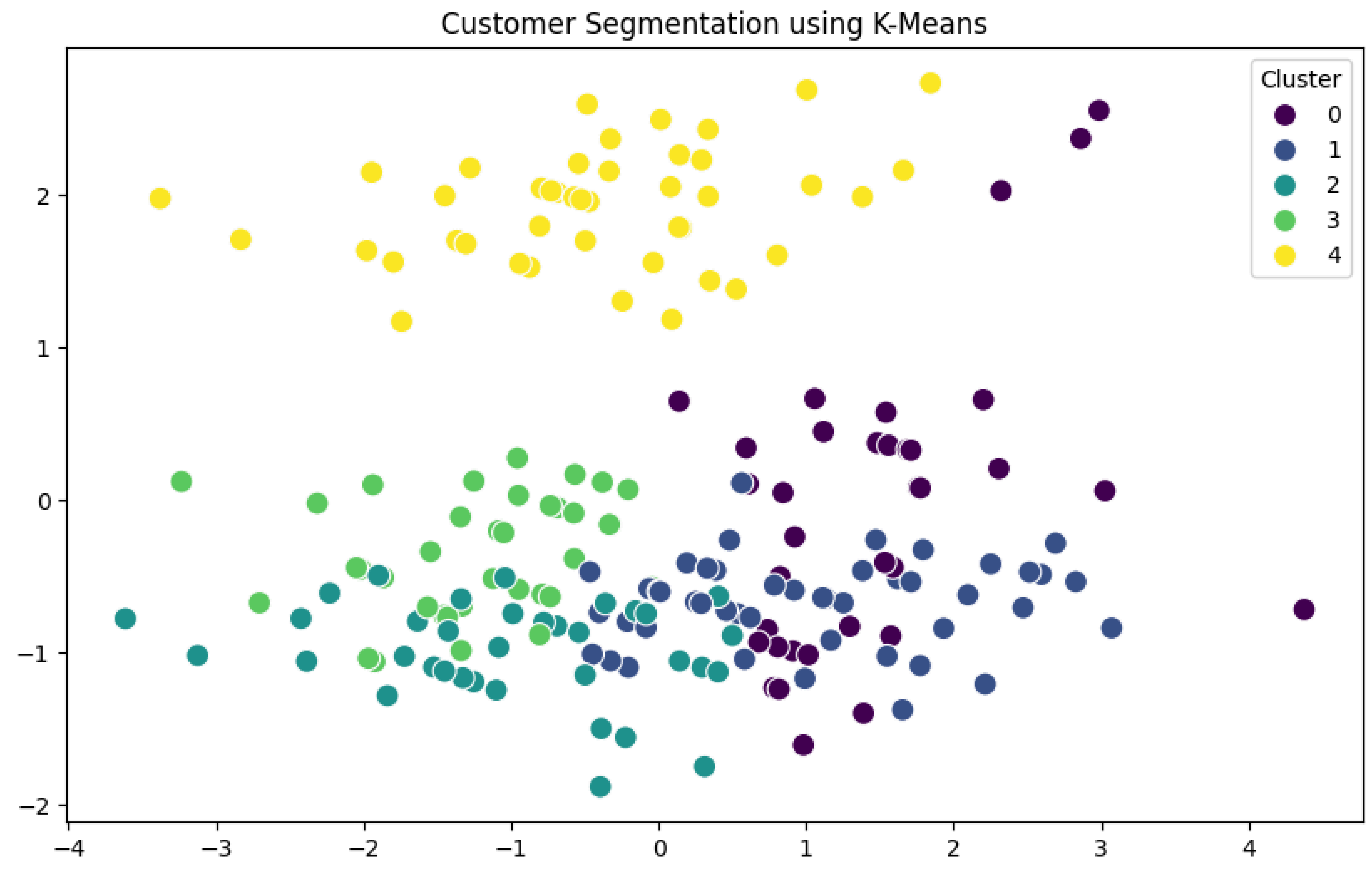
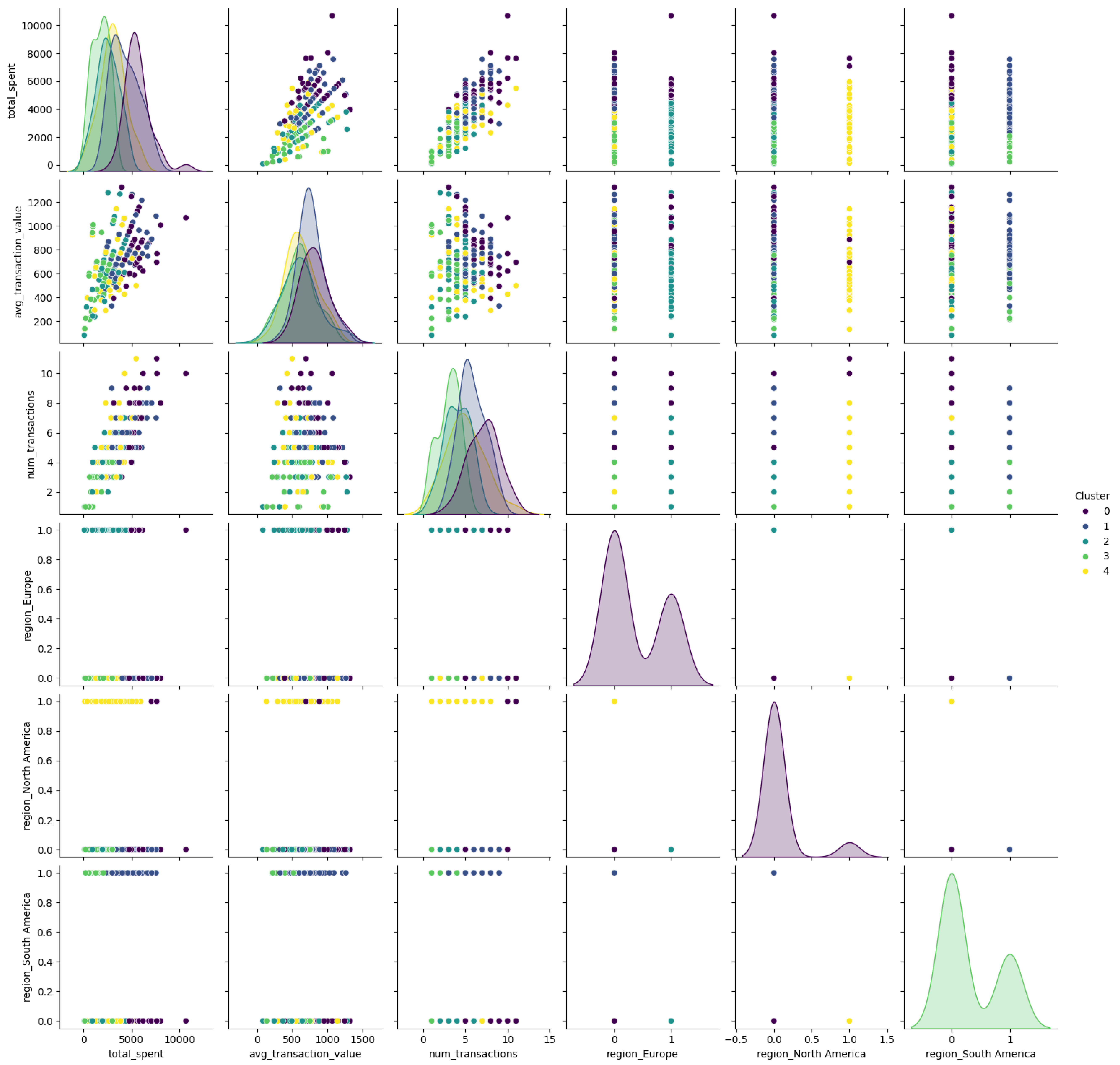
In [12]:

```python
# Step 1: Data Preprocessing
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
import seaborn as sns
import matplotlib.pyplot as plt

# Load the data
customers_df = pd.read_csv(r"C:/Users/Administrator/Downloads/Customers.csv")
products_df = pd.read_csv(r"C:/Users/Administrator/Downloads/Products.csv")
transactions_df = pd.read_csv(r"C:/Users/Administrator/Downloads/Transactions.csv")

# Merge the customer and transaction data
customer_transactions_df = pd.merge(transactions_df, customers_df, on='CustomerID')

# Feature Engineering: Aggregate data by CustomerID
customer_agg_df = customer_transactions_df.groupby('CustomerID').agg(
    total_spent=('TotalValue', 'sum'),
    avg_transaction_value=('TotalValue', 'mean'),
    num_transactions=('TotalValue', 'count'),
    region=('Region', 'first')
).reset_index()

# One-hot encoding for categorical features
customer_agg_df = pd.get_dummies(customer_agg_df, columns=['region'], drop_first=True)

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(customer_agg_df.drop('CustomerID', axis=1))

# Step 2: Clustering using K-Means
# Determine the optimal number of clusters (e.g., use the elbow method or silhouette score)
kmeans = KMeans(n_clusters=5, random_state=42)
customer_agg_df['Cluster'] = kmeans.fit_predict(scaled_data)

# Step 3: Evaluate Clustering with DB Index
db_index = davies_bouldin_score(scaled_data, customer_agg_df['Cluster'])
print(f'Davies-Bouldin Index: {db_index}')

# Step 4: Visualize the Clusters
# Use PCA or TSNE for dimensionality reduction (2D visualization)
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
pca_components = pca.fit_transform(scaled_data)

plt.figure(figsize=(10, 6))
sns.scatterplot(x=pca_components[:, 0], y=pca_components[:, 1], hue=customer_agg_df['Cluster'], palette='viridis', s=100)
plt.title('Customer Segmentation using K-Means')
plt.show()

# Optional: If you'd like to visualize the clusters more clearly
sns.pairplot(customer_agg_df, hue='Cluster', palette='viridis')
plt.show()

# Optional: Evaluate with other metrics such as Silhouette Score
from sklearn.metrics import silhouette_score
silhouette = silhouette_score(scaled_data, customer_agg_df['Cluster'])
print(f'Silhouette Score: {silhouette}')
Report for Clustering Results:

    Number of Clusters: 5 (chosen based on business logic or clustering metrics).
    DB Index Value: The DB Index will be printed, and lower values indicate better clustering.
    Silhouette Score: This score will be printed, providing an indication of how distinct the clusters are.
    Visual Representation: The scatter plot of the clustered customers and pair plot will show the clusters in 2D space.
```

```
Davies-Bouldin Index: 1.1236219126170808
```






```
Silhouette Score: 0.3303172676214734
```

In [ ]: