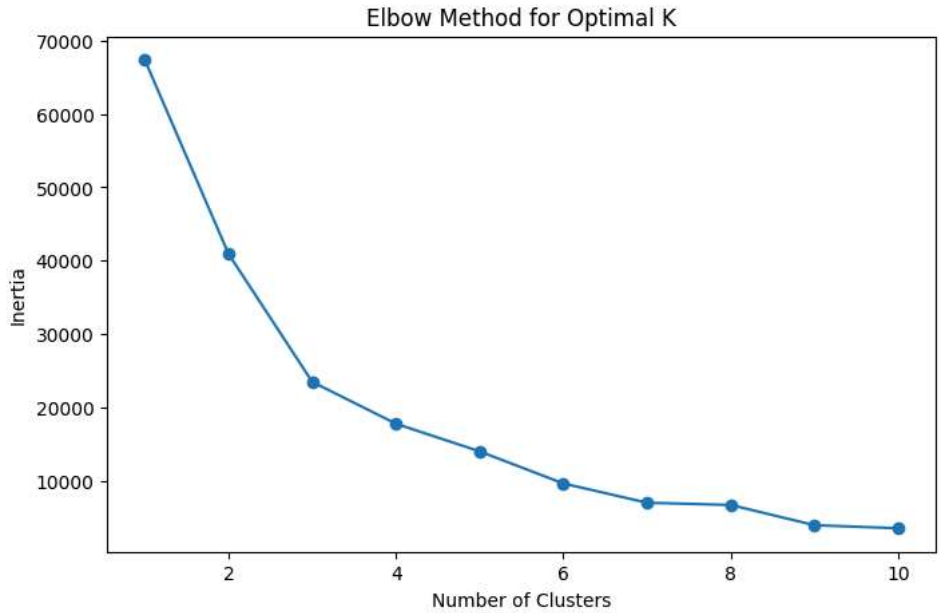
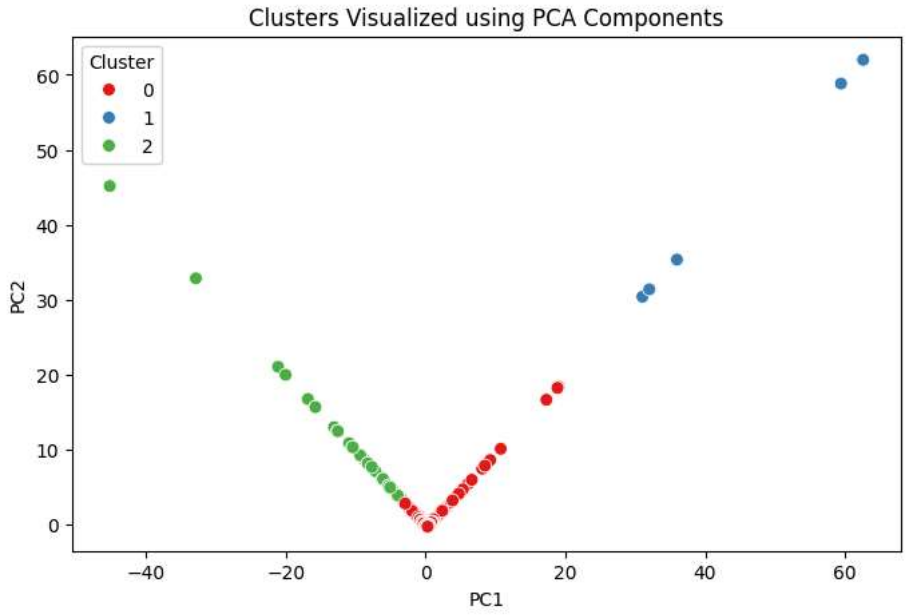
```
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
url = 'https://raw.githubusercontent.com/SagarPatel98/Customer-Segmentation-using-Machine-Learning/master/Files/int_online_tx.csv'
data = pd.read_csv(url)

# Display first few rows of the dataset
print(data.head())

# Handling missing values
data = data.dropna()

# Standardizing numerical columns
numerical_features = ['Quantity', 'UnitPrice']  # Replace with your numerical columns
scaler = StandardScaler()
data[numerical_features] = scaler.fit_transform(data[numerical_features])

pca = PCA(n_components=2)  # Reduce to 2 dimensions for visualization
pca_data = pca.fit_transform(data[numerical_features])

# Create a new dataframe for PCA-transformed data
pca_df = pd.DataFrame(data=pca_data, columns=['PC1', 'PC2'])

# Determine the optimal number of clusters using the Elbow Method
inertia = []
range_clusters = range(1, 11)
for k in range_clusters:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(pca_df)
    inertia.append(kmeans.inertia_)

# Plot the Elbow Method
plt.figure(figsize=(8, 5))
plt.plot(range_clusters, inertia, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()

# Apply K-Means with the optimal number of clusters (choose based on Elbow Method)
optimal_clusters = 3  # Replace with the chosen number
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
clusters = kmeans.fit_predict(pca_df)
print(clusters)

# Add cluster labels to the dataframe
pca_df['Cluster'] = clusters
data['Cluster'] = clusters

# Scatterplot of PCA Components with Clusters
plt.figure(figsize=(8, 5))
sns.scatterplot(x='PC1', y='PC2', hue='Cluster', data=pca_df, palette='Set1', s=50)
plt.title('Clusters Visualized using PCA Components')
plt.show()

# 2. Silhouette Score Visualization
silhouette_avg = silhouette_score(pca_df[['PC1', 'PC2']], pca_df['Cluster'])
print(f'Silhouette Score: {silhouette_avg:.2f}')

# Cluster Size Distribution
plt.figure(figsize=(8, 5))
sns.countplot(x='Cluster', data=data, palette='Set2')
plt.title('Cluster Size Distribution')
plt.xlabel('Cluster')
plt.ylabel('Count')
plt.show()

