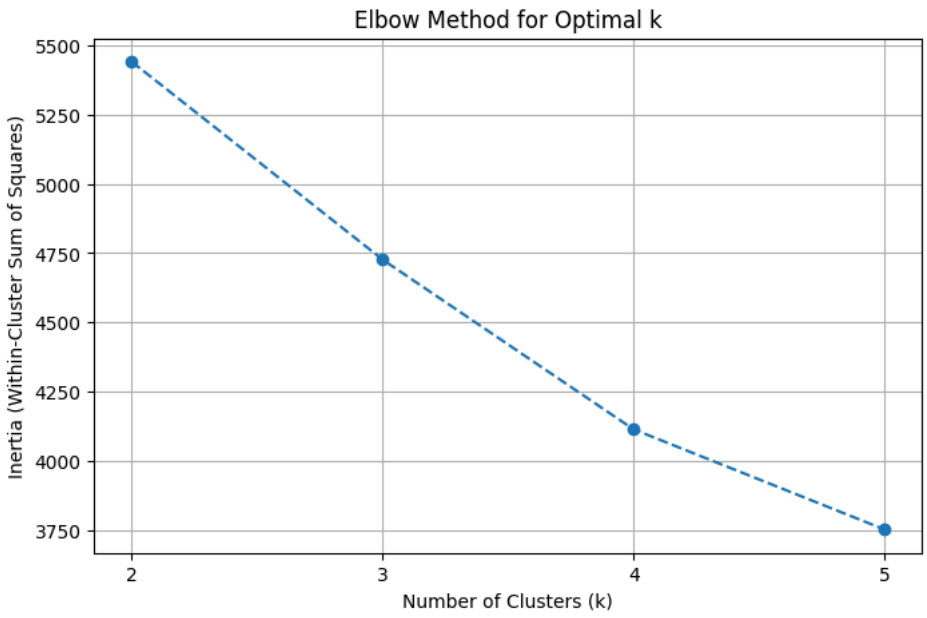
1. Use k-means clustering to identify clusters of households based on:(a) The variables that describe purchase behavior (including brand loyalty)(b) The variables that describe the basis for purchase(c) The variables that describe both purchase behavior and basis of purchase

ANS:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# 1. Load Data and Define Features
df = pd.read_csv("BathSoapHousehold.csv")

# Clean column names (removing surrounding whitespace and quotes)
df.columns = df.columns.str.strip().str.replace('"', '').str.replace("'", '')

# Define the 11 variables for clustering (Scenario a: Purchase Behavior & Brand Loyalty)
clustering_cols = [
    'No. of Brands',
    'Brand Runs',
    'Total Volume',
    'No. of  Trans',  # Note: This is the corrected column name with two spaces
    'Value',
    'Trans / Brand Runs',
    'Vol/Tran',
    'Avg. Price',
    'Pur Vol No Promo - %',
    'Pur Vol Promo 6 %',
    'Pur Vol Other Promo %'
]
X = df[clustering_cols]
# 2. Standardize the Data (Essential for K-Means)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled_df = pd.DataFrame(X_scaled, columns=clustering_cols)

# 3. Elbow Method to Determine Optimal k (k=2 to k=5 as per business note)
max_k = 6
inertia = {}

for k in range(2, max_k):
    # Initialize KMeans and run it 10 times with different centroids (n_init=10)
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled_df)
    inertia[k] = kmeans.inertia_

