Assignment 2: K-Means Cluster Analysis

Yu-Chen Su

1. **Download** the mall customer dataset, Mall_Customers.csv

   Import pandas to read file and show data head

[1]:
```
import pandas as pd
df = pd.read_csv('Mall_Customers-1.csv')
df.head()
```

[1]:

	Age	Annual Income	Spending Score
0	19	15	39
1	21	15	81
2	20	16	6
3	23	16	77
4	31	17	40

2. Determine the **data dimensionality** by finding the following (5pts):
   A. Total number of customers.

[3]:
```
# Display the count
num_rows = len(df)
print(f"Total number of customers: {num_rows}")
df.count()
```

```
Total number of customers: 200
```

[3]:
```
Age               200
Annual Income     200
Spending Score    200
dtype: int64
```

   B. Number of attributes (categories).

[5]:
```
# Display the column (categories)
num_column = df.shape[1]
print(f"Total number of categories: {num_column}")
```

```
Total number of categories: 3
```

C. Data types.

```
[7]: # Check data types of each column
     print(df.dtypes)
```

```
Age                int64
Annual Income      int64
Spending Score     int64
dtype: object
```

D. Missing values.

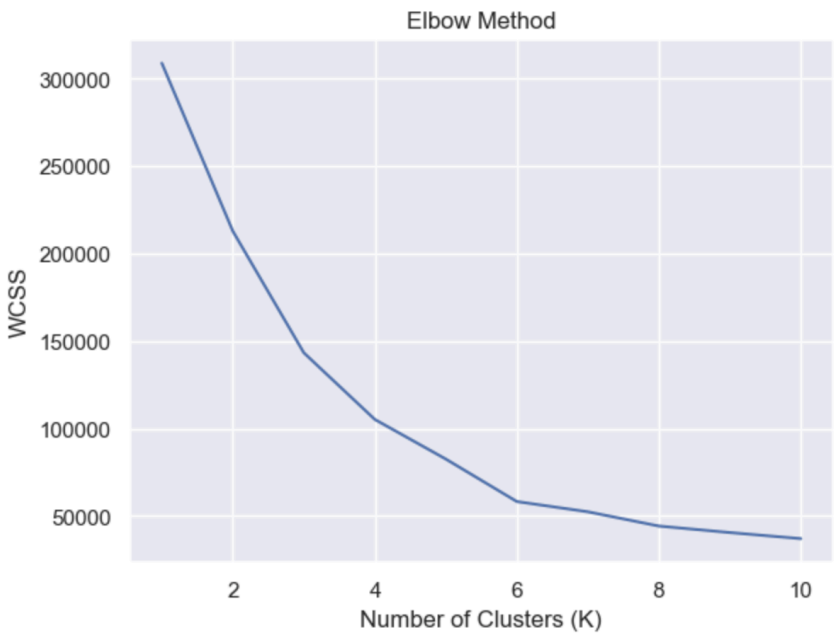
```
[11]: missing_values = df.isnull().sum()
      print(missing_values)
```

```
Age                0
Annual Income      0
Spending Score     0
dtype: int64
```

3. Determine the **optimal K value** and explain the following (5pts):
   A. Method utilized to determine optimal K.

Use elbow method (set `random_state=0`)

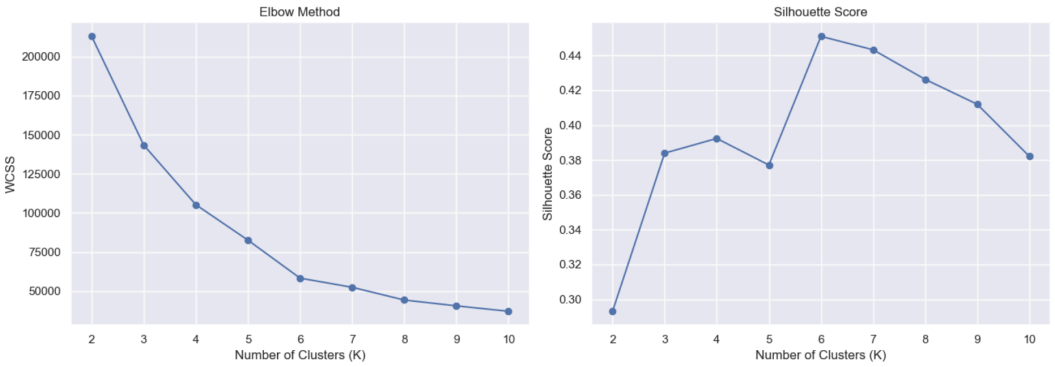
```
•[17]: #Elbow methods (set random_state=0)
       wcss = []
       for k in range(1, 11):
           kmeans = KMeans(n_clusters=k, random_state=0)
           kmeans.fit(df)
           wcss.append(kmeans.inertia_)

       sns.set()
       plt.plot(range(1, 11), wcss)
       plt.xlabel('Number of Clusters (K)')
       plt.ylabel('WCSS')
       plt.title('Elbow Method')
       plt.show()
```



