In [1]:
```
import pandas as pd
import numpy as np
import datetime
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
```

In [2]:
```
sales = pd.read_csv("C:/Users/Pratik/Desktop/Internship/raw data/customer_data.csv"
```

In [3]:
```
print("Number of data points:", len(sales))

sales.head()
```

```
Number of data points: 2240
```

Out[3]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014

5 rows × 29 columns

Data Cleaning and Feature Extraction

In [4]:
```
sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

In [5]:

```
## removing the rows with missing income values

sales = sales.dropna()
print("Data Points after removing the missing value rows: ", len(sales))
```

```
Data Points after removing the missing value rows:  2216
```

In [12]:

```
sales["Dt_Customer"] = pd.to_datetime(sales["Dt_Customer"], dayfirst=True)
```

In [13]:

```
sales["Dt_Customer"] = pd.to_datetime(sales["Dt_Customer"], format="%d-%m-%Y")
```

In [14]:

```
sales["Dt_Customer"] = pd.to_datetime(sales["Dt_Customer"], errors='coerce', dayfir
print(sales[sales["Dt_Customer"].isna()])  # Check invalid records
```

```
Empty DataFrame
Columns: [ID, Year_Birth, Education, Marital_Status, Income, Kidhome, Teenhome, Dt_C
ustomer, Recency, MntWines, MntFruits, MntMeatProducts, MntFishProducts, MntSweetPro
ducts, MntGoldProds, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumSto
rePurchases, NumWebVisitsMonth, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, AcceptedCm
p1, AcceptedCmp2, Complain, Z_CostContact, Z_Revenue, Response]
Index: []

[0 rows x 29 columns]
```

In [15]:
```
## creating a feature out of 'Dt_Customer' at shows number of days
sales["Dt_Customer"] = pd.to_datetime(sales["Dt_Customer"])
dates = []
for i in sales["Dt_Customer"]:
    i = i.date()
    dates.append(i)
#Dates of the newest and oldest recorded customer
print("The newest customer's enrolment date in therecords:",max(dates))
print("The oldest customer's enrolment date in the records:",min(dates))
```

```
The newest customer's enrolment date in therecords: 2014-06-29
The oldest customer's enrolment date in the records: 2012-07-30
```

In [18]:
```
#Created a feature "Customer_For"
days = []
d1 = max(dates) #taking it to be the newest customer
for i in dates:
    delta = d1 - i
    days.append(delta)
sales["Customer_For"] = days
sales["Customer_For"] = pd.to_numeric(sales["Customer_For"], errors="coerce")
```

In [21]:
```
#Feature Engineering
#Age of customer today
sales["Age"] = 2021-sales["Year_Birth"]
```

In [22]:
```
sales["Spent"] = (sales["MntWines"] + sales["MntFruits"] + sales["MntMeatProducts"]
                  sales["MntFishProducts"] + sales["MntSweetProducts"] + sales["Mnt
```

In [23]:
```
#Dropping some of the redundant features
to_drop = ["Marital_Status", "Dt_Customer", "Z_CostContact", "Z_Revenue", "Year_Bir
sales = sales.drop(to_drop, axis=1)
```

In [24]:
```
sales.describe()
```

Out[24]:

	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	Mı
count	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	2216.000000	
mean	52247.251354	0.441787	0.505415	49.012635	305.091606	26.356047	
std	25173.076661	0.536896	0.544181	28.948352	337.327920	39.793917	
min	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	35303.000000	0.000000	0.000000	24.000000	24.000000	2.000000	
50%	51381.500000	0.000000	0.000000	49.000000	174.500000	8.000000	
75%	68522.000000	1.000000	1.000000	74.000000	505.000000	33.000000	
max	666666.000000	2.000000	2.000000	99.000000	1493.000000	199.000000	

8 rows × 25 columns

Data Preprocessing

In [40]:

