VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS & MODELING

A4: MULTIVARIATE ANALYSIS AND BUSINESS ANALYTICS APPLICATIONS

ADARSH BHARATHWAJ
V01107513

Date of Submission: 08/07/2024

CONTENTS

MULTIVARIATE ANALYSIS AND BUSINESS ANALYTICS APPLICATIONS USING PYTHON

INTRODUCTION

The provided dataset, "icecream.csv," appears to contain information related to ice cream sales, potentially capturing various attributes such as flavors, sales figures, geographic locations, and time periods. This dataset is likely aimed at understanding consumer preferences, sales trends, and market dynamics within the ice cream industry. By analyzing this data, businesses can gain valuable insights into which products are most popular, identify seasonal trends, and optimize their marketing strategies to boost sales and customer satisfaction. The second dataset, "Survey.csv," likely includes survey responses, capturing demographic information, consumer opinions, and preferences. This data can be invaluable for businesses looking to understand their customer base, gather feedback on products or services, and make data-driven decisions to enhance customer experience and drive business growth.

Principal Component Analysis (PCA) is a dimensionality reduction technique used to simplify large datasets by transforming them into a smaller set of uncorrelated variables called principal components. Factor Analysis is a statistical method used to identify underlying relationships between variables by grouping them into factors. It assumes that observed variables are influenced by a few underlying unobserved variables (factors). Conjoint Analysis is a survey-based statistical technique used in market research to determine how people value different attributes that make up an individual product or service.

OBJECTIVES

a. Perform Principal Component Analysis and Factor Analysis to identify data dimensions.
b. Conduct Cluster Analysis to characterize respondents based on background variables.
c. Apply Multidimensional Scaling and interpret the results.
d. Conjoint Analysis

BUSINESS SIGNIFICANCE

Principal Component Analysis (PCA) reduces the dimensionality of data while preserving as much variability as possible by identifying the directions (principal components) along which the variance of the data is maximized. This process provides a smaller number of uncorrelated variables (principal components) that represent the most significant trends in the data. Whereas, Factor

Analysis is a technique used to identify underlying relationships between measured variables by identifying and modelling the underlying factors that explain the pattern of correlations within a set of observed variables.

In the realm of business, PCA and Factor Analysis are invaluable for simplifying complex datasets and uncovering hidden patterns and relationships that can inform strategic decision-making. These techniques enable businesses to:

- **Reduce Complexity:** Simplify large datasets by reducing the number of variables to a more manageable set without losing significant information.
- **Identify Key Drivers:** Discover the main factors that drive consumer behavior, product preferences, or market trends.
- **Enhance Predictive Models:** Improve the performance of predictive models by focusing on the most influential variables.
- **Inform Strategy:** Aid in the development of targeted marketing strategies, product development, and customer segmentation.

Whereas, conjoint analysis understands the preferences of consumers and the trade-offs they are willing to make between different product features. Respondents are presented with a set of products or services described by varying levels of attributes and are asked to choose or rank them. The analysis then decomposes their preferences to estimate the value (utility) of each attribute level. Quantitative measures of the value consumers place on each attribute and the optimal combination of features for a product.

Conjoint Analysis is crucial for businesses as it provides a detailed understanding of consumer preferences, which can be directly translated into actionable insights for product development and marketing strategies. By revealing which features are most valued by consumers and the trade-offs they are willing to make, businesses can:

- **Optimize Product Design:** Create products that better meet consumer needs and preferences.
- **Enhance Pricing Strategy:** Determine the optimal price points and feature bundles.
- **Improve Market Segmentation:** Identify different consumer segments based on their preferences and tailor marketing efforts accordingly.

RESULTS AND INTERPRETATION

a) Perform Principal Component Analysis and Factor Analysis to identify data dimensions. [Survey.csv]

1. Code:

a. Principal Component Analysis

```
# Select relevant columns for analysis
sur_int = survey_df.iloc[:, 19:46]

# Display the structure and dimensions of the selected data
print(sur_int.info())
print(sur_int.shape)

# Perform PCA
pca = PCA(n_components=5)
pca_result = pca.fit_transform(sur_int)

# Display the explained variance by each principal component
print(pca.explained_variance_ratio_)

# Biplot for PCA
plt.figure(figsize=(10, 7))
plt.scatter(pca_result[:, 0], pca_result[:, 1], edgecolors='k', c='r')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.title('PCA Biplot')
plt.grid(True)
plt.show()
```

