In [2]:
```python
# Task 3: Customer Segmentation using K-Means Clustering

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

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.metrics import silhouette_score
```

In [3]:
```python
# -------------------- Step 1: Load Dataset --------------------
# Working public dataset URL (UCI mirror)
url = "Mall_Customers.csv"
df = pd.read_csv(url)

print("Sample Data:\n", df.head())
```

```
Sample Data:
    CustomerID  Gender  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
```

In [4]:
```python
# -------------------- Step 2: Preprocessing --------------------
# Drop 'CustomerID'
df = df.drop('CustomerID', axis=1)
```

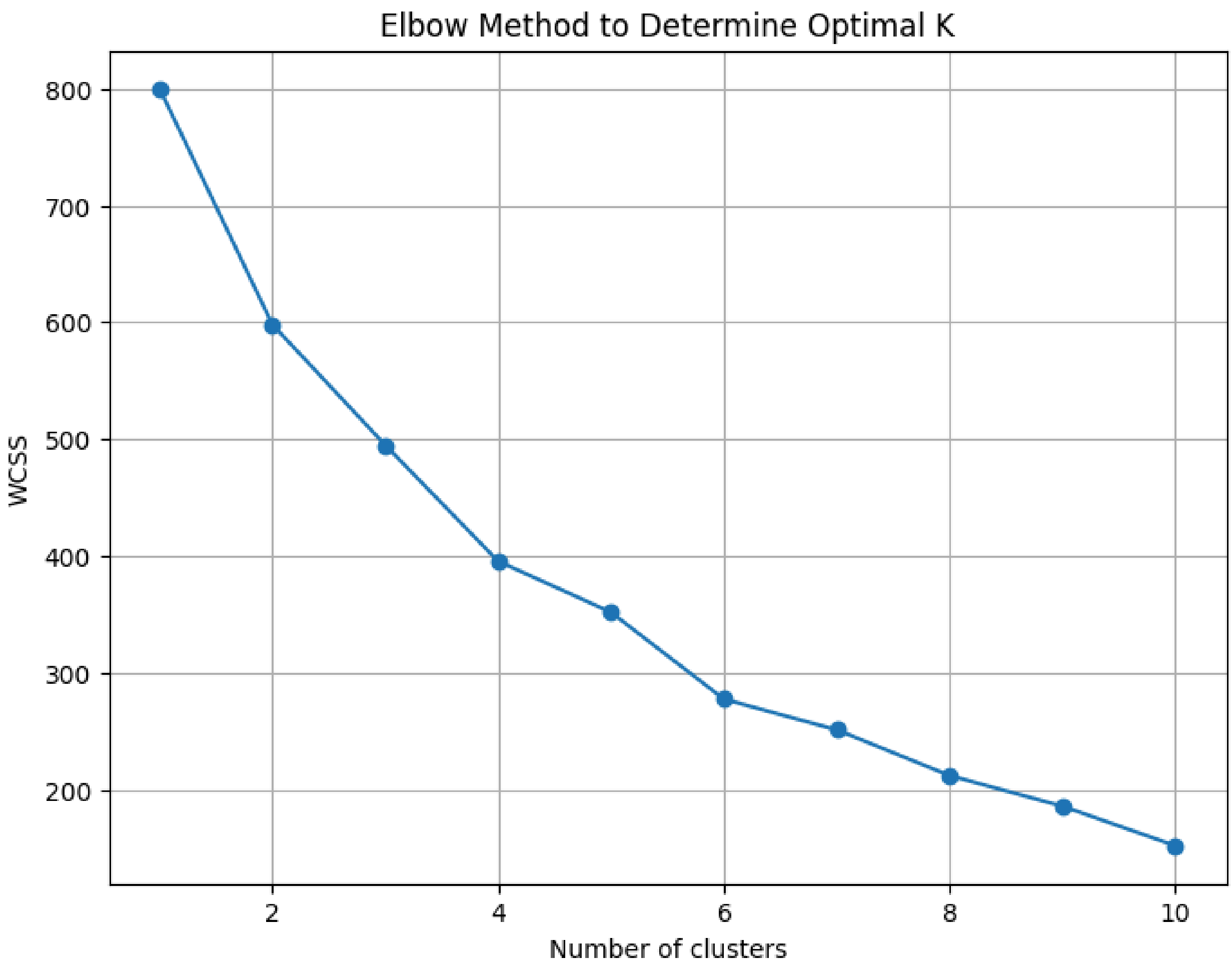
In [5]:
```python
# Encode Gender
df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
```

In [6]:
```python
# Scale features
scaler = StandardScaler()
scaled_features = scaler.fit_transform(df)
```

In [7]:
```python
# -------------------- Step 3: Elbow Method --------------------
wcss = []
for i in range(1, 11):
    km = KMeans(n_clusters=i, random_state=42)
    km.fit(scaled_features)
    wcss.append(km.inertia_)
```

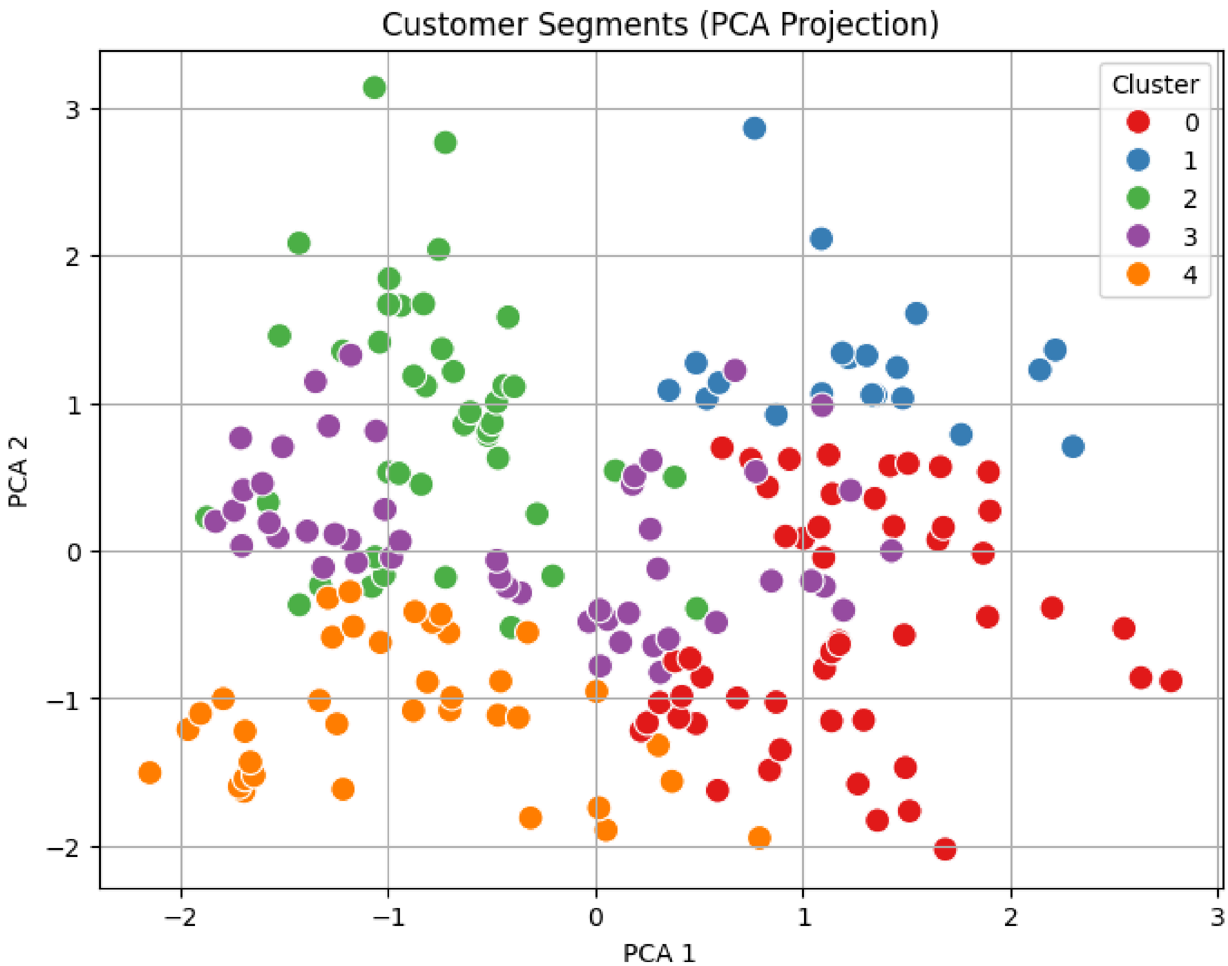
In [8]:
```python
# Plot Elbow Curve
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), wcss, marker='o')
plt.title("Elbow Method to Determine Optimal K")
plt.xlabel("Number of clusters")
plt.ylabel("WCSS")
```

```
plt.grid(True)
plt.show()
```