```
from sklearn.preprocessing import StandardScaler


features = sales[['Age', 'Income', 'Spent']]

scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)


scaled_df = pd.DataFrame(scaled_features, columns=features.columns)
print("\nScaled features:")
print(scaled_df.head())
```

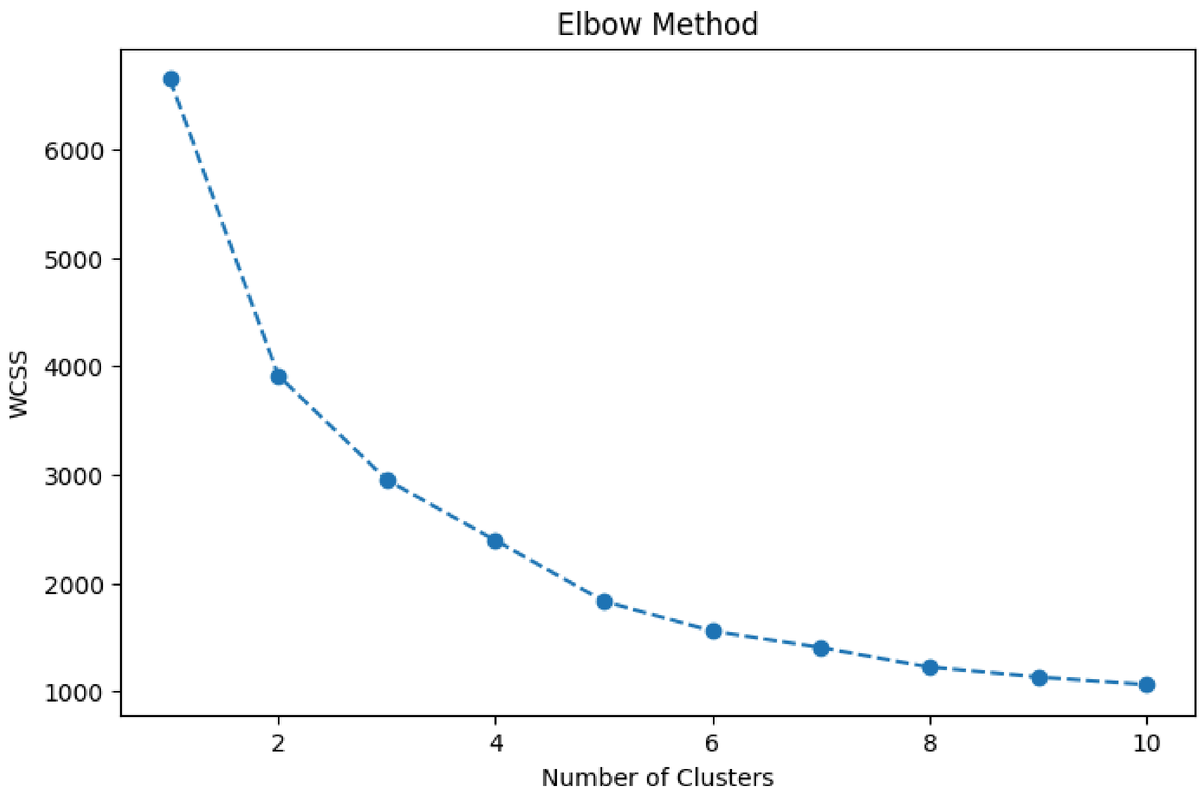
```
Scaled features:
        Age    Income     Spent
0  0.986443  0.234063  1.675488
1  1.236801 -0.234559 -0.962358
2  0.318822  0.769478  0.280250
3 -1.266777 -1.017239 -0.919224
4 -1.016420  0.240221 -0.307044
```

Clustering

In [39]:

```
## clustering
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Elbow Method
wcss = []
for i in range(1, 11):
```

```
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(scaled_features)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```



In [41]:
```
## Applying k means clustering
# Optimal number of clusters from the Elbow Method (choose based on the plot)
optimal_clusters = 4  # Example


kmeans = KMeans(n_clusters=optimal_clusters, init='k-means++', random_state=42)
sales['Cluster'] = kmeans.fit_predict(scaled_features)

print("\nClustered dataset:")
print(sales.head())
```

```
Clustered dataset:
    Education   Income  Kidhome  Teenhome  Recency  MntWines  MntFruits  \
0  Graduation  58138.0        0         0       58       635         88   
1  Graduation  46344.0        1         1       38        11          1   
2  Graduation  71613.0        0         0       26       426         49   
3  Graduation  26646.0        1         0       26        11          4   
4         PhD  58293.0        1         0       94       173         43   

   MntMeatProducts  MntFishProducts  MntSweetProducts  ...  AcceptedCmp4  \
0              546              172                88  ...             0   
1                6                2                 1  ...             0   
2              127              111                21  ...             0   
3               20               10                 3  ...             0   
4              118               46                27  ...             0   

   AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  Response  \
0             0             0             0         0         1   
1             0             0             0         0         0   
2             0             0             0         0         0   
3             0             0             0         0         0   
4             0             0             0         0         0   

        Customer_For  Age  Spent  Cluster  
0  57283200000000000   64   1617        1  
1   9763200000000000   67     27        2  
2  26956800000000000   56    776        1  
3  12009600000000000   37     53        0  
4  13910400000000000   40    422        0  

[5 rows x 27 columns]
```

