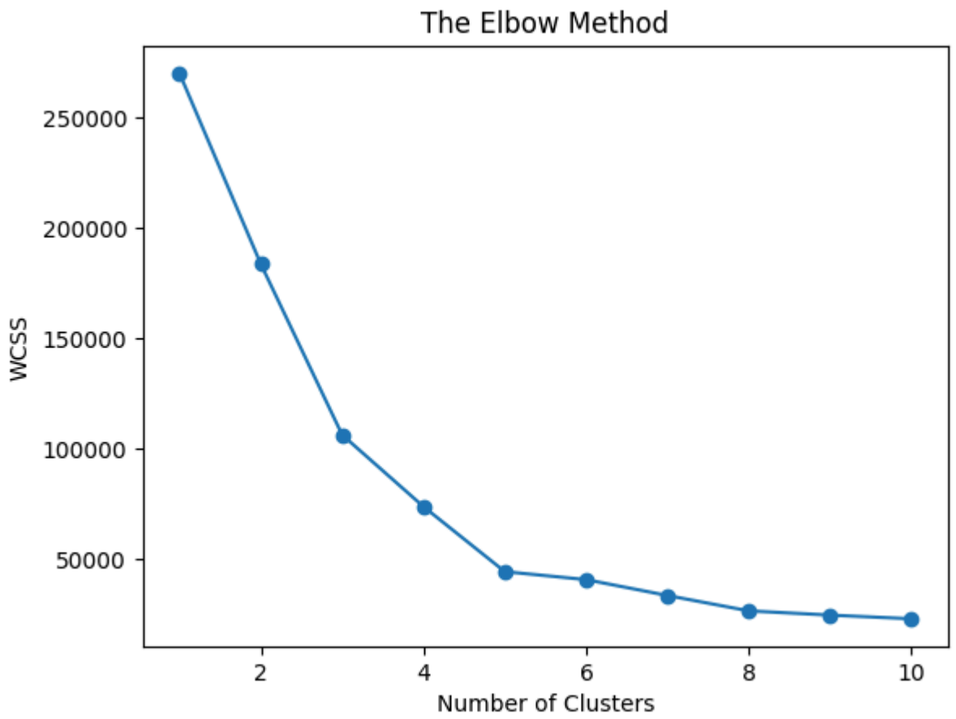
	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	19	15	39
1	1	21	15	81
2	0	20	16	6
3	0	23	16	77
4	0	31	17	40

Next steps: Generate code with A | View recommended plots | New interactive sheet

```
data=A.copy()
x = data.iloc[:,[2,3]]
from sklearn.cluster import KMeans
wcss=[]
for i in range (1,11):
   kmeans =KMeans(n_clusters=i,init="k-means++",random_state=42)
   kmeans.fit(x)
   wcss.append(kmeans.inertia_)
   print('k:',i,"-> wcss:",kmeans.inertia_)
```

```
k: 1 -> wcss: 269981.28000000014
k: 2 -> wcss: 183653.3289473683
k: 3 -> wcss: 106348.37306211119
k: 4 -> wcss: 73880.64496247198
k: 5 -> wcss: 44448.45544793369
k: 6 -> wcss: 40825.16946386947
k: 7 -> wcss: 33642.57922077922
k: 8 -> wcss: 26686.837785187785
k: 9 -> wcss: 24766.471609793436
k: 10 -> wcss: 23103.122085983905
```

```
plt.plot(range(1,11),wcss,marker='o')
plt.title('The Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```




```
kmeans=KMeans(n_clusters=5)
kmeans.fit(data)
y=kmeans.predict(data)
data["label"]=y
data.head()
```

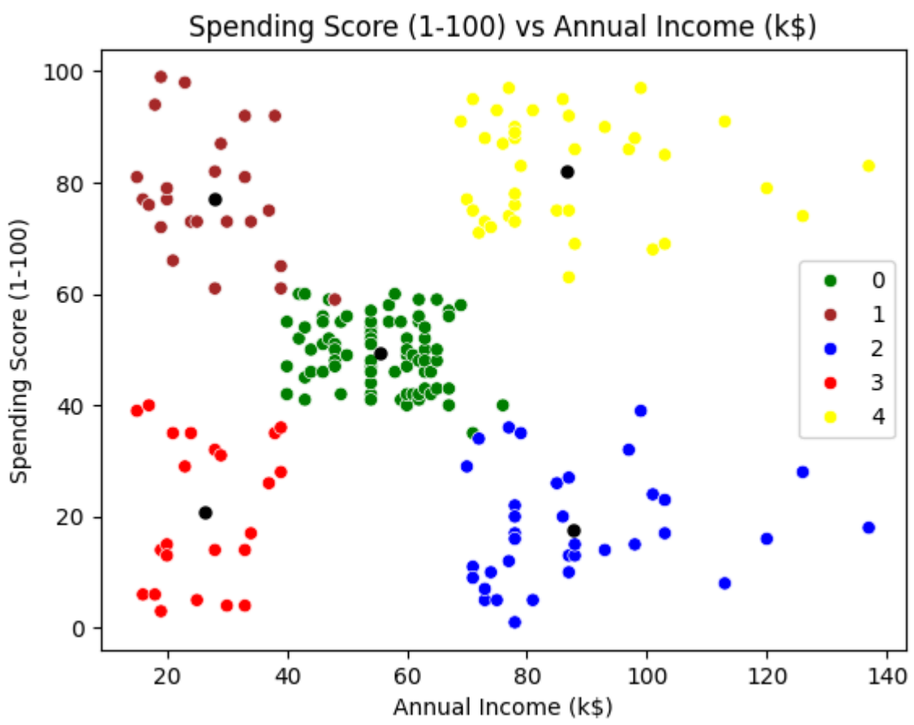
	Genre	Age	Annual Income (k$)	Spending Score (1-100)	label
0	1	19	15	39	3
1	1	21	15	81	1
2	0	20	16	6	3
3	0	23	16	77	1
4	0	31	17	40	3

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
A.corr()
```

	Genre	Age	Annual Income (k$)	Spending Score (1-100)
Genre	1.000000	0.060867	0.056410	-0.058109
Age	0.060867	1.000000	-0.012398	-0.327227
Annual Income (k$)	0.056410	-0.012398	1.000000	0.009903
Spending Score (1-100)	-0.058109	-0.327227	0.009903	1.000000

```
sb.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',hue="label",palette =['green','brown','blue','red','yellow'],data = data)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
centroids  = kmeans.cluster_centers_
plt.scatter(centroids[:,2],centroids[:,3],c='black',s=100,marker='.')
plt.legend()
plt.show()
```



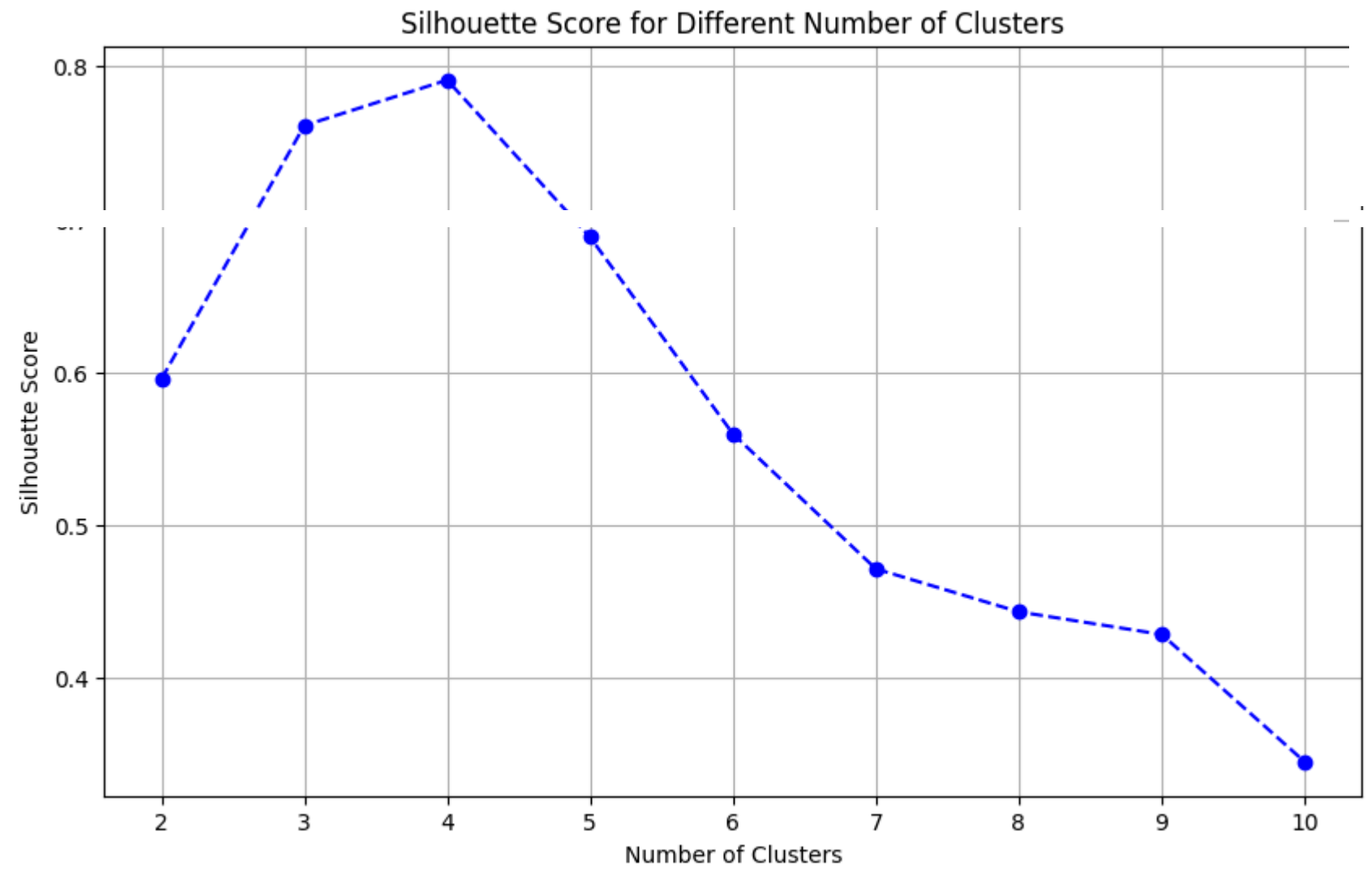

```
from sklearn.metrics.cluster import silhouette_score
print('Silhouette Score = ',silhouette_score(x,y))
```

```
Silhouette Score =  0.5343082164403306
```

```
X, _ = make_blobs(n_samples=500, n_features=2, centers=4, cluster_std=1.0, random_state=42)
range_n_clusters = list(range(2, 11))
silhouette_scores = []
for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    cluster_labels = kmeans.fit_predict(X)
    silhouette_avg = silhouette_score(X, cluster_labels)
    silhouette_scores.append(silhouette_avg)
    print(f"For n_clusters = {n_clusters}, the average silhouette score is {silhouette_avg:.3f}")
plt.figure(figsize=(10, 6))
plt.plot(range_n_clusters, silhouette_scores, marker='o', color='b', linestyle='--')
plt.title('Silhouette Score for Different Number of Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.grid(True)
plt.show()
```

```
best_n_clusters = range_n_clusters[np.argmax(silhouette_scores)]
print(f"The best number of clusters is {best_n_clusters} with a silhouette score of {max(silhouette_scores):.3f}"
```

```
For n_clusters = 2, the average silhouette score is 0.596
For n_clusters = 3, the average silhouette score is 0.761
For n_clusters = 4, the average silhouette score is 0.791
For n_clusters = 5, the average silhouette score is 0.688
For n_clusters = 6, the average silhouette score is 0.560
For n_clusters = 7, the average silhouette score is 0.471
For n_clusters = 8, the average silhouette score is 0.443
For n_clusters = 9, the average silhouette score is 0.429
For n_clusters = 10, the average silhouette score is 0.345
```




```
The best number of clusters is 4 with a silhouette score of 0.791
```

```
data = A.copy()
data = data.iloc[:,[2,3]]
data.head()
```

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

Next steps: Generate code with `data` | View recommended plots | New interactive sheet

```
from scipy.cluster.hierarchy import dendrogram, linkage
np.random.seed(42)
data = np.random.rand(10, 3)
Z = linkage(data, method='ward')
plt.figure(figsize=(10, 6))
dendrogram(Z)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```