In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.manifold import TSNE
import umap
from sklearn.metrics import silhouette_score, calinski_harabasz_score, davies_bould
import warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```
plt.style.use('default')
sns.set_palette("husl")
```

In [ ]:
```
original_data = pd.read_csv(r"data\01_marketing_campaign.csv", sep='\t')
cleaned_data = pd.read_csv(r"data\02_removed_outliers_redundant.csv", index_col=0)
scaled_data = pd.read_csv(r"data\03_scaled_proprocessed_marketing_campaign.csv", in

print(f"Original data shape: {original_data.shape}")
print(f"Cleaned data shape: {cleaned_data.shape}")
print(f"Scaled data shape: {scaled_data.shape}")
print()
```

```
Original data shape: (2240, 29)
Cleaned data shape: (2184, 17)
Scaled data shape: (2184, 17)
```

In [6]:
```
cleaned_data.head()
```

Out[6]:

	Education	Marital_Status	Income	Recency	Response	Age	Customer_Since	Total_Spe
0	0	0	58138.0	58	1	68	663	16
1	0	0	46344.0	38	0	71	113	2
2	0	1	71613.0	26	0	60	312	77
3	0	1	26646.0	26	0	41	139	5
4	1	1	58293.0	94	0	44	161	42

In [5]:
```
scaled_data.head()
```

Out[5]:

	Education	Marital_Status	Income	Recency	Response	Age	Customer_Since	T
0	-0.938689	-1.351057	0.323276	0.309449	2.391652	1.020547	1.528805	
1	-0.938689	-1.351057	-0.252104	-0.382368	-0.418121	1.278260	-1.187852	
2	-0.938689	0.740161	0.980665	-0.797458	-0.418121	0.333312	-0.204916	
3	-0.938689	0.740161	-1.213087	-0.797458	-0.418121	-1.298871	-1.059428	
4	1.065316	0.740161	0.330838	1.554719	-0.418121	-1.041158	-0.950762	

In [ ]:

```
# final features
print(scaled_data.columns.tolist())
```

```
['Education', 'Marital_Status', 'Income', 'Recency', 'Response', 'Age', 'Customer_Si
nce', 'Total_Spent', 'RatioWines', 'RatioFruits', 'RatioMeatProducts', 'RatioFishPro
ducts', 'RatioSweetProducts', 'RatioGoldProds', 'Total_Accepted_Campaign', 'Total_Pu
rchase', 'Total_Web_Engagement']
```

In [ ]:

```
print(cleaned_data.describe())
```

	Education	Marital_Status	Income	Recency	Response
count	2184.000000	2184.000000	2184.000000	2184.000000	2184.000000
mean	0.468407	0.646062	51511.571886	49.054029	0.148810
std	0.499115	0.478300	20502.451226	28.916006	0.355982
min	0.000000	0.000000	1730.000000	0.000000	0.000000
25%	0.000000	0.000000	35191.500000	24.000000	0.000000
50%	0.000000	1.000000	51144.500000	49.000000	0.000000
75%	1.000000	1.000000	67956.250000	74.000000	0.000000
max	1.000000	1.000000	105471.000000	99.000000	1.000000

	Age	Customer_Since	Total_Spent	RatioWines	RatioFruits
count	2184.000000	2184.000000	2184.000000	2184.000000	2184.000000
mean	56.119963	353.486264	598.756410	0.459924	0.049712
std	11.643517	202.501065	593.060286	0.228072	0.055931
min	29.000000	0.000000	5.000000	0.000000	0.000000
25%	48.000000	180.000000	68.750000	0.290575	0.008924
50%	55.000000	355.000000	394.000000	0.458445	0.030027
75%	66.000000	529.000000	1035.000000	0.641425	0.070915
max	85.000000	699.000000	2524.000000	0.963303	0.445545

	RatioMeatProducts	RatioFishProducts	RatioSweetProducts
count	2184.000000	2184.000000	2184.000000
mean	0.247795	0.071978	0.050515
std	0.121710	0.078295	0.058039
min	0.000000	0.000000	0.000000
25%	0.156217	0.012578	0.008629
50%	0.232882	0.048342	0.033475
75%	0.327472	0.105263	0.070411
max	0.749084	0.590909	0.600000

	RatioGoldProds	Total_Accepted_Campaign	Total_Purchase
count	2184.000000	2184.000000	2184.000000
mean	0.120077	0.295330	21.122253
std	0.107150	0.675639	7.126704
min	0.000000	0.000000	7.000000
25%	0.038205	0.000000	15.000000
50%	0.086294	0.000000	20.000000
75%	0.171192	0.000000	26.000000
max	0.702413	4.000000	42.000000

	Total_Web_Engagement
count	2184.000000
mean	9.407967
std	3.455728
min	1.000000
25%	7.000000
50%	9.000000
75%	11.000000
max	20.000000

Demographics:

Average age: 56 years (mature customer base)

Education: 47% have higher education

Marital Status: 65% are coupled

Financial Behavior:

Average income: $51,512 (middle-class focus)

Average spending: $599 (11.6% of income spent)

High variation: Income std = $20,502 (diverse economic segments)

Engagement Patterns:

Purchase frequency: 21 transactions average

Web engagement: 9.4 average interactions

In [ ]:
```
print(cleaned_data.isnull().sum())
```