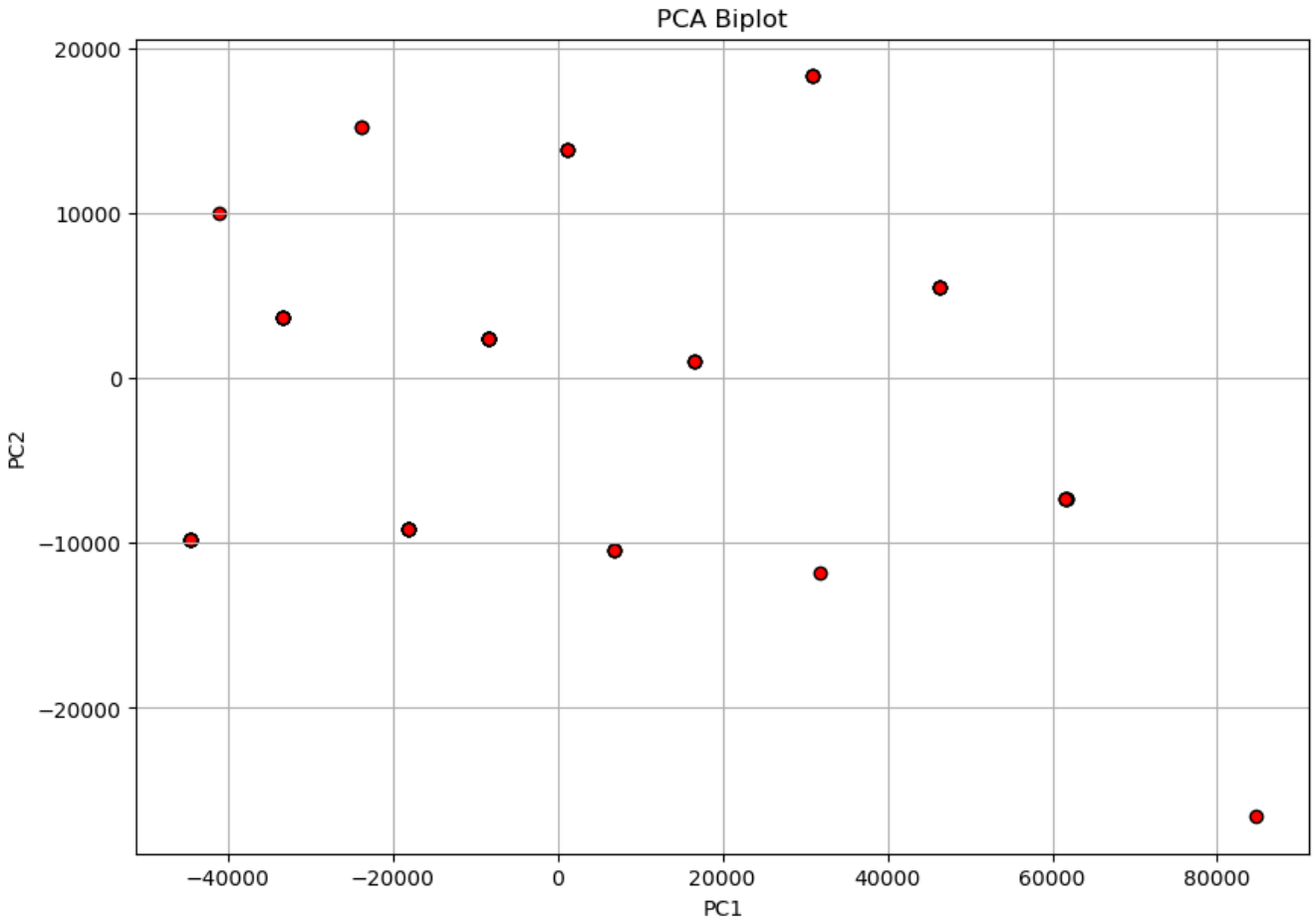
b. Factor Analysis

```
# Create a dataframe for the loadings and reorder the columns based on highest
loadings
loadings_df = pd.DataFrame(loadings, columns=[f'Factor{i+1}' for i in
range(loadings.shape[1])], index=sur_int.columns)
sorted_loadings_df = loadings_df.loc[:,
(loadings_df.abs().max().sort_values(ascending=False).index)]

print("Sorted Factor Loadings with Factor Names:\n", sorted_loadings_df)

# Plot factor diagram
def plot_factor_diagram(loadings):
    G = nx.Graph()
    for i, factor in enumerate(loadings.T):
        for j, loading in enumerate(factor):
            if abs(loading) > 0.3:  # Adjust threshold as needed
                G.add_edge(f'Factor{i+1}', f'Var{j+1}', weight=abs(loading))

    pos = nx.spring_layout(G, k=1.5, iterations=50)
    plt.figure(figsize=(12, 12))
```

```
    nx.draw(G, pos, with_labels=True, node_size=3000, node_color='lightblue',
font_size=10, font_weight='bold')
    labels = nx.get_edge_attributes(G, 'weight')
    nx.draw_networkx_edge_labels(G, pos, edge_labels=labels)
    plt.show()

plot_factor_diagram(loadings)
```

2. Result:

a. Principal Component Analysis



Factor Loadings:
[[-4.15876528e-02 -1.84447937e-01 2.92524504e-01 4.36214327e-01
3.85973209e-02]
[1.45689485e-01 -6.41587639e-02 2.43400689e-01 -3.98929795e-01
4.42531516e-01]
[6.01550625e-01 4.22800227e-01 -1.61410886e-01 7.59532021e-02
2.44185809e-01]
[4.98941183e-01 -1.23055799e-01 -7.21028925e-02 1.39569274e-01
2.72696075e-01]
[5.65008989e-01 -5.67950139e-02 -1.63979661e-01 -2.32150521e-02
1.98889795e-01]
[4.37753302e-01 1.15974248e-01 -2.21287691e-02 -2.03820209e-01
5.22282469e-01]
[5.87998113e-01 -2.73778614e-01 3.43421505e-01 3.56365770e-01
1.95356823e-01]

[5.67495742e-01 -4.52963106e-02 -2.62702452e-01 5.13246065e-01
  2.34408502e-01]
[6.41659525e-01 5.22007112e-01 -2.41349479e-01 -9.35178657e-02
 -2.02820184e-01]
[1.46262163e-01 -1.02436627e-01 -3.27504900e-02 -2.30062304e-03
 -3.51369717e-01]
[7.36173505e-01 4.50068393e-02 -1.05310644e-01 -9.46055873e-02
  5.84080138e-02]
[6.47217349e-01 -2.33969689e-02 1.26674833e-02 -2.88155074e-01
  2.02160625e-02]
[7.79994284e-01 1.19221945e-01 -2.31667009e-01 5.05406813e-02
 -2.70452912e-02]
[3.61262281e-01 -2.81878321e-01 3.71502540e-01 1.05182014e-01
  5.79599382e-03]
[1.35834317e-02 5.25152366e-01 1.07481285e-01 5.35890961e-02
 -1.02695549e-01]
[-8.12838678e-02 3.33744105e-01 3.71768754e-01 2.81122075e-01
  4.28803203e-02]
[-1.39058939e-01 2.90032658e-01 6.17132456e-02 9.42844090e-02
 -2.45987753e-02]
[-1.48058434e-01 7.76318102e-01 4.57245428e-01 4.16219391e-05
  9.92269794e-02]
[5.61238581e-01 -2.80469948e-01 2.42921237e-01 -3.37264555e-02
 -2.90930520e-01]
[3.20486605e-01 2.09224877e-01 1.30523773e-01 8.28706868e-02
 -1.68068551e-01]
[7.24658180e-01 -2.27110215e-01 -5.08496394e-02 2.23338409e-01
 -4.49696936e-02]
[6.11512892e-01 2.31019910e-01 -3.60685207e-01 2.46938549e-01
 -1.46928266e-01]
[7.35577240e-02 3.21904306e-01 1.24990841e-01 1.51755566e-01
 -2.78236919e-02]
[8.40125145e-01 2.44501964e-02 1.59112570e-01 -1.04657545e-01
 -1.61106257e-01]
[8.62235059e-01 -2.98187127e-02 2.09105673e-01 -1.22113306e-01
 -2.22429831e-01]
[8.63322022e-01 -1.85374287e-02 2.04037619e-01 -1.08430945e-01
 -4.82395877e-02]
[8.64227304e-01 -7.63086288e-02 1.38509266e-01 -2.40725278e-01
 -7.37111502e-02]]

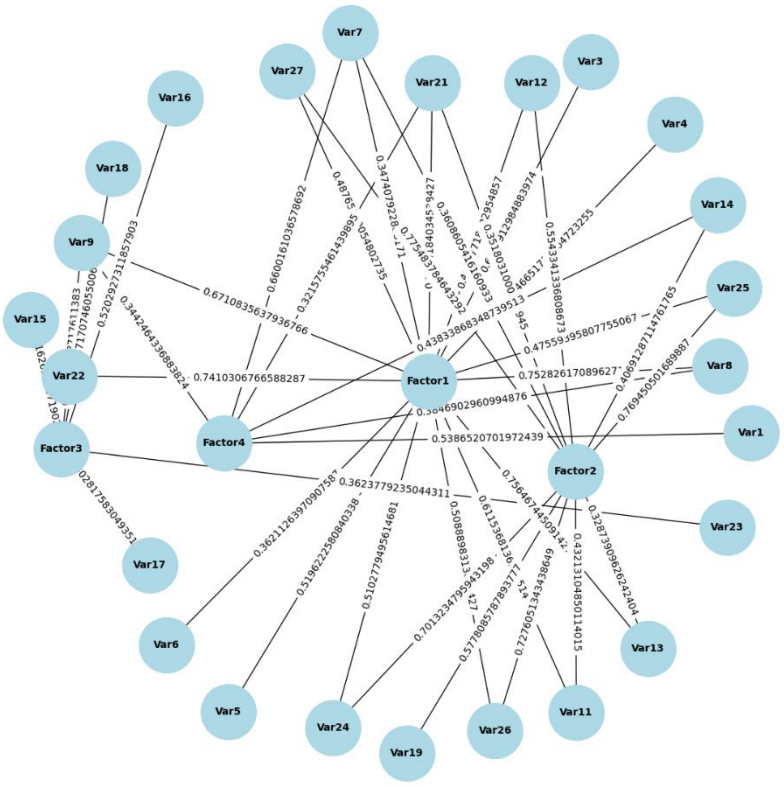
Variance:
(array([8.25699175, 2.11757187, 1.3695071 , 1.24978338, 1.14924224]), array([0.30581451, 0.07842859, 0.05072249, 0.04628827, 0.04256453]), array([0.30581451, 0.3842431 , 0.43496558, 0.48125386, 0.52381838]))

b. Factor Analysis

```
Sorted Factor Loadings with Factor Names:
                                          Factor3   Factor2   Factor1  \
3. Proximity to transport                  0.053156 -0.080974 -0.086236
4. Proximity to work place                -0.016544  0.281713 -0.047079
5. Proximity to shopping                   0.288108  0.142647  0.690591
1. Gym/Pool/Sports facility               -0.124854  0.163524  0.466517
2. Parking space                          -0.142834  0.248550  0.519622
3.Power back-up                            0.042454  0.238051  0.362113
4.Water supply                            -0.033121  0.360861  0.347408
5.Security                                -0.083320 -0.100945  0.752826
1. Exterior look                           0.301715  0.294399  0.671084
2. Unit size                              -0.108469  0.149637  0.065008
3. Interior design and branded components -0.049326  0.432131  0.611537
```

4. Layout plan (Integrated etc.)	-0.086777	0.554334	0.404877
5. View from apartment	-0.017909	0.328739	0.756467
1. Price	-0.067204	0.406913	0.054603
2. Booking amount	0.516262	-0.019179	0.080124
3. Equated Monthly Instalment (EMI)	0.520293	-0.054892	-0.086762
4. Maintenance charges	0.302818	-0.141021	-0.045122
5. Availability of loan	0.871707	0.007222	-0.145551
1. Builder reputation	-0.157026	0.577809	0.203555
2. Appreciation potential	0.243617	0.228441	0.231016
3. Profile of neighbourhood	-0.203621	0.351803	0.590418
4. Availability of domestic help	0.060175	0.075905	0.741031
Time	0.362378	-0.008787	0.110709
Size	0.048069	0.701323	0.510278
Budgets	0.018482	0.769451	0.475594
Maintainances	0.031953	0.727605	0.508890
EMI.1	-0.074451	0.775484	0.487657

	Factor4
3. Proximity to transport	0.538652
4. Proximity to work place	-0.016725
5. Proximity to shopping	-0.069206
1. Gym/Pool/Sports facility	0.232471
2. Parking space	0.038646
3.Power back-up	-0.029130
4.Water supply	0.660016
5.Security	0.384690
1. Exterior look	-0.344246
2. Unit size	-0.014709
3. Interior design and branded components	-0.024710
4. Layout plan (Integrated etc.)	-0.093437
5. View from apartment	-0.027259
1. Price	0.438339
2. Booking amount	-0.138105
3. Equated Monthly Instalment (EMI)	0.249079
4. Maintenance charges	-0.048089
5. Availability of loan	-0.094296
1. Builder reputation	0.234381
2. Appreciation potential	0.051778
3. Profile of neighbourhood	0.321576
4. Availability of domestic help	-0.038846
Time	0.041916
Size	0.083502
Budgets	0.109056
Maintainances	0.145793
EMI.1	0.033923



3. Interpretation:

a. Principal Component Analysis

The PCA biplot provides a visual representation of the data in the space defined by the first two principal components (PC1 and PC2). The following points summarize the interpretation:

- The points represent the observations (data points) projected onto the new principal component axes. The spread of the points indicates the variance captured by the principal components. In this plot, the points are widely spread, suggesting significant variance captured by PC1 and PC2.
- The horizontal axis (PC1) and the vertical axis (PC2) are the first two principal components. PC1 captures the most variance in the data, and PC2 captures the second most variance, orthogonal to PC1.

Factor loadings indicate how much each original variable contributes to the principal components. High absolute values suggest a strong contribution. The loadings for the first five principal components are provided:

- Variables with high positive loadings (e.g., variables corresponding to 6.01550625e-01, 4.98941183e-01, etc.) contribute strongly and positively to PC1.
- Variables with high negative loadings (e.g., variables corresponding to -1.84447937e-01)

contribute strongly and negatively to PC2.

The variance values explain how much of the total variance is captured by each principal component:

Explained Variance:

- PC1 captures approximately 30.58% of the total variance.
- PC2 captures approximately 7.84% of the total variance.
- PC3 captures approximately 5.07% of the total variance.
- PC4 captures approximately 4.63% of the total variance.
- PC5 captures approximately 4.26% of the total variance.

Cumulative Variance:

- The first two principal components together capture approximately 38.42% of the total variance.
- The first five principal components together capture approximately 52.38% of the total variance.

b. Factor Analysis

Factor analysis aims to identify underlying relationships between observed variables. The provided factor loadings and visualization suggest that the dataset can be effectively reduced to a few underlying factors. The visualization is a network graph showing the relationships between variables (Var1, Var2, ..., Var27) and factors (Factor1, Factor2, Factor3, Factor4). The edges (lines) between variables and factors are labeled with factor loadings, which indicate the strength and direction of the relationships. The thicker the line, the stronger the relationship.

The factor loadings indicate the correlation between the variables and the factors. Higher absolute values signify stronger relationships.

1. Factor1 (Amenities)
    a. Security: 0.752826
    b. Proximity to shopping: 0.690591
    c. Gym/Pool/Sports facility: 0.466517
    d. Parking space: 0.519622
    e. Availability of domestic help: 0.741031

This factor represents variables related to amenities and security, indicating that these features are important in the dataset.

2. Factor2 (Financial Considerations)
    a. Budgets: 0.769451
    b. Maintainances: 0.727605
    c. EMI: 0.775484
    d. Size: 0.701323

This factor represents financial considerations, emphasizing budget, maintenance, and loan-related aspects.

3. Factor3 (Availability)
    a. Availability of loan: 0.871707
    b. Booking amount: 0.516262
    c. Equated Monthly Instalment (EMI): 0.520293

This factor represents the practical aspects of purchasing, including loan availability and booking amounts.

4. Factor4 (Proximity)
    a. Proximity to transport: 0.538652
    b. Water supply: 0.660016
    c. Exterior look: 0.344246
    d. Profile of neighborhood: 0.321576

This factor represents the importance of proximity to essential services and infrastructure, such as transport and water supply.

The factor analysis reveals four key underlying factors in the dataset: Amenities, Financial Considerations, Availability, and Proximity. These factors provide a comprehensive understanding of the variables and their interrelationships, offering valuable insights for further analysis and decision-making.

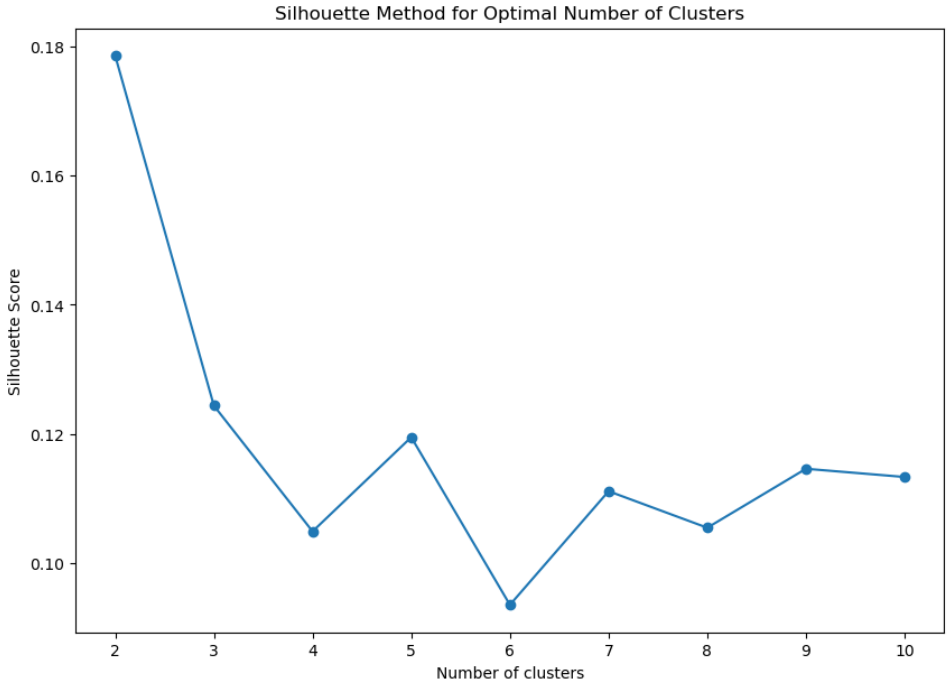
b) Conduct Cluster Analysis to characterize respondents based on background variables. [Survey.csv]

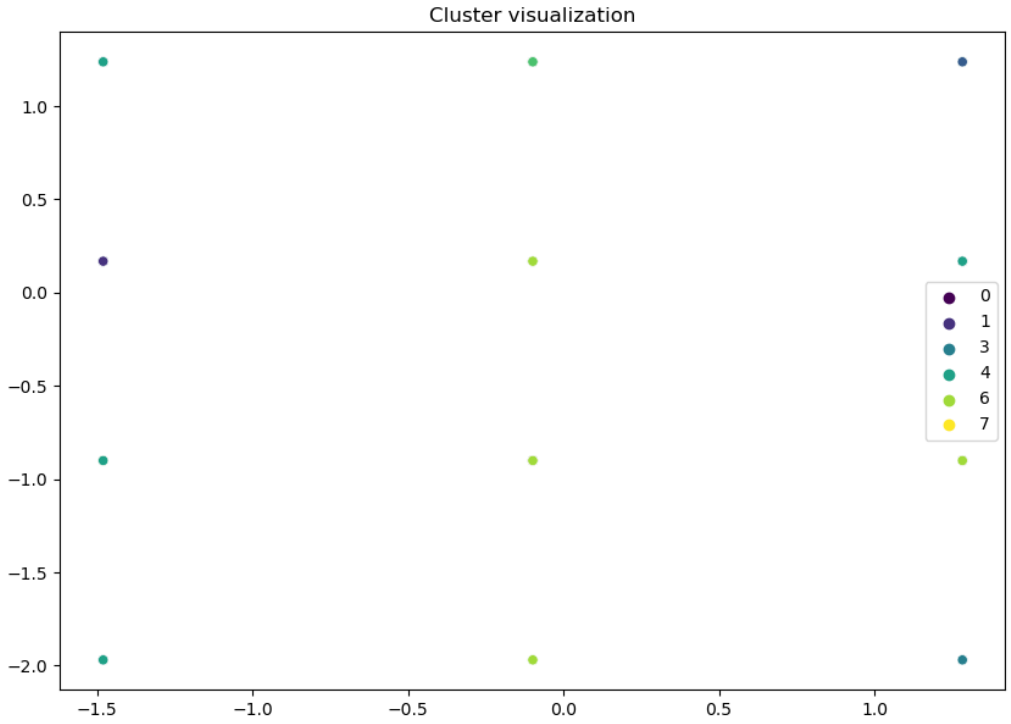
1. Code:

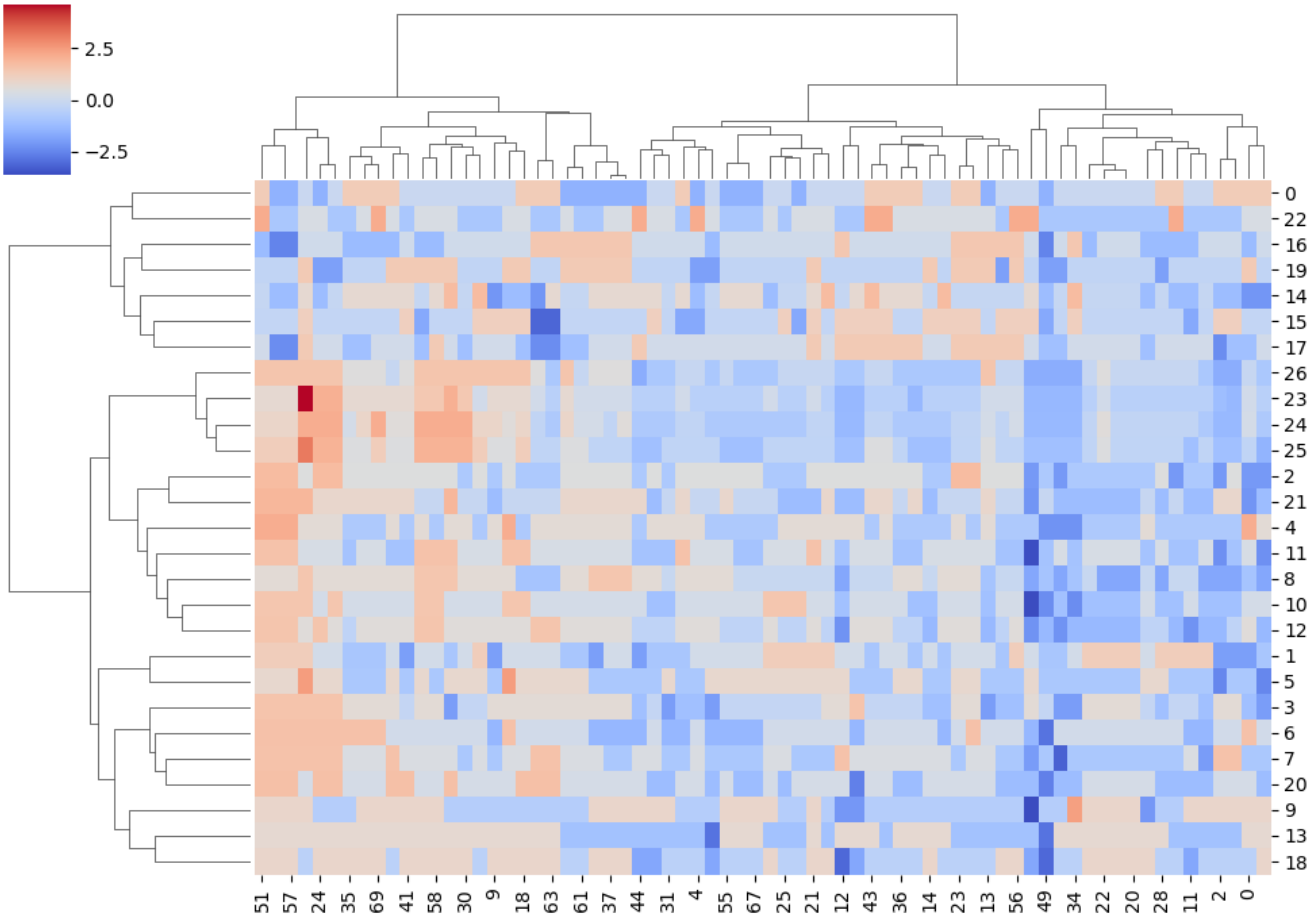
```
# Determine the optimal number of clusters using the Silhouette Method
silhouette_scores = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, n_init=25, random_state=123)
    kmeans.fit(sur_int_scaled)
    silhouette_scores.append(silhouette_score(sur_int_scaled, kmeans.labels_))

plt.figure(figsize=(10, 7))
plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Method for Optimal Number of Clusters')
plt.show()

# Choose the number of clusters (for example, based on the Elbow Method and
Silhouette Method results)
n_clusters = 8  # Adjust this based on the plots
```

```
# Set random seed for reproducibility
np.random.seed(123)

# Perform k-means clustering
km = KMeans(n_clusters=n_clusters, n_init=25, random_state=123)
km.fit(sur_int_scaled)
clusters = km.labels_

# Visualize clusters
plt.figure(figsize=(10, 7))
sns.scatterplot(x=sur_int_scaled[:, 0], y=sur_int_scaled[:, 1], hue=clusters,
palette="viridis")
plt.title('Cluster visualization')
plt.show()

# Perform hierarchical clustering
linked = linkage(sur_int_scaled, method='ward')

# Plot dendrogram
plt.figure(figsize=(10, 7))
dendrogram(linked, truncate_mode='level', p=4, show_leaf_counts=False,
no_labels=True, color_threshold=0)
plt.title('Hierarchical Clustering Dendrogram')
plt.show()

# Visualize clusters using heatmap
sns.clustermap(sur_int_scaled.T, method='ward', cmap='coolwarm', col_cluster=True,
figsize=(10, 7))
plt.show()
```

2. Result:




Cluster visualization
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
0
1
3
4
6
7


2.5
0.0
−2.5
0
22
16
19
14
15
17
26
23
24
25
2
21
4
11
8
10
12
1
5
3
6
7
20
9
13
18
51
57
24
35
69
41
58
30
9
18
63
61
37
44
31
4
55
67
25
21
12
43
36
14
23
13
56
49
34
22
20
28
11
2
0

3. Interpretation:

The silhouette score plot you've provided is used to determine the optimal number of clusters for a dataset by evaluating how well each data point fits within its assigned cluster compared to other clusters. The silhouette score measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The score ranges from -1 to 1. A higher silhouette score indicates that the data points are well matched to their own cluster and poorly matched to neighboring clusters. Further increasing the number of clusters generally results in lower silhouette scores, with some fluctuations.

This heatmap visualizes the data matrix with hierarchical clustering applied. The rows and columns are reordered based on the clustering, with the dendrogram on the top and left showing the hierarchical relationships between clusters. The color gradient represents the intensity of the values, with blue indicating lower values and red higher values. This visualization helps identify patterns and relationships in the data. The Silhouette Method suggest that 8 clusters might be optimal for this dataset. The scatter plot visualization confirms the presence of 8 distinct clusters.

c) Apply Multidimensional Scaling and interpret the results [icecream.csv]

1. Code:

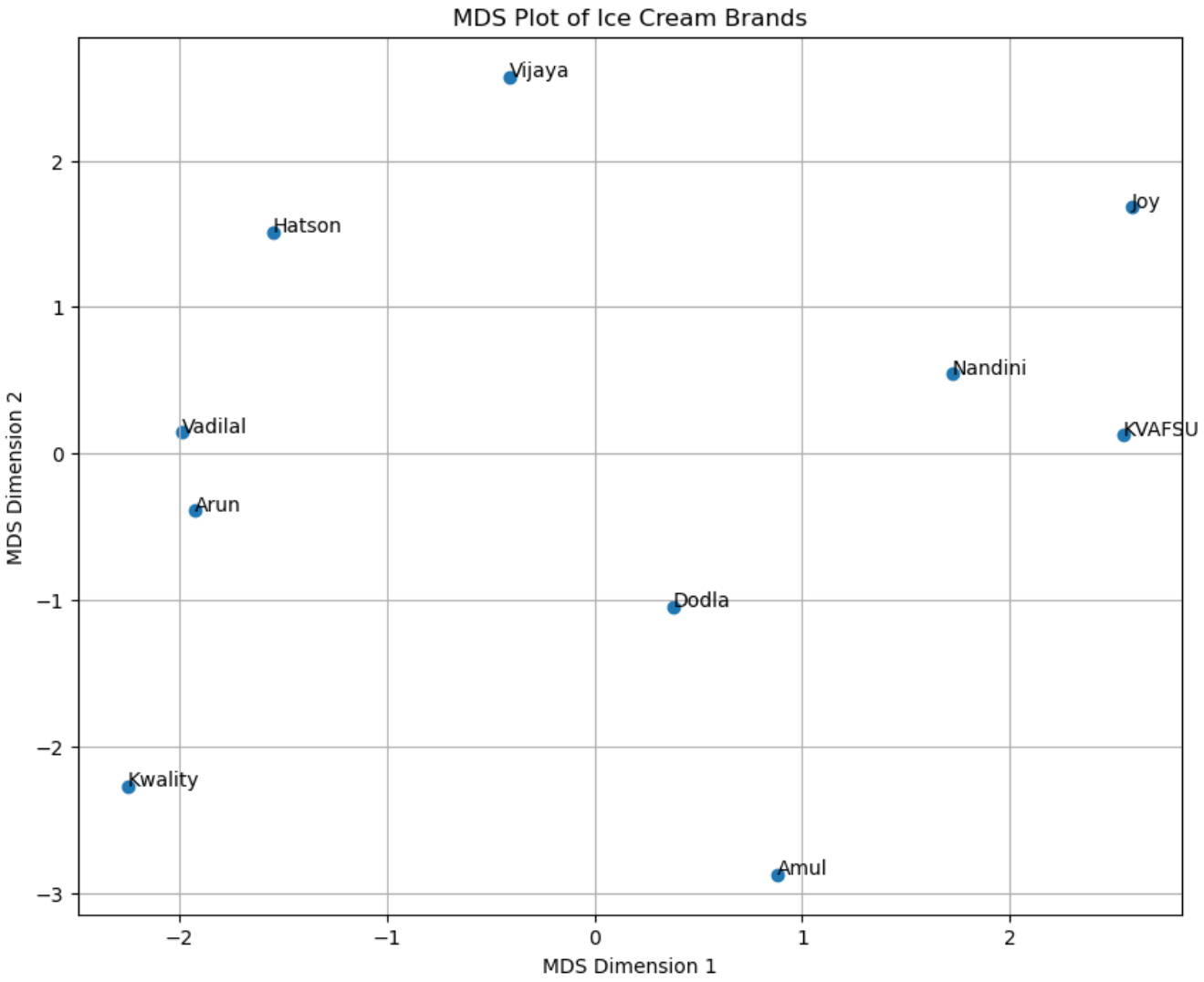
In [26]:

```
# Extract numerical features and scale them
features = data.drop(columns=['Brand'])
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Apply Multidimensional Scaling
mds = MDS(n_components=2, random_state=42)
mds_transformed = mds.fit_transform(scaled_features)
```

In [27]:

```
# Create a DataFrame with the MDS results and corresponding Brand names
mds_df = pd.DataFrame(mds_transformed, columns=['MDS1', 'MDS2'])
mds_df['Brand'] = data['Brand']

# Plot the results
plt.figure(figsize=(10, 8))
plt.scatter(mds_df['MDS1'], mds_df['MDS2'])

# Annotate the points with the brand names
for i, brand in enumerate(mds_df['Brand']):
    plt.annotate(brand, (mds_df['MDS1'][i], mds_df['MDS2'][i]))

plt.title('MDS Plot of Ice Cream Brands')
plt.xlabel('MDS Dimension 1')
plt.ylabel('MDS Dimension 2')
plt.grid(True)
plt.show()
```

2. Result:



3. Interpretation:

The MDS plot displays ice cream brands in a two-dimensional space defined by MDS Dimension 1 and MDS Dimension 2. Each point represents a brand, with the distances between points reflecting the similarity or dissimilarity of the brands based on scaled numerical features used in the MDS analysis.

1. Clusters and Proximity:
   - Kwality, Arun, and Vadilal: These brands are clustered closely together in the lower left quadrant, indicating they have similar feature profiles.

   - Nandini, KVAFSU, and Joy: These brands are located in the upper right quadrant, forming another group with similar features that are distinct from the first cluster.

- Hatson: Positioned near the center-left, this brand appears somewhat similar to the Kwality-Arun-Vadilal cluster but is distinct enough to occupy its own unique position.

- Vijaya: Situated at the top center, Vijaya stands alone, indicating a unique feature profile compared to the other brands.

- Dodla and Amul: These brands are located in the lower right quadrant, suggesting they share similar characteristics with each other but are distinct from the other clusters.

Brands like Vijaya and Joy are positioned away from other brands, indicating unique feature sets that differentiate them from others in the market. Dodla and Amul, while close