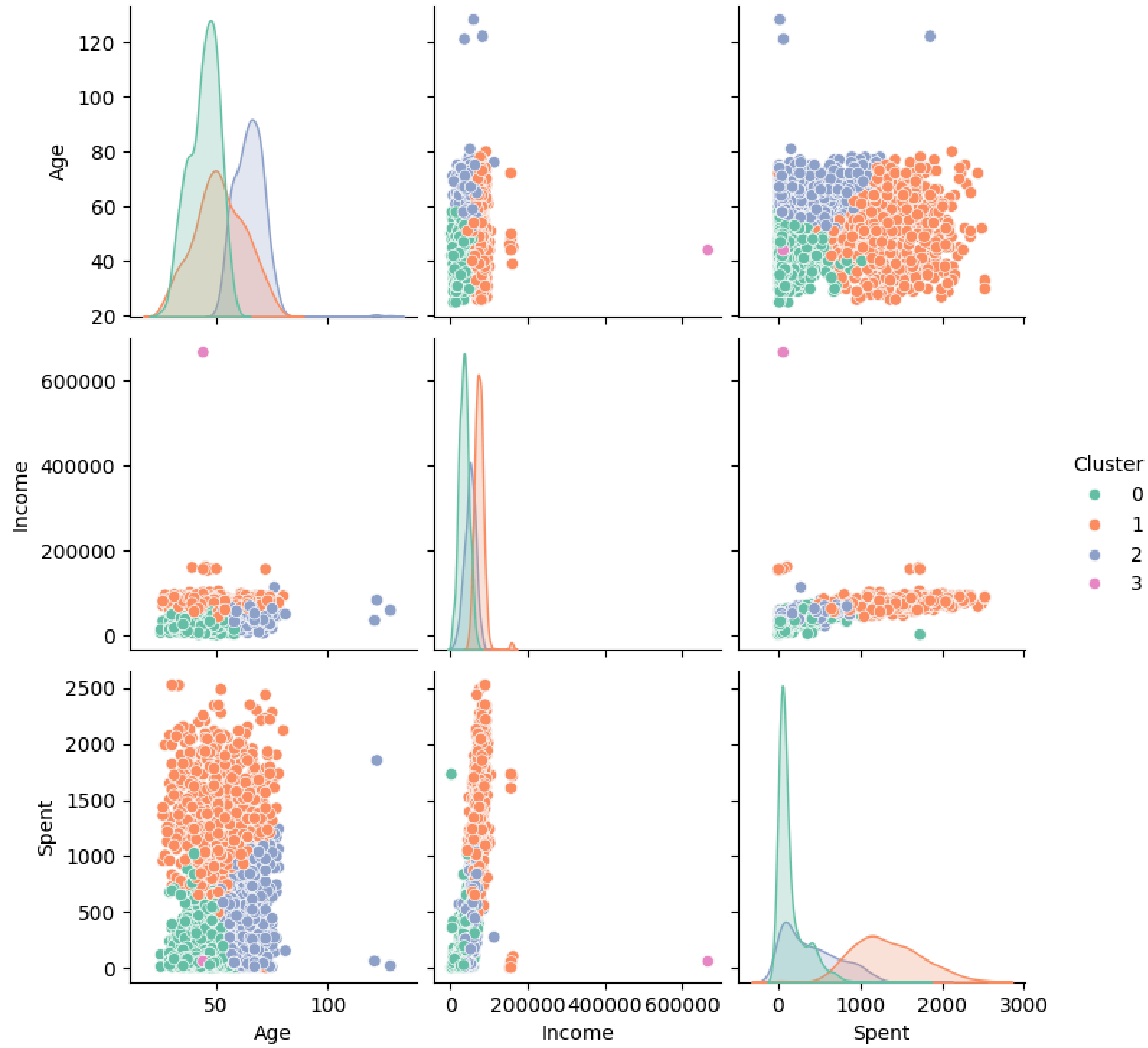
Visualization

In [46]:

```python
## 2D Scatter Plot

from sklearn.decomposition import PCA
import seaborn as sns

# Reduce dimensions using PCA
pca = PCA(n_components=2)
pca_features = pca.fit_transform(scaled_features)

# Create a DataFrame with PCA components and clusters
pca_df = pd.DataFrame(pca_features, columns=['PCA1', 'PCA2'])
pca_df['Cluster'] = sales['Cluster']

# Scatter plot
plt.figure(figsize=(8, 5))
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', data=pca_df, palette='Set1', s=1
plt.title('Customer Clusters (2D PCA)')
plt.show()
```

In [ ]:
```
## pair plots
```

In [35]:
```
sns.pairplot(sales[['Age', 'Income', 'Spent', 'Cluster']], hue='Cluster', palette='
plt.show()
```



In [36]:
```
## Centoid Visualization
centroids = kmeans.cluster_centers_
centroids_df = pd.DataFrame(centroids, columns=features.columns)

print("\nCentroids of clusters:")
print(centroids_df)
```

```
Centroids of clusters:
        Age     Income     Spent
0 -0.664558  -0.696772 -0.754073
1 -0.064854   0.875130  1.193879
2  1.050829  -0.076586 -0.335088
3 -0.682609  24.413282 -0.904293
```

