In [1]:

```
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 matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
# Load customer and transaction data
customers = pd.read_csv('Customers.csv')
transactions = pd.read_csv('Transactions.csv')

# Merge datasets on 'CustomerID'
data = pd.merge(transactions, customers, on='CustomerID', how='inner')
```

In [3]:

```
# Check for missing values
print(data.isnull().sum())

# Drop rows with missing values
data.dropna(inplace=True)
```

```
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
CustomerName       0
Region             0
SignupDate         0
dtype: int64
```

In [4]:

```
#Convert 'TransactionDate' to datetime
data['TransactionDate'] = pd.to_datetime(data['TransactionDate'])
```

In [5]:

```
# Define a reference date for recency calculation
reference_date = data['TransactionDate'].max() + pd.Timedelta(days=1)

# Calculate Recency, Frequency, and Monetary value for each customer
rfm = data.groupby('CustomerID').agg({
    'TransactionDate': lambda x: (reference_date - x.max()).days,
    'TransactionID': 'nunique',
    'Price': 'sum'
}).reset_index()

# Rename columns
rfm.columns = ['CustomerID', 'Recency', 'Frequency', 'Monetary']
```

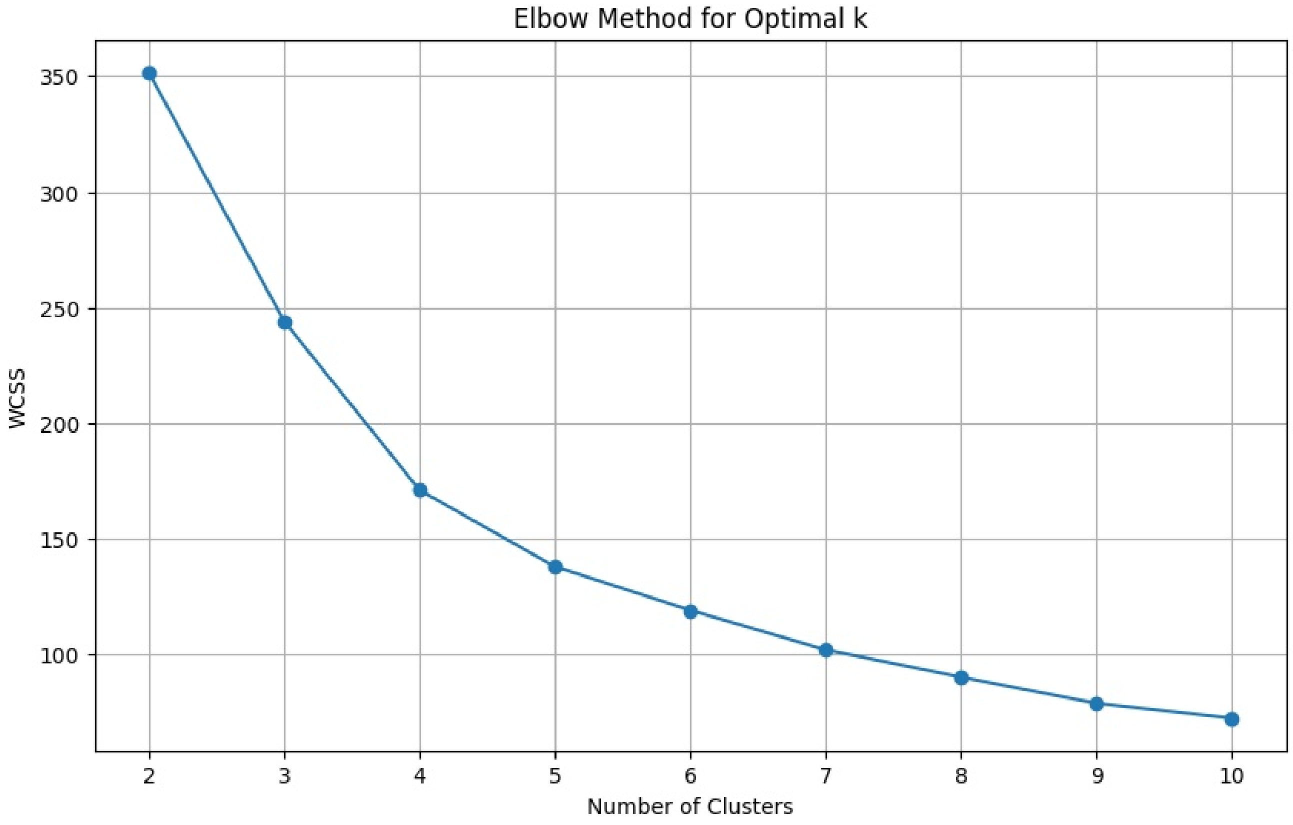
In [6]:

```
# Select features for clustering
features = rfm[['Recency', 'Frequency', 'Monetary']]

# Log transformation to reduce skewness
features_log = np.log1p(features)

# Standardize the features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features_log)
```