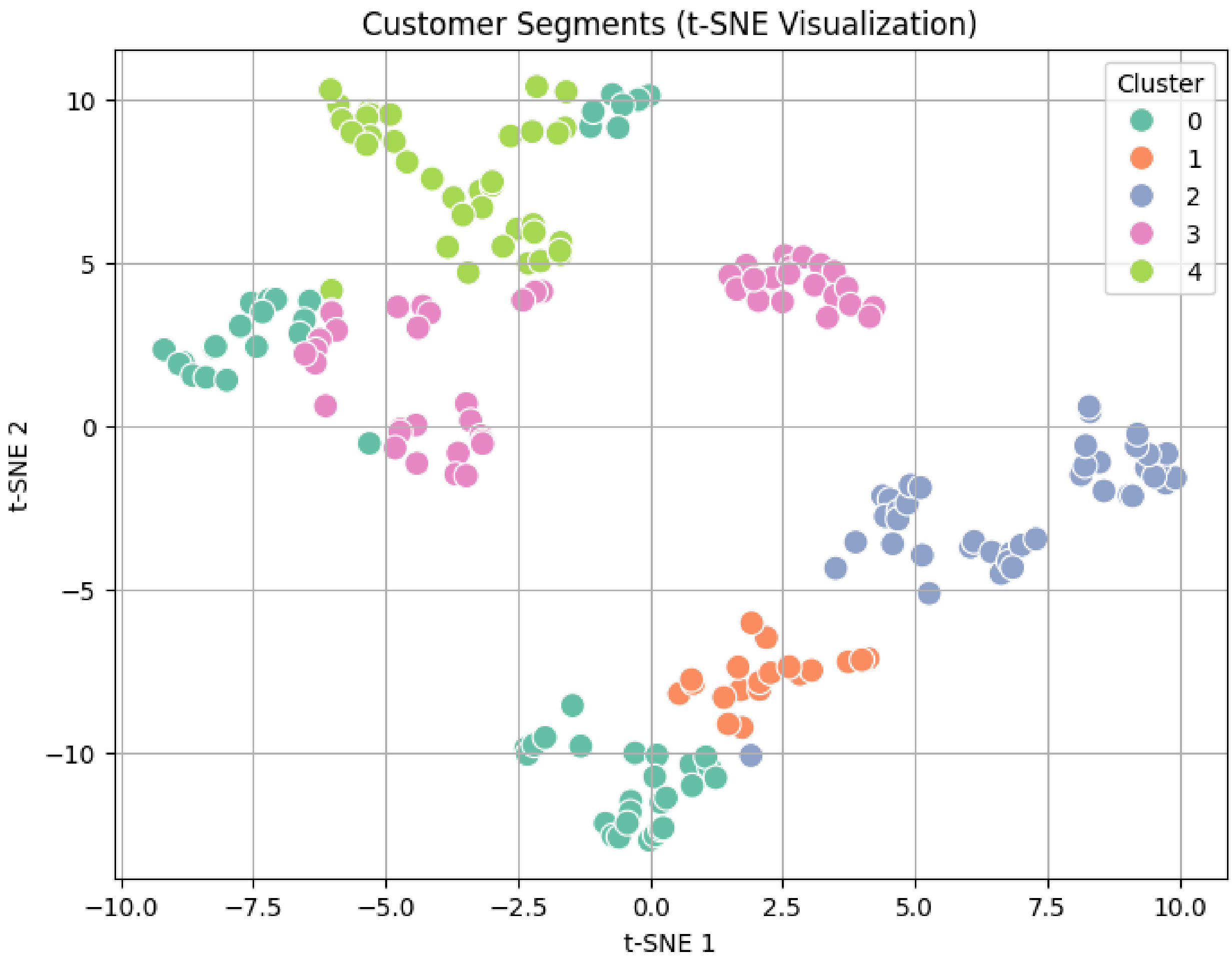


In [9]:
```
# -------------------- Step 4: Apply K-Means --------------------
optimal_k = 5
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
clusters = kmeans.fit_predict(scaled_features)

# Add cluster label
df['Cluster'] = clusters
```