# Boxplot of a Numerical Feature by Cluster
```

```
plt.figure(figsize=(8, 5))
sns.boxplot(x='Cluster', y='Quantity', data=data, palette='Set3')
plt.title('Quantity Distribution by Cluster')
plt.show()

# PCA Explained Variance
explained_variance = pca.explained_variance_ratio_
plt.figure(figsize=(8, 5))
plt.bar(['PC1', 'PC2'], explained_variance, color='skyblue')
plt.title('PCA Explained Variance')
plt.ylabel('Variance Ratio')
plt.show()

# Heatmap of Cluster Centers
cluster_centers = pd.DataFrame(kmeans.cluster_centers_, columns=['PC1', 'PC2'])
plt.figure(figsize=(8, 5))
sns.heatmap(cluster_centers.T, cmap='coolwarm', annot=True)
plt.title('Cluster Centers Heatmap')
plt.show()


# Top 10 Selling Products by Quantity
top_products = data.groupby('Description')['Quantity'].sum().sort_values(ascending=False).head(10)
plt.figure(figsize=(10, 6))
top_products.plot(kind='bar', color='skyblue')
plt.title('Top 10 Selling Products by Quantity')
plt.xlabel('Product Description')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Distribution of Quantity Purchased
plt.figure(figsize=(10, 6))
sns.histplot(data['Quantity'], bins=50, kde=True, color='green')
plt.title('Distribution of Quantity Purchased')
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.show()
```

```
   InvoiceNo StockCode                          Description  Quantity  \
0     536370     22728       ALARM CLOCK BAKELIKE PINK         24
1     536370     22727       ALARM CLOCK BAKELIKE RED          24
2     536370     22726       ALARM CLOCK BAKELIKE GREEN        12
3     536370     21724  PANDA AND BUNNIES STICKER SHEET        12
4     536370     21883                 STARS GIFT TAPE         24

     InvoiceDate  UnitPrice  CustomerID Country
0  12/1/10 8:45       3.75     12583.0  France
1  12/1/10 8:45       3.75     12583.0  France
2  12/1/10 8:45       3.75     12583.0  France
3  12/1/10 8:45       0.85     12583.0  France
4  12/1/10 8:45       0.65     12583.0  France
```




```
[0 0 0 ... 0 0 0]
```




```
Silhouette Score: 0.95
<ipython-input-6-a80cde99a5f2>:70: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legenc

  sns.countplot(x='Cluster', data=data, palette='Set2')
```