B. K value selected.

The Elbow method suggests that the optimal K value might be around 6, so I used the Silhouette Score to further validate this choice.

[33]:
```
wcss = []
silhouette_scores = []

# Evaluate for a range of K
k_range = range(2, 11)

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(df)
    wcss.append(kmeans.inertia_)  # WCSS for the Elbow Method
    silhouette_scores.append(silhouette_score(df, kmeans.labels_))  # Silhouette Score

# Plotting the Elbow Method results
plt.figure(figsize=(14, 5))

plt.subplot(1, 2, 1)
plt.plot(k_range, wcss, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.title('Elbow Method')

# Plotting the Silhouette Scores
plt.subplot(1, 2, 2)
plt.plot(k_range, silhouette_scores, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score')

plt.tight_layout()
plt.show()

# Determining the optimal K
optimal_k = k_range[silhouette_scores.index(max(silhouette_scores))]
print(f"Optimal number of clusters according to Silhouette Score: {optimal_k}")
```




```
Optimal number of clusters according to Silhouette Score: 6
```

I determined the optimal K value to be 6, as it resulted in the highest Silhouette Score in this case.

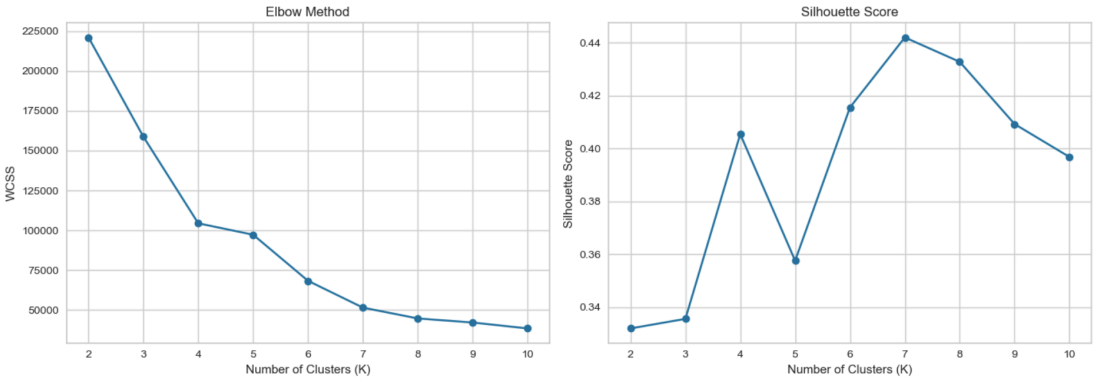
4. **Apply K-Means Cluster Analysis.** (5pts)

A. Is this an accurate result? Explain your reasoning.

K-Means is not a method with 100% accuracy, as many factors can affect the results. For example, I experimented with different

`random_state` values, which control the randomness in the initialization of centroids. In this case, the optimal K value is 7, as shown in the following graphs.

```
wcss = []
silhouette_scores = []

# Evaluate for a range of K
k_range = range(2, 11)

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df)
    wcss.append(kmeans.inertia_)  # WCSS for the Elbow Method
    silhouette_scores.append(silhouette_score(df, kmeans.labels_))  # Silhouette Score

# Plotting the Elbow Method results
plt.figure(figsize=(14, 5))

plt.subplot(1, 2, 1)
plt.plot(k_range, wcss, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.title('Elbow Method')

# Plotting the Silhouette Scores
plt.subplot(1, 2, 2)
plt.plot(k_range, silhouette_scores, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score')

plt.tight_layout()
plt.show()

# Determining the optimal K
optimal_k = k_range[silhouette_scores.index(max(silhouette_scores))]
print(f"Optimal number of clusters according to Silhouette Score: {optimal_k}")
```




```
Optimal number of clusters according to Silhouette Score: 7
```