In [47]:
```
# Analyze clusters for actionable insights
for cluster in sales['Cluster'].unique():
    cluster_data = sales[sales['Cluster'] == cluster]
    print(f"\nCluster {cluster} Analysis:")
    print(cluster_data.describe())

print("\nRecommendations:")
print("- Target customers in high-income, high-spending clusters for premium produc
```

```
print("- Introduce loyalty programs for high-spending customers.")
print("- Design tailored promotions for age-specific or income-specific segments.")
```

```
Cluster 1 Analysis:
              Income     Kidhome    Teenhome     Recency     MntWines  \
count     728.000000  728.000000  728.000000  728.000000   728.000000
mean    74271.984890    0.089286    0.395604   49.696429   661.876374
std     13150.391059    0.290132    0.529807   29.026476   310.165477
min     44802.000000    0.000000    0.000000    0.000000     1.000000
25%     66326.250000    0.000000    0.000000   25.000000   423.000000
50%     73452.000000    0.000000    0.000000   51.500000   626.000000
75%     80881.500000    0.000000    1.000000   74.000000   896.250000
max    162397.000000    2.000000    2.000000   99.000000  1493.000000

        MntFruits  MntMeatProducts  MntFishProducts  MntSweetProducts  \
count  728.000000       728.000000       728.000000        728.000000
mean    58.048077       390.817308        82.188187         59.170330
std     49.773536       248.609566        67.634844         50.875429
min      0.000000         1.000000         0.000000          0.000000
25%     20.000000       184.000000        28.000000         19.000000
50%     43.000000       352.000000        64.000000         43.000000
75%     86.000000       541.250000       127.000000         92.000000
max    199.000000      1725.000000       259.000000        198.000000

       MntGoldProds  ...  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  \
count    728.000000  ...    728.000000    728.000000    728.000000
mean      74.603022  ...      0.145604      0.214286      0.171703
std       59.844207  ...      0.352952      0.410608      0.377382
min        0.000000  ...      0.000000      0.000000      0.000000
25%       29.750000  ...      0.000000      0.000000      0.000000
50%       54.000000  ...      0.000000      0.000000      0.000000
75%      108.000000  ...      0.000000      0.000000      0.000000
max      249.000000  ...      1.000000      1.000000      1.000000

       AcceptedCmp2    Complain    Response  Customer_For         Age  \
count    728.000000  728.000000  728.000000  7.280000e+02  728.000000
mean       0.034341    0.004121    0.251374  3.275628e+16   51.402473
std        0.182228    0.064106    0.434101  1.786404e+16   11.341096
min        0.000000    0.000000    0.000000  8.640000e+13   26.000000
25%        0.000000    0.000000    0.000000  1.678320e+16   44.000000
50%        0.000000    0.000000    0.000000  3.490560e+16   51.000000
75%        0.000000    0.000000    1.000000  4.918320e+16   60.000000
max        1.000000    1.000000    1.000000  6.022080e+16   80.000000

             Spent  Cluster
count   728.000000    728.0
mean   1326.703297      1.0
std     414.958426      0.0
min       6.000000      1.0
25%    1026.250000      1.0
50%    1284.500000      1.0
75%    1609.000000      1.0
max    2525.000000      1.0

[8 rows x 26 columns]

Cluster 2 Analysis:
              Income     Kidhome    Teenhome     Recency     MntWines  \
count     604.000000  604.000000  604.000000  604.000000   604.000000
```

```
mean    50319.774834    0.344371    0.847682   49.223510   223.642384
std     14567.776051    0.528415    0.467850   29.001425   209.520746
min      5648.000000    0.000000    0.000000    0.000000     0.000000
25%     40754.250000    0.000000    1.000000   24.000000    45.000000
50%     50884.000000    0.000000    1.000000   51.000000   172.000000