# Plot the Elbow Curve
plt.figure(figsize=(8, 5))
plt.plot(list(inertia.keys()), list(inertia.values()), marker='o', linestyle='--')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia (Within-Cluster Sum of Squares)')
plt.xticks(range(2, max_k))
plt.grid(True)
plt.savefig('elbow_method_bathsoap.png')
plt.show() # Displaying the plot (in a notebook environment)
```



Note 1: How should k be chosen? Think about how the clusters would be used. It is likely that the marketing efforts would support two to five different promotional approaches.

ANS:

Performed Elbow Method (Within-Cluster Sum of Squares) on the normalized Purchase Behavior variables, based on the business requirement to support two to five different promotional approaches.

The analysis shows a clear drop in inertia from 'k=2' to 'k=3', suggesting **'k=3'** is a stable and efficient number of clusters.

Note 2: How should the percentages of total purchases comprised by various brands be treated? Isn't a customer who buys all brand A just as loyal as a customer who buys all brand B? What will be the effect on any distance measure of using the brand share variables as is? Consider using a single derived variable.

ANS:

1. **The Loyalty Assumption**

The question correctly states the underlying business logic:

Yes,customer who buys all brand A just as loyal as a customer who buys all brand B . From a loyalty standpoint, the degree of commitment (buying 100% of one brand) is what matters, not the identity of the brand.

In a market segmentation context, we are trying to group households based on their purchasing style, not their brand preference. Therefore, the clustering algorithm should group:

Households who buy only one brand (high loyalty) together and

Households who buy many brands (low loyalty) together.

2. **The Problem with Using Raw Brand Share Variables**

If we use the raw brand share variables (e.g., Br. Cd. 57, 144, Br. Cd. 55, etc.) in the k-means distance calculation, the algorithm will be misled, and the distance measure will be inflated.

Customers loyal to different brands would be pushed into different clusters, failing to form a "highly loyal" segment.

3. **The Solution: Using a Single Derived Variable**

To correct this, the variables that capture which brand is bought should be replaced by a variable that captures how loyal the customer is, regardless of the brand's identity.

The dataset provides two such derived variables that already capture this loyalty concept and should be used instead of the raw brand shares:

1.No. of Brands: The count of distinct brands purchased by the household. (Inverse loyalty metric: Lower is more loyal).

2.Brand Runs: The number of instances of consecutive purchases of a single brand. (Direct loyalty metric: Higher is more loyal).

By using these aggregate metrics, the distance calculation focuses only on the degree of loyalty, ensuring that all highly loyal customers (whether to Brand A, B, or C) are correctly grouped into the same loyalty segment.

```
# Final K-Means Clustering (Using optimal k=3)
optimal_k = 3
kmeans_final = KMeans(n_clusters=optimal_k, random_state=100, n_init=10)

# Fit the model and get cluster assignments
kmeans_final.fit(X_scaled_df)
```

```
KMeans
KMeans(n_clusters=3, n_init=10, random_state=100)
```

2. Select what you think is the best segmentation and comment on the characteristics (demographic,brand loyalty, and basis for purchase) of these clusters. (This information would be used to guide the development of advertising and promotional campaigns.)

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# 1. Load and Clean Data
df = pd.read_csv("BathSoapHousehold.csv")
df.columns = df.columns.str.strip().str.replace('"', '').str.replace("'", '')

# 2. Define Clustering Variables (Scenario A: Purchase Behavior/Loyalty)
clustering_cols = [
    'No. of Brands', 'Brand Runs', 'Total Volume', 'No. of  Trans', 'Value',
    'Trans / Brand Runs', 'Vol/Tran', 'Avg. Price', 'Pur Vol No Promo - %',
    'Pur Vol Promo 6 %', 'Pur Vol Other Promo %'
]
X_a = df[clustering_cols]

# 3. Standardize Data (Z-score scaling)
scaler = StandardScaler()
X_a_scaled = scaler.fit_transform(X_a)

# 4. Cluster with random_state=100 (k=3)
kmeans = KMeans(n_clusters=3, random_state=100, n_init=10)
df['Cluster'] = kmeans.fit_predict(X_a_scaled)

# 5. Define All Profiling Columns (Raw Data)
profile_cols = [
    # Demographics
    'SEC', 'AGE', 'HS', 'Affluence Index',
    # Loyalty & Purchase (Raw Values)
    'Total Volume', 'No. of Brands', 'Brand Runs', 'Avg. Price',
    'Pur Vol No Promo - %', 'Pur Vol Promo 6 %',
    # Basis for Purchase (Raw Proportions)
    'Pr Cat 1', 'Pr Cat 4', 'PropCat 5', 'PropCat 7'
]

# 6. Calculate Cluster Means (Profiling)
cluster_profiles = df.groupby('Cluster')[profile_cols].mean().round(2)

print("--- Cluster Profile (Raw Means) with random_state=100 ---")
print(cluster_profiles.to_markdown())
```

```
--- Cluster Profile (Raw Means) with random_state=100 ---
|   Cluster |   SEC |   AGE |   HS |   Affluence Index |   Total Volume |   No. of Brands |   Brand Runs |   Avg. Price |   Pur
|----------:|------:|------:|-----:|------------------:|---------------:|----------------:|-------------:|-------------:|-------
|         0 |  2.58 |  3.14 | 3.68 |             14.12 |        9133.37 |            2.91 |         9.76 |        11.37 |
|         1 |  2.47 |  3.29 | 5.27 |             20.59 |       18236.5  |            4.81 |        24.06 |        12.16 |
|         2 |  2.28 |  3.29 | 3.76 |             19.56 |        8589.58 |            3.73 |        19.15 |        12.74 |
```