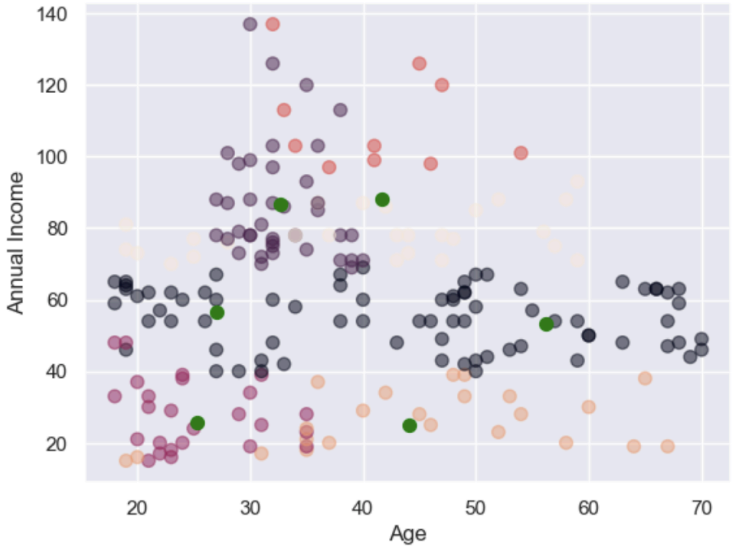
Moreover, the Silhouette Scores in this case are all below 0.45, suggesting that the clusters are not very well separated.

K-Means may not be perfectly accurate, but it still helps identify potential clusters or relationships within the data.

B. Describe the clusters. Be as specific as possible.

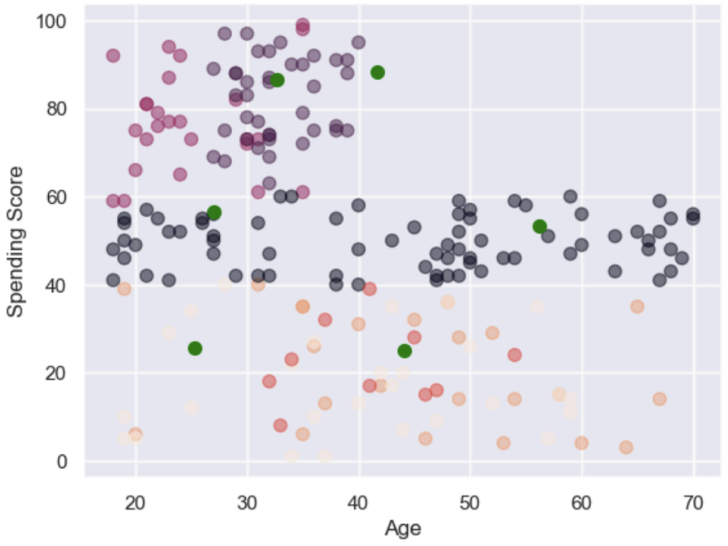
First, I used 2D scatter plots to explore the relationships between the three attributes (categories) individually. The data is divided into six clusters, each represented by a different color.

[39]:
```
plt.scatter(df['Age'], df['Annual Income'], c= kmeans.labels_.astype(float), s=50,
alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='green', s=50)
plt.xlabel('Age')
plt.ylabel('Annual Income')
plt.show()
```



It's difficult to directly identify a pattern for the clusters in this chart.

