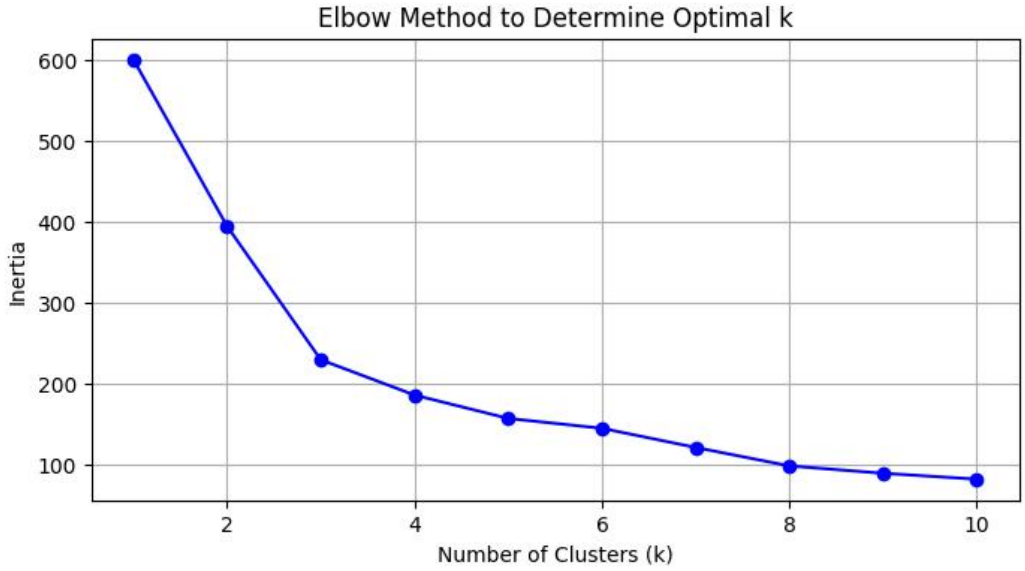
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

np.random.seed(42)
n_customers = 300
data = {
    'CustomerID': np.arange(1, n_customers + 1),
    'TotalSpent': np.random.gamma(shape=2, scale=200, size=n_customers),
    'VisitFrequency': np.random.poisson(lam=10, size=n_customers)
}
df = pd.DataFrame(data)

features = df[['TotalSpent', 'VisitFrequency']]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

inertia = []
K = range(1, 11)
for k in K:
    km = KMeans(n_clusters=k, random_state=42)
    km.fit(scaled_features)
    inertia.append(km.inertia_)


plt.figure(figsize=(8, 4))
plt.plot(K, inertia, 'bo-')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method to Determine Optimal k')
plt.grid(True)
plt.show()

kmeans = KMeans(n_clusters=4, random_state=42)
df['Cluster'] = kmeans.fit_predict(scaled_features)
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='TotalSpent', y='VisitFrequency', hue='Cluster', palette='Set2', s=100)
plt.title('Customer Segments Based on Spending Patterns')
plt.xlabel('Total Amount Spent')
plt.ylabel('Visit Frequency')
plt.grid(True)
plt.legend(title='Cluster')
plt.show()
cluster_summary = df.groupby('Cluster')[['TotalSpent', 'VisitFrequency']].mean().round(2)
print("\nCluster Summary:")
print(cluster_summary)
```

OUTPUT



Cluster Summary:

Cluster	TotalSpent	VisitFrequency
0	210.78	12.95
1	299.54	7.10
2	999.18	9.63
3	567.84	11.90