```
Education                  0
Marital_Status             0
Income                     0
Recency                    0
Response                   0
Age                        0
Customer_Since             0
Total_Spent                0
RatioWines                 0
RatioFruits                0
RatioMeatProducts          0
RatioFishProducts          0
RatioSweetProducts         0
RatioGoldProds             0
Total_Accepted_Campaign    0
Total_Purchase             0
Total_Web_Engagement       0
dtype: int64
```

In [9]:
```
# Examine the correlation matrix
correlation_matrix = cleaned_data.corr()
```

In [12]:
```
# Find highly correlated features (>0.5 or <-0.5)
high_corr_pairs = []
for i in range(len(correlation_matrix.columns)):
    for j in range(i+1, len(correlation_matrix.columns)):
        corr_val = correlation_matrix.iloc[i, j]
        if abs(corr_val) > 0.5:
            high_corr_pairs.append((correlation_matrix.columns[i],
                                  correlation_matrix.columns[j],
                                  corr_val))

print("Highly correlated feature pairs (|correlation| > 0.5):")
for pair in high_corr_pairs:
```

```
    print(f"{pair[0]} - {pair[1]}: {pair[2]:.3f}")
print()
```

```
Highly correlated feature pairs (|correlation| > 0.5):
Income - Total_Spent: 0.832
Income - RatioGoldProds: -0.545
Income - Total_Purchase: 0.506
Total_Spent - Total_Purchase: 0.594
RatioWines - RatioFruits: -0.585
RatioWines - RatioFishProducts: -0.618
RatioWines - RatioSweetProducts: -0.575
RatioWines - RatioGoldProds: -0.542
Total_Purchase - Total_Web_Engagement: 0.669
```

choose some key feature based on heatmap

Income , total spent = high corr

total_purchase , Total_Web_Engagement = good corr

choosing age as another feature, becasue it has sufficient variability and marketing significance.

In [21]:

```
# Examine data distributions
# Check for skewness in key variables
key_vars = ['Income', 'Total_Spent', 'Age', 'Total_Purchase', 'Total_Web_Engagement
for var in key_vars:
    if var in cleaned_data.columns:
        skewness = cleaned_data[var].skew()
        print(f"{var} skewness: {skewness:.3f}")
print()
```

```
Income skewness: 0.019
Total_Spent skewness: 0.845
Age skewness: 0.092
Total_Purchase skewness: 0.476
Total_Web_Engagement skewness: 0.485
```

Income skewness: 0.019 # Nearly normal

Total_Spent skewness: 0.845 # Right-skewed

Age skewness: 0.092 # Nearly normal

Total_Purchase skewness: 0.476 # Moderately right-skewed

Total_Web_Engagement skewness: 0.485 # Moderately right-skewed

Income is well-distributed - good feature for clustering

Spending is right-skewed - suggests a small group of high spenders

Engagement metrics are skewed - indicates power users vs. casual users

In [28]:
```
final_data_sumry = {
    'original_shape': original_data.shape,
    'cleaned_shape': cleaned_data.shape,
    'scaled_shape': scaled_data.shape,
    'final_features': scaled_data.columns.tolist(),
    'high_correlations': high_corr_pairs,
    'missing_values': cleaned_data.isnull().sum().sum()
}
```

In [29]:
```
final_data_sumry
```

Out[29]:
```
{'original_shape': (2240, 29),
 'cleaned_shape': (2184, 17),
 'scaled_shape': (2184, 17),
 'final_features': ['Education',
  'Marital_Status',
  'Income',
  'Recency',
  'Response',
  'Age',
  'Customer_Since',
  'Total_Spent',
  'RatioWines',
  'RatioFruits',
  'RatioMeatProducts',
  'RatioFishProducts',
  'RatioSweetProducts',
  'RatioGoldProds',
  'Total_Accepted_Campaign',
  'Total_Purchase',
  'Total_Web_Engagement'],
 'high_correlations': [('Income', 'Total_Spent', 0.8321899203330584)],
 'missing_values': 0}
```

only Income and Total Spent are well-distributedwith high correlation

I see that features like Income and Total Spent are highly correlated, and the data is well-prepared with no missing values. Next, I'll focus on exploring advanced clustering and dimensionality reduction methods, such as t-SNE or UMAP, to improve segmentation clarity beyond the initial PCA-based clusters. This will help discover more distinct customer groups for targeted marketing.