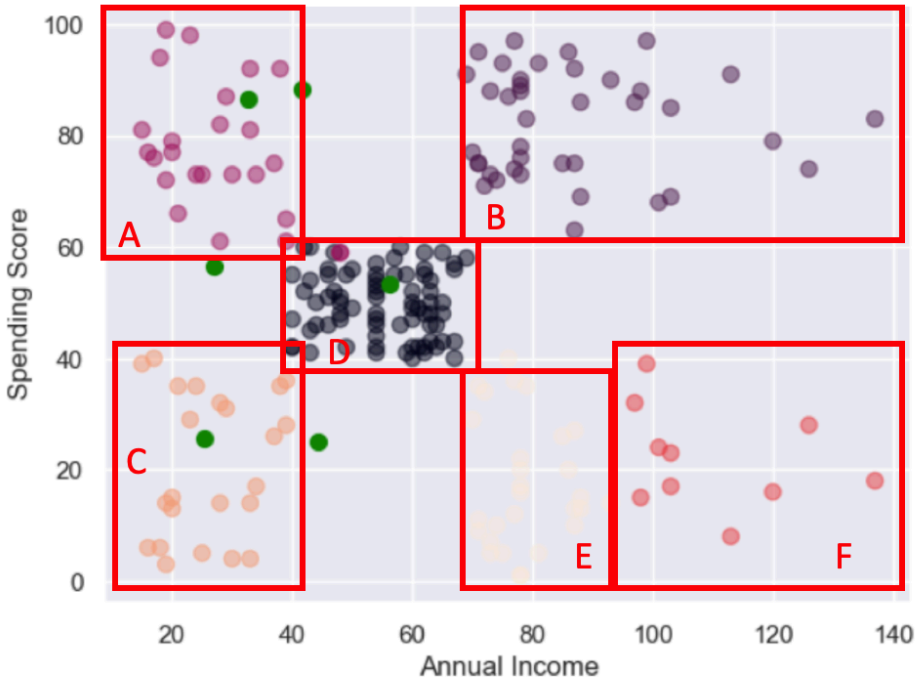
[43]:
```
plt.scatter(df['Age'], df['Spending Score'], c= kmeans.labels_.astype(float), s=50,
alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='green', s=50)
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.show()
```



It's difficult to directly identify a pattern for the clusters in this chart.

```
[41]: plt.scatter(df['Annual Income'], df['Spending Score'], c= kmeans.labels_.astype(float), s=50,
alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='green', s=50)
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.show()
```

In the scatterplot comparing Spending Score and Annual Income, the clusters are well-separated. We can observe six distinct groups (based on the optimal K value identified in question 4), labeled A through F.



Cluster A: Low annual income, high spending score.

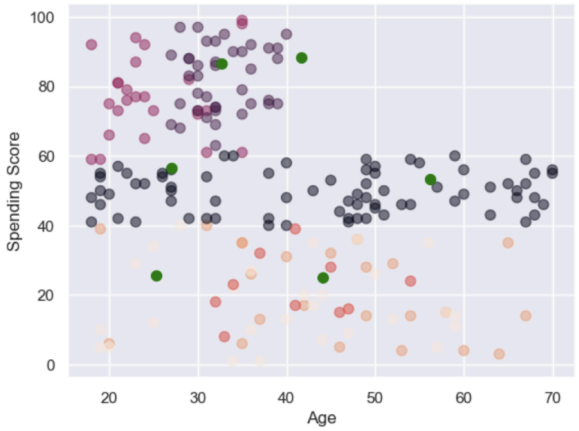
Cluster B: High annual income, high spending score.

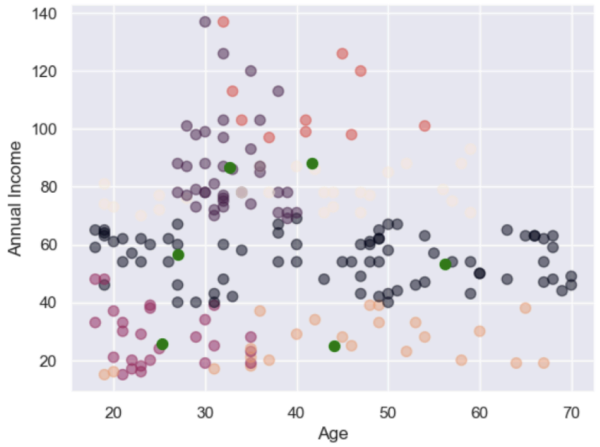
Cluster C: Low annual income, low spending score.

