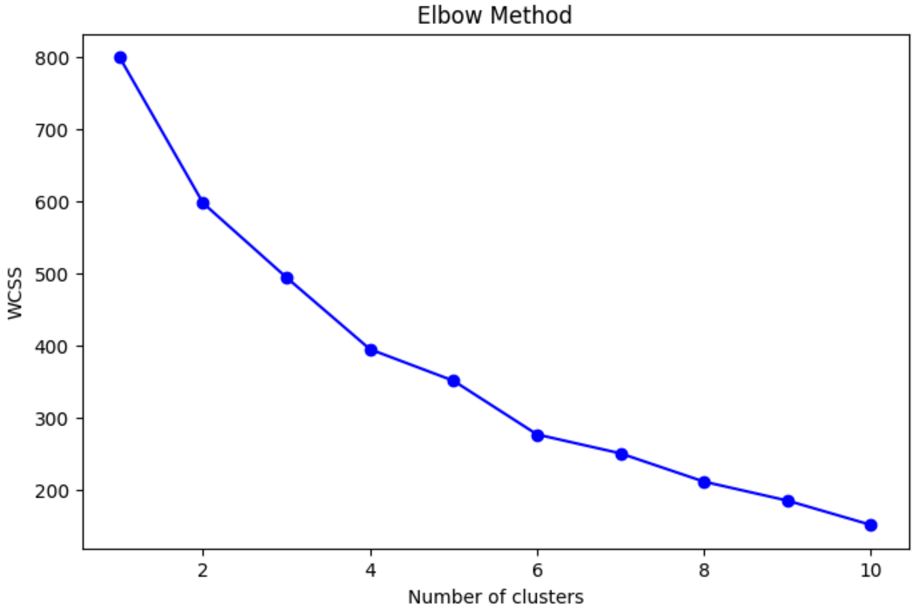
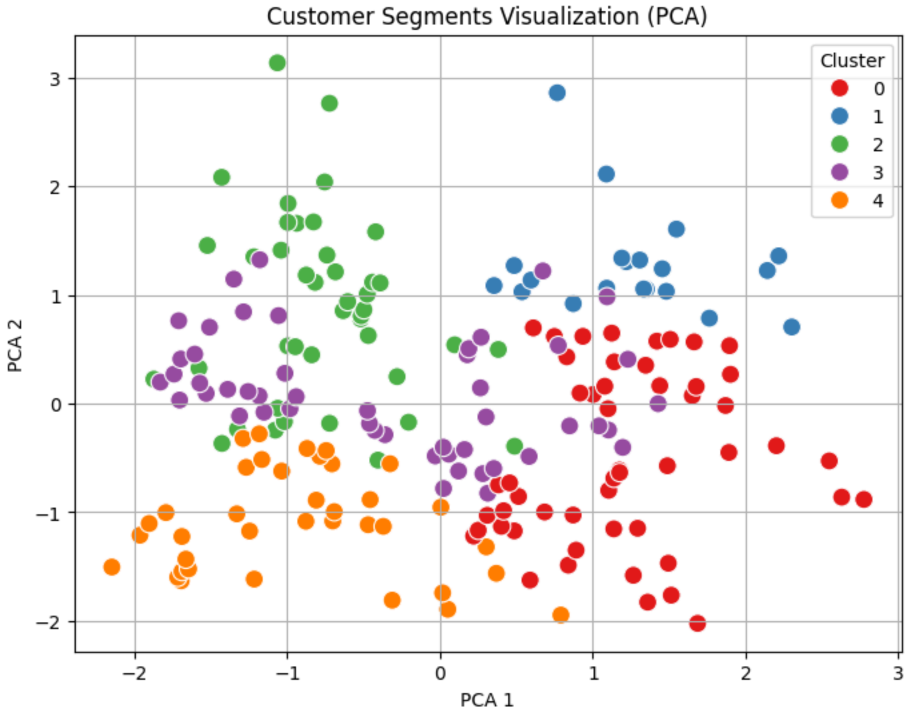
```
# Step 1: Install Required Libraries (if needed)
!pip install seaborn --quiet

# Step 2: Import Required Libraries
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 sklearn.decomposition import PCA

# Step 3: Load Dataset (Mall Customers Dataset from GitHub)

data = pd.read_csv("/content/Mall_Customers.csv")
data.head()

# Step 4: Explore the Data
print("\nDataset Info:")
print(data.info())

print("\nSummary Statistics:")
print(data.describe())

# Step 5: Encode Gender Column
data['Genre'] = data['Genre'].map({'Male': 0, 'Female': 1})

# Step 6: Feature Selection
X = data[['Genre', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)']]

# Optional: Scale the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 7: Find Optimal Number of Clusters using Elbow Method
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot the Elbow Curve
plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), wcss, 'bo-')
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Step 8: Apply KMeans Clustering
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=42)
clusters = kmeans.fit_predict(X_scaled)
data['Cluster'] = clusters

# Step 9: Visualize Clusters using PCA for 2D
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(8, 6))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=clusters, palette='Set1', s=100)
plt.title('Customer Segments Visualization (PCA)')
plt.xlabel('PCA 1')
plt.ylabel('PCA 2')
plt.legend(title='Cluster')
plt.grid(True)
plt.show()

# Step 10: View Clustered Data
print("\nClustered Data Sample:")
print(data.groupby('Cluster').mean())
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
None

Summary Statistics:
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
```






```
Clustered Data Sample:
        CustomerID     Genre        Age  Annual Income (k$)  \
Cluster
0        65.333333  0.490196  56.470588           46.098039
```

```
1        159.500000  0.000000  39.500000           85.150000
2        100.809524  0.000000  28.690476           60.904762
3        151.510204  1.000000  37.897959           82.122449
4         50.526316  1.000000  27.315789           38.842105

         Spending Score (1-100)
Cluster
0                     39.313725
1                     14.050000
2                     70.238095
3                     54.448980
4                     56.210526
```

Start coding or generate with AI.