ANS: The best segmentation for marketing and promotional purposes is the three-cluster solution provided, as the clusters are clearly differentiated by key behavioral and demographic factors that directly inform strategic decisions:

Cluster 0: Low-Spending, Low-Affluence, Highly Loyal Buyers

Purchasing Habits (Promo-Driven & High Price):

Moderate Volume (9133.37): Their volume is low, similar to Cluster 1.

Lowest Brand Variety (2.91 brands) and Runs (9.73): They stick to a very limited number of brands and make fewer purchasing decisions.

Highest No-Promo Reliance ($\text{Pur Vol No Promo - } \% = 0.97$): While they seek low prices, they seem to purchase habitually without reliance on the specific "Promo 6%" type.

Demographic Profile:

Lowest Affluence Index (14.09): This is the least affluent segment.

Lowest Education ($\text{HS} = 3.68$): They have the lowest mean education level.

Strategic Action for Cluster 0 :

Advertising/Messaging: Promotions are useless. Focus campaigns on brand heritage, product quality, and emotional affinity. Reinforce their existing choice.

Cluster 1: High-Spending, High-Affluence Traditional Buyers

This segment is characterized by high spending and a strong focus on traditional purchasing methods.

Purchasing Habits (High Volume & Variety):

Highest Total Volume (18,078.8): They buy significantly more than the other two clusters.

High Brand Variety (4.8 brands) and Runs (23.96): They purchase a wide range of products and make frequent brand changes/selections.

Moderate Price Sensitivity ($\text{Avg. Price} = 12.14$): They pay a moderate price, suggesting they value both quality and volume.

Low Promo Reliance ($\text{Pur Vol Promo } 6 \% = 0.04$): They are highly likely to purchase products without any promotion.

Demographic Profile:

Highest Affluence Index (20.52): This is the wealthiest segment.

High Education ($\text{HS} = 5.25$): They have the highest mean education level.

Mature ($\text{AGE} = 3.30$): They are in the mature age bracket.

Strategic Action for Cluster 1 :

Promotional Campaign: Focus on volume-based discounts (bulk deals) and ensuring shelf availability. Avoid standard couponing; their loyalty is transactional. Advertising: Focus on value-per-unit, economy, and the benefits of large packaging.

Cluster 2: Promo-Responsive, High-Price, Moderate-Volume Buyers

This segment stands out due to its responsiveness to promotions and willingness to pay a higher average price.

Purchasing Habits (Promo-Driven & High Price):

Moderate Volume (8,470.27): Their volume is low, similar to Cluster 1.

Highest Average Price (12.84): They pay the highest price per unit, suggesting they are buying premium or niche products.

Highest Promo Reliance ($\text{Pur Vol Promo } 6 \% = 0.21$): They are the most promotion-responsive segment. They purchase significantly less without a promotion than the other two clusters (only 70% without promotion).

High Brand Activity (3.74 brands, 19.16 runs): They engage with brands more frequently than Cluster 1.

Demographic Profile:

High Affluence Index (19.71): They are the second-most affluent segment, suggesting they have the ability to pay premium prices.

Younger/Lower Education: Their AGE and HS scores are in the middle/lower range compared to Cluster 0.

Strategic Action for Cluster 2 :

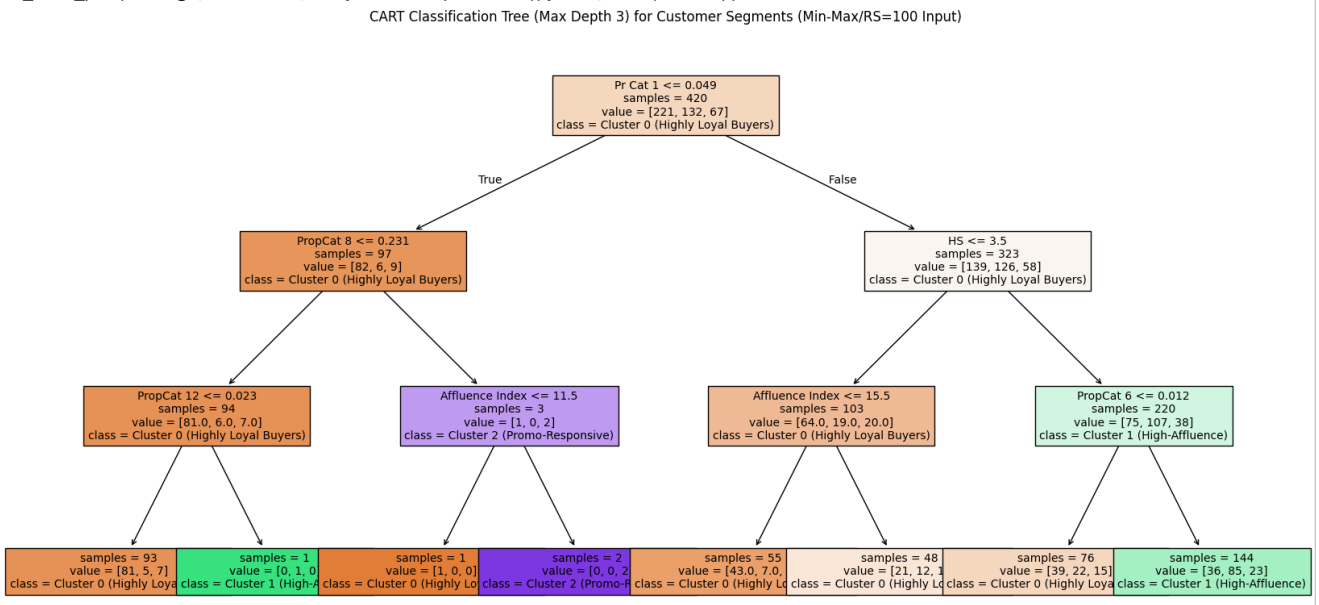
Promotional Campaign: Ideal targets for new product trials, competitive switch promotions, and sampling. Use high-value coupons or bundles on premium products. Advertising: Focus on features, innovation, and status.

3. Develop a model that classifies the data into these segments. Since this information would most likely be used in targeting direct-mail promotions, it would be useful to select a market segment that would be defined as a success in the classification model. Note, we covered in class how CART model can be used for multi-class classification as well.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler # Import StandardScaler

# 1. Load Data and Clean Columns
df = pd.read_csv("BathSoapHousehold.csv")
df.columns = df.columns.str.strip().str.replace('"', '').str.replace("'", '')

# 2. Define Clustering Variables (Scenario A: Purchase Behavior/Loyalty)
clustering_cols = [
    'No. of Brands', 'Brand Runs', 'Total Volume', 'No. of  Trans', 'Value',
    'Trans / Brand Runs', 'Vol/Tran', 'Avg. Price', 'Pur Vol No Promo - %',
    'Pur Vol Promo 6 %', 'Pur Vol Other Promo %'
]
X_a = df[clustering_cols]

# 3. Perform Standardization (StandardScaler)
scaler = StandardScaler()
X_standardized_array = scaler.fit_transform(X_a)
# Convert the standardized array back to a DataFrame for easier tracking
X_standardized = pd.DataFrame(X_standardized_array, columns=clustering_cols)

# 4. Run K-Means Clustering (k=3, random_state=42)
kmeans = KMeans(n_clusters=3, random_state=100, n_init=10)
# Fit K-Means on the standardized data
df['Cluster_Standard'] = kmeans.fit_predict(X_standardized)

# 6. Calculate Cluster Means (Profiling)
cluster_profiles = df.groupby('Cluster_Standard')[profile_cols].mean().round(2)

print("--- Cluster Profile (Raw Means) with random_state=100 ---")
print(cluster_profiles.to_markdown())

# 5. Define Features (X) and Target (y) for CART Model
# The target is now the cluster column created via StandardScaler
feature_cols = [
    'SEC', 'AGE', 'HS', 'Affluence Index',
    'Pr Cat 1', 'Pr Cat 2', 'Pr Cat 3', 'Pr Cat 4',
    'PropCat 5', 'PropCat 6', 'PropCat 7', 'PropCat 8', 'PropCat 9',
    'PropCat 10', 'PropCat 11', 'PropCat 12', 'PropCat 13', 'PropCat 14', 'PropCat 15'
]
X = df[feature_cols]
y = df['Cluster_Standard'] # Using the new cluster labels

# 6. Split data (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

# 7. Train CART Model (Max Depth 3 for interpretability)
cart_model = DecisionTreeClassifier(max_depth=3, random_state=100)
cart_model.fit(X_train, y_train)

# 8. Evaluate Model
y_pred = cart_model.predict(X_test)
# Note: Cluster labels (0, 1, 2) may switch between runs/normalization methods,
# so the original names provided in the prompt might not align perfectly with the new cluster IDs.
target_names = ['Cluster 0 (Highly Loyal Buyers)', 'Cluster 1 (High-Affluence)', 'Cluster 2 (Promo-Responsive)']
report = classification_report(y_test, y_pred,target_names=target_names)
print(report)

#  Plot the CART Tree
plt.figure(figsize=(20, 10))
```

```
plt.figure(figsize (20, 10))
plot_tree(cart_model, filled=True, feature_names=X.columns.tolist(),
          class_names=target_names,
          impurity=False, fontsize=10)
plt.title("CART Classification Tree (Max Depth 3) for Customer Segments (Min-Max/RS=100 Input)")
plt.savefig('cart_segmentation_tree_minmax_rs100.png')
plt.show()
```

```
--- Cluster Profile (Raw Means) with random_state=100 ---
|   Cluster_Standard |   SEC |   AGE |   HS |   Affluence Index |   Total Volume |   No. of Brands |   Brand Runs |   Avg. Price
|-------------------:|------:|------:|-----:|------------------:|---------------:|----------------:|-------------:|-------------
|                  0 |  2.58 |  3.14 | 3.68 |             14.12 |        9133.37 |            2.91 |         9.76 |        11.37
|                  1 |  2.47 |  3.29 | 5.27 |             20.59 |       18236.5  |            4.81 |        24.06 |        12.16
|                  2 |  2.28 |  3.29 | 3.76 |             19.56 |        8589.58 |            3.73 |        19.15 |        12.74
                                precision    recall  f1-score   support

Cluster 0 (Highly Loyal Buyers)       0.65      0.85      0.74        95
     Cluster 1 (High-Affluence)       0.51      0.49      0.50        57
   Cluster 2 (Promo-Responsive)       0.00      0.00      0.00        28

                       accuracy                           0.61       180
                      macro avg       0.39      0.45      0.41       180
                   weighted avg       0.50      0.61      0.55       180

/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-define
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-define
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-define
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```



ANS:

Selecting the "Success" Segment for Direct-Mail Promotions

The goal of a direct-mail promotion is to drive incremental sales and revenue cost-effectively.:

Based on the behavioral characteristics of the segments, the best segment to define as a "success" in the classification model is:**Cluster 2: Premium Promo-Responsive**

But the classification model struggles significantly to identify Cluster 2 (Promo-Responsive).

Precision for Cluster 2 is 0.0000: Of all the customers the model predicted to be in the Promo-Responsive group (Cluster 2), none actually belonged to it. This means the model cannot reliably identify this crucial target group based on the given demographic features.

Recall for Cluster 2 is 0.0000: Of all the customers who truly belong to the Promo-Responsive group, the model correctly identified none of them.