75%     60544.000000    1.000000    1.000000   74.000000   356.750000
max    113734.000000    2.000000    2.000000   99.000000  1099.000000

        MntFruits  MntMeatProducts  MntFishProducts  MntSweetProducts  \
count  604.000000       604.000000        604.00000        604.000000
mean    15.546358        88.228477         22.56457         16.922185
std     26.657079       106.129747         35.07411         30.163219
min      0.000000         1.000000          0.00000          0.000000
25%      1.000000        16.000000          2.00000          0.000000
50%      5.000000        49.500000          7.50000          5.000000
75%     17.000000       115.750000         28.00000         19.000000
max    178.000000       818.000000        199.00000        262.000000

       MntGoldProds  ...  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  \
count    604.000000  ...    604.000000    604.000000    604.000000
mean      38.192053  ...      0.076159      0.009934      0.021523
std       45.375780  ...      0.265472      0.099254      0.145241
min        0.000000  ...      0.000000      0.000000      0.000000
25%        7.000000  ...      0.000000      0.000000      0.000000
50%       22.500000  ...      0.000000      0.000000      0.000000
75%       48.000000  ...      0.000000      0.000000      0.000000
max      229.000000  ...      1.000000      1.000000      1.000000

       AcceptedCmp2    Complain    Response  Customer_For         Age  \
count    604.000000  604.000000  604.000000  6.040000e+02  604.000000
mean       0.008278    0.016556    0.086093  2.930033e+16   64.771523
std        0.090682    0.127707    0.280733  1.702079e+16    7.328410
min        0.000000    0.000000    0.000000  0.000000e+00   51.000000
25%        0.000000    0.000000    0.000000  1.466640e+16   60.000000
50%        0.000000    0.000000    0.000000  2.959200e+16   65.000000
75%        0.000000    0.000000    0.000000  4.328640e+16   69.000000
max        1.000000    1.000000    1.000000  6.030720e+16  128.000000

             Spent  Cluster
count   604.000000    604.0
mean    405.096026      2.0
std     334.790671      0.0
min       9.000000      2.0
25%      93.750000      2.0
50%     319.500000      2.0
75%     637.000000      2.0
max    1853.000000      2.0

[8 rows x 26 columns]

Cluster 0 Analysis:
             Income     Kidhome    Teenhome     Recency    MntWines  \
count    883.000000  883.000000  883.000000  883.000000  883.000000
mean   34711.318233    0.798414    0.362401   48.334088   66.985277
std    12843.434724    0.476331    0.501734   28.868268   99.930729
min     1730.000000    0.000000    0.000000    0.000000    0.000000
```

```
25%    25293.000000    1.000000    0.000000   24.000000    8.000000
50%    34600.000000    1.000000    0.000000   47.000000   23.000000
75%    42801.000000    1.000000    1.000000   74.000000   80.000000
max    73395.000000    2.000000    2.000000   99.000000  728.000000

        MntFruits  MntMeatProducts  MntFishProducts  MntSweetProducts  \
count  883.000000       883.000000       883.000000        883.000000
mean     7.635334        36.511891        11.251416          7.472254
std     13.744864        70.808392        20.616358         13.214868
min      0.000000         0.000000         0.000000          0.000000
25%      1.000000         8.000000         2.000000          1.000000
50%      3.000000        17.000000         4.000000          3.000000
75%      7.000000        46.000000        12.000000          8.000000
max    122.000000      1725.000000       208.000000        129.000000

       MntGoldProds  ...  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  \
count    883.000000  ...    883.000000         883.0    883.000000
mean      22.690827  ...      0.013590           0.0      0.004530
std       33.630994  ...      0.115847           0.0      0.067191
min        0.000000  ...      0.000000           0.0      0.000000
25%        5.000000  ...      0.000000           0.0      0.000000
50%       12.000000  ...      0.000000           0.0      0.000000
75%       26.000000  ...      0.000000           0.0      0.000000
max      321.000000  ...      1.000000           0.0      1.000000

       AcceptedCmp2    Complain    Response  Customer_For         Age  \
count         883.0  883.000000  883.000000  8.830000e+02  883.000000
mean            0.0    0.009060    0.110985  2.956759e+16   44.216308
std             0.0    0.094806    0.314292  1.735155e+16    6.737017
min             0.0    0.000000    0.000000  1.728000e+14   25.000000
25%             0.0    0.000000    0.000000  1.442880e+16   39.000000
50%             0.0    0.000000    0.000000  2.859840e+16   45.000000
75%             0.0    0.000000    0.000000  4.458240e+16   49.000000
max             0.0    1.000000    1.000000  6.039360e+16   60.000000

             Spent  Cluster
count   883.000000    883.0
mean    152.546999      0.0
std     178.146303      0.0
min       5.000000      0.0
25%      41.000000      0.0
50%      72.000000      0.0
75%     198.500000      0.0
max    1730.000000      0.0

[8 rows x 26 columns]

Cluster 3 Analysis:
         Income  Kidhome  Teenhome  Recency  MntWines  MntFruits  \
count       1.0      1.0       1.0      1.0       1.0        1.0
mean   666666.0      1.0       0.0     23.0       9.0       14.0
std         NaN      NaN       NaN      NaN       NaN        NaN
min    666666.0      1.0       0.0     23.0       9.0       14.0
25%    666666.0      1.0       0.0     23.0       9.0       14.0
50%    666666.0      1.0       0.0     23.0       9.0       14.0
75%    666666.0      1.0       0.0     23.0       9.0       14.0
```

```
max    666666.0      1.0       0.0     23.0       9.0       14.0

       MntMeatProducts  MntFishProducts  MntSweetProducts  MntGoldProds  ...  \
count              1.0              1.0               1.0           1.0  ...
mean              18.0              8.0               1.0          12.0  ...
std                NaN              NaN               NaN           NaN  ...
min               18.0              8.0               1.0          12.0  ...
25%               18.0              8.0               1.0          12.0  ...
50%               18.0              8.0               1.0          12.0  ...
75%               18.0              8.0               1.0          12.0  ...
max               18.0              8.0               1.0          12.0  ...

       AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
count           1.0           1.0           1.0           1.0       1.0
mean            0.0           0.0           0.0           0.0       0.0
std             NaN           NaN           NaN           NaN       NaN
min             0.0           0.0           0.0           0.0       0.0
25%             0.0           0.0           0.0           0.0       0.0
50%             0.0           0.0           0.0           0.0       0.0
75%             0.0           0.0           0.0           0.0       0.0
max             0.0           0.0           0.0           0.0       0.0

       Response  Customer_For   Age  Spent  Cluster
count       1.0  1.000000e+00   1.0    1.0      1.0
mean        0.0  3.386880e+16  44.0   62.0      3.0
std         NaN           NaN   NaN    NaN      NaN
min         0.0  3.386880e+16  44.0   62.0      3.0
25%         0.0  3.386880e+16  44.0   62.0      3.0
50%         0.0  3.386880e+16  44.0   62.0      3.0
75%         0.0  3.386880e+16  44.0   62.0      3.0
max         0.0  3.386880e+16  44.0   62.0      3.0

[8 rows x 26 columns]

Recommendations:
- Target customers in high-income, high-spending clusters for premium products.
- Introduce loyalty programs for high-spending customers.
- Design tailored promotions for age-specific or income-specific segments.
```