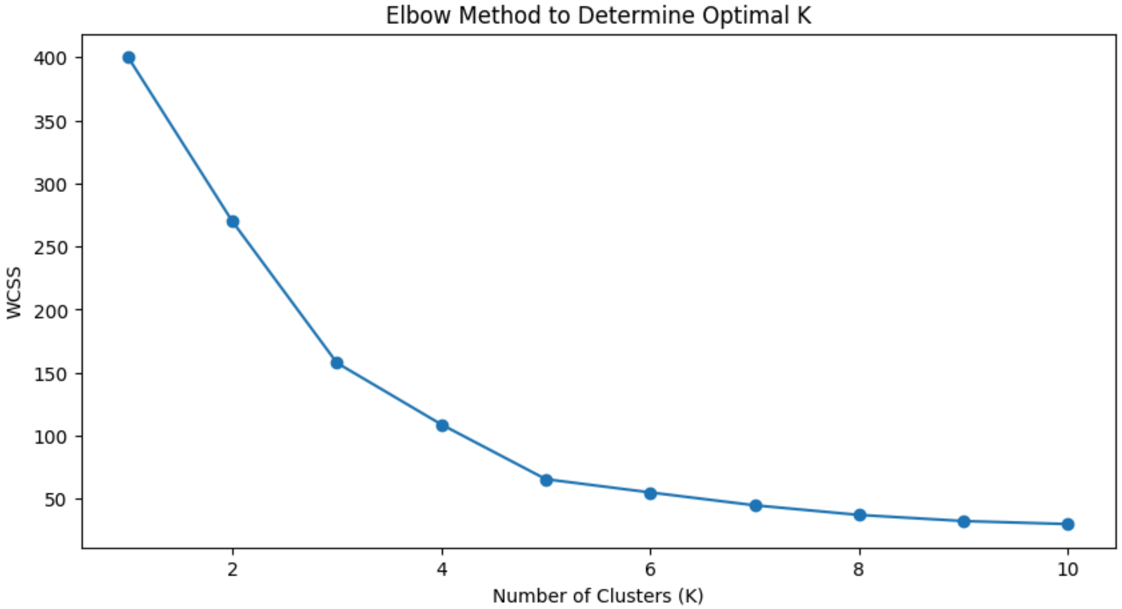
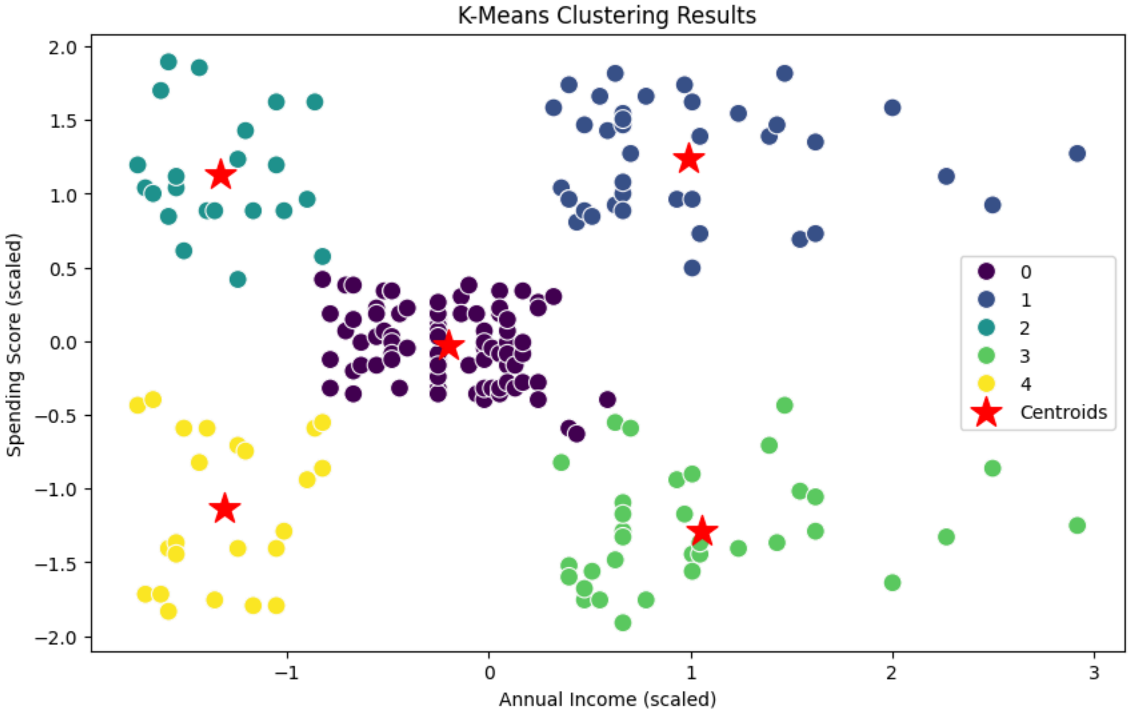
```
# K-Means Clustering Implementation

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from google.colab import files


uploaded = files.upload()

filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

# selecting numerical features for clustering
X = df.iloc[:, [3, 4]].values  # Assuming 'Annual Income' and 'Spending Score'

# scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# elbow method to determine optimal K
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# ploting Elbow Graph
plt.figure(figsize=(10,5))
plt.plot(range(1, 11), wcss, marker='o', linestyle='-')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.title('Elbow Method to Determine Optimal K')
plt.show()

# applying KMeans with K=5
kmeans = KMeans(n_clusters=5, init='k-means++', max_iter=300, n_init=10, random_state=42)
clusters = kmeans.fit_predict(X_scaled)

# adding cluster labels to dataset
df['Cluster'] = clusters

# visualizing Clusters
plt.figure(figsize=(10,6))
sns.scatterplot(x=X_scaled[:, 0], y=X_scaled[:, 1], hue=clusters, palette='viridis', s=100)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', label='Centroids', marker='*')
plt.xlabel('Annual Income (scaled)')
plt.ylabel('Spending Score (scaled)')
plt.title('K-Means Clustering Results')
plt.legend()
plt.show()
```

Choose Files Mall_Customers.csv

- **Mall_Customers.csv**(text/csv) - 3981 bytes, last modified: 3/19/2025 - 100% done

```
Saving Mall_Customers.csv to Mall_Customers (1).csv
```


Elbow Method to Determine Optimal K
WCSS
Number of Clusters (K)


K-Means Clustering Results
Spending Score (scaled)
Annual Income (scaled)
Centroids