Cluster D: Mid-level annual income, mid-level spending score.

Cluster E: Mid-high annual income, low spending score.

Cluster F: High annual income, low spending score.





Combined with graphs together, we will see the **ages** of clusters

Cluster A: Low annual income, high spending score, with ages around 15 to 35.

Cluster B: High annual income, high spending score, with ages around 25 to 40.

Cluster C: Low annual income, low spending score, spanning all ages in the data.

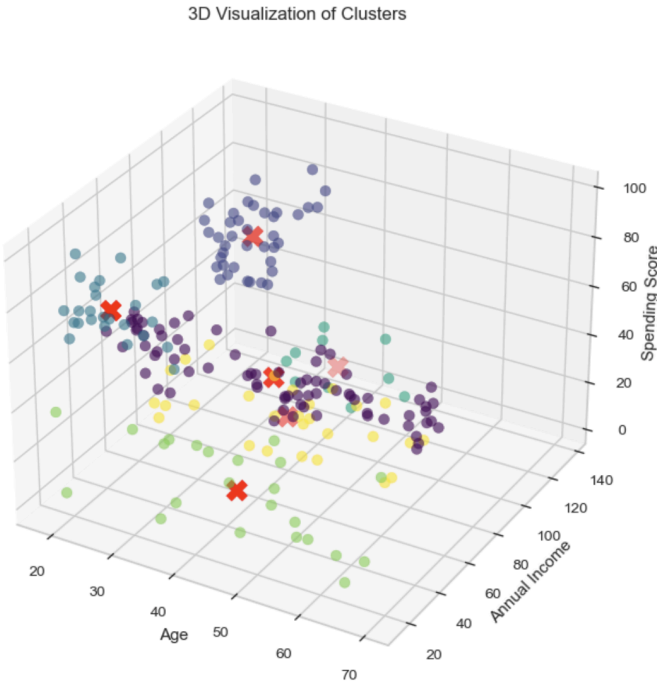
Cluster D: Mid-level annual income, mid-level spending score, spanning all ages in the data.

Cluster E: Mid-high annual income, low spending score, spanning all ages in the data.

Cluster F: High annual income, low spending score, with ages around 30 to 55.

I also created a 3D graph for this data. Although it's not easy to distinguish all the data points, it's still possible to identify the general locations of the clusters.

```
[49]: # Get cluster labels and centroids
labels = kmeans.labels_
centroids = kmeans.cluster_centers_

# Plotting the clusters in 3D
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot for data points
ax.scatter(df['Age'], df['Annual Income'], df['Spending Score'], c=labels, s=50, alpha=0.6, cmap='viridis')

# Scatter plot for centroids
ax.scatter(centroids[:, 0], centroids[:, 1], centroids[:, 2], c='red', s=200, marker='X')

# Setting labels
ax.set_xlabel('Age')
ax.set_ylabel('Annual Income')
ax.set_zlabel('Spending Score')
ax.set_title('3D Visualization of Clusters')

plt.show()
```



C. Assume you are a supervisor who is provided the cluster descriptions by an employee. Are there distinct customer clusters that can be used for future advertising, or not? Explain your reasoning.

Yes, the customer clusters can be used for future advertising.

Cluster A: Low annual income, high spending score, with ages around 15 to 35.

Cluster B: High annual income, high spending score, with ages around 25 to 40.

Cluster C: Low annual income, low spending score, spanning all ages in the data.

Cluster D: Mid-level annual income, mid-level spending score, spanning all ages in the data. (have the most biggest amount of data)

Cluster E: Mid-high annual income, low spending score, spanning all ages in the data.

Cluster F: High annual income, low spending score, with ages around 30 to 55.

Based on the characteristics of the clusters we've identified, if we want to maintain high spending scores, we should focus on Clusters A and B, which are also relatively young. To improve the overall spending score, we should target Clusters C, D, E, and F. Among them, Cluster D has the largest number of customers, so increasing their spending score would have a significant impact. Additionally, Clusters E and F are promising for advertising, given their higher incomes. Cluster F, in particular, is the best target due to its specific age range and high incomes.

In conclusion, clusters can assist in targeting specific customer groups and help in planning strategies tailored to different income levels and age ranges.