In [10]:
```
# -------------------- Step 5: PCA Visualization --------------------
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_features)

plt.figure(figsize=(8, 6))
sns.scatterplot(x=pca_data[:, 0], y=pca_data[:, 1], hue=df['Cluster'], palette
plt.title("Customer Segments (PCA Projection)")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.legend(title="Cluster")
plt.grid(True)
plt.show()
```



In [12]:
```
from sklearn.manifold import TSNE

# Correct keyword: use max_iter instead of n_iter
tsne = TSNE(n_components=2, perplexity=30, max_iter=300, random_state=42)
tsne_data = tsne.fit_transform(scaled_features)

# Visualization
plt.figure(figsize=(8, 6))
sns.scatterplot(x=tsne_data[:, 0], y=tsne_data[:, 1], hue=df['Cluster'], palet
plt.title("Customer Segments (t-SNE Visualization)")
plt.xlabel("t-SNE 1")
plt.ylabel("t-SNE 2")
plt.legend(title="Cluster")
plt.grid(True)
plt.show()
```



In [13]:

```
# -------------------- Step 7: Cluster Summary --------------------
summary = df.groupby('Cluster').mean().round(2)
print("\nCluster Summary (Average Feature Values):\n", summary)
```

```
Cluster Summary (Average Feature Values):
         Gender    Age  Annual Income (k$)  Spending Score (1-100)
Cluster
0          0.51  56.47               46.10                   39.31
1          1.00  39.50               85.15                   14.05
2          1.00  28.69               60.90                   70.24
3          0.00  37.90               82.12                   54.45
4          0.00  27.32               38.84                   56.21
```

In [14]:

```
# -------------------- Step 8: Silhouette Score --------------------
score = silhouette_score(scaled_features, df['Cluster'])
print(f"\nSilhouette Score: {round(score, 3)}")
```

```
Silhouette Score: 0.272
```

In [ ]:

Task 3: Customer Segmentation using K-Means Clustering

Dataset Used: Mall_Customers.csv
Objective: Segment customers into different behavioral groups using unsupervised machine learning techniques.

Dataset Overview

The dataset contains information about 200 mall customers with features:

- **Gender**
- **Age**
- **Annual Income (k$)**
- **Spending Score (1–100)**

Model Chosen: K-Means Clustering

Why K-Means?
K-Means is a popular unsupervised learning algorithm for clustering:

- It groups data into `k` clusters based on similarity.
- It is computationally efficient and works well for low-dimensional data like this.

Preprocessing Steps

- **Removed** the `CustomerID` column (not relevant for clustering).
- **Encoded** `Gender` as binary (Male = 1, Female = 0).
- **Scaled** features using `StandardScaler` to ensure equal weighting.

Optimal Clusters: Elbow Method

- The **Elbow Curve** indicated an optimal number of clusters = **5**.
- We used `KMeans(n_clusters=5)` with a fixed random state for reproducibility.

Visualization

Method	Purpose	Outcome
PCA	Reduce dimensions to 2D for plotting	Revealed well-separated clusters
t-SNE	Non-linear visualization of structure	Confirmed cluster distinction visually

---

Cluster Insights (Mean Values)

Cluster	Gender	Age	Income	Spending Score
0	0.51	56.47	46.10	39.31
1	1.00	39.50	85.15	14.05
2	1.00	28.69	60.90	70.24
3	0.00	37.90	82.12	54.45
4	0.00	27.32	38.84	56.21

> Indicates different clusters represent **high spenders**, **low spenders**, and **older customers with average spend**.

Evaluation Metric

- **Silhouette Score**: **0.272**
  Indicates **fair clustering structure**. Higher scores (0.5+) could be achieved with more complex clustering algorithms.

Conclusion

K-Means clustering was successfully applied to segment customers. Each group shows distinct spending and income behavior, useful for targeted marketing strategies. Visualizations (PCA and t-SNE) helped confirm the cluster separations.

For future enhancement:

- Apply advanced clustering (DBSCAN, Hierarchical)
- Use additional behavioral or demographic features
- Conduct cluster profiling for business recommendations