In [7]:

```
# Calculate Within-Cluster Sum of Squares (WCSS) for different cluster counts
wcss = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(features_scaled)
    wcss.append(kmeans.inertia_)

# Plot the Elbow Method graph
plt.figure(figsize=(10, 6))
plt.plot(range(2, 11), wcss, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()
```

```
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```



In [8]:

```python
# Choose the optimal number of clusters (e.g., k=4)
optimal_k = 4
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
rfm['Cluster'] = kmeans.fit_predict(features_scaled)
```

```
C:\Users\panmo\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: Future
Warning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
y to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

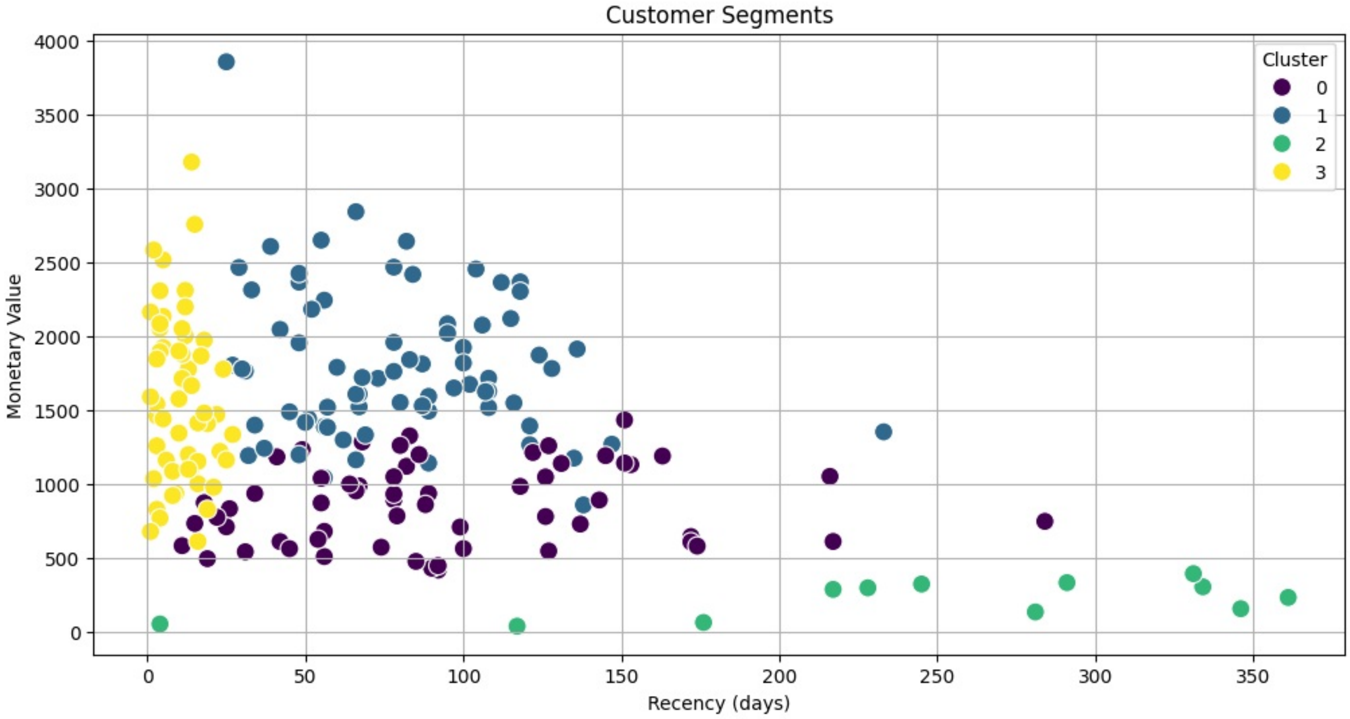
In [9]:

```python
# Calculate Davies-Bouldin Index
db_index = davies_bouldin_score(features_scaled, rfm['Cluster'])
print(f'Davies-Bouldin Index: {db_index:.2f}')
```

```
Davies-Bouldin Index: 0.85
```

In [10]:

```
# Plot clusters
plt.figure(figsize=(12, 6))
sns.scatterplot(data=rfm, x='Recency', y='Monetary', hue='Cluster', palette='viridis', s=100)
plt.title('Customer Segments')
plt.xlabel('Recency (days)')
plt.ylabel('Monetary Value')
plt.legend(title='Cluster')
plt.grid(True)
plt.show()
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js