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

from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import silhouette_score, davies_bouldin_score
from scipy.cluster.hierarchy import dendrogram, linkage
```

```
df = pd.read_csv('Mall_Customers.csv')
print(df.shape)
df.head()
```

```
le = LabelEncoder()
df['Genre'] = le.fit_transform(df['Genre'])  # Encode Genre instead of Gender

scaler = StandardScaler()
df[['Annual Income (k$)', 'Spending Score (1-100)']] = scaler.fit_transform(
    df[['Annual Income (k$)', 'Spending Score (1-100)']]
)
```

```
df.head()
```

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

```
wcss = []
sil_scores = []
for k in range(2,11):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
    sil_scores.append(silhouette_score(X, kmeans.labels_))

plt.plot(range(2,11), wcss, marker='o')
plt.title('Elbow Method')
plt.xlabel('k')
plt.ylabel('WCSS')
plt.show()
```

```
plt.plot(range(2,11), sil_scores, marker='o')
plt.title('Silhouette Score vs k')
plt.xlabel('k')
plt.ylabel('Silhouette Score')
plt.show()
```

```
kmeans = KMeans(n_clusters=5, random_state=42, n_init=10)
df['KMeans_Cluster'] = kmeans.fit_predict(X)

sns.scatterplot(x=X['Annual Income (k$)'], y=X['Spending Score (1-100)'],
                hue=df['KMeans_Cluster'], palette='tab10')
plt.title('Customer Segments (KMeans)')
plt.show()
```

```
linkage_matrix = linkage(X, method='ward')
plt.figure(figsize=(10,5))
dendrogram(linkage_matrix)
plt.title('Dendrogram')
plt.show()
```

```
agg = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
df['Agglo_Cluster'] = agg.fit_predict(X)

sns.scatterplot(x=X['Annual Income (k$)'], y=X['Spending Score (1-100)'],
                hue=df['Agglo_Cluster'], palette='tab10')
plt.title('Customer Segments (Agglomerative)')
plt.show()
```

```
print('KMeans Silhouette:', silhouette_score(X, df['KMeans_Cluster']))
print('KMeans Davies-Bouldin:', davies_bouldin_score(X, df['KMeans_Cluster']))

print('Agglo Silhouette:', silhouette_score(X, df['Agglo_Cluster']))
print('Agglo Davies-Bouldin:', davies_bouldin_score(X, df['Agglo_Cluster']))
```

```
sil_vals = []
for seed in range(10):
    km = KMeans(n_clusters=5, random_state=seed, n_init=10)
    labels = km.fit_predict(X)
    sil_vals.append(silhouette_score(X, labels))

print('Silhouette CV mean:', np.mean(sil_vals))
print('Silhouette CV std:', np.std(sil_vals))
```

```
cluster_summary = df.groupby('KMeans_Cluster')[['Annual Income (k$)', 'Spending Score (1-100)']].mean()
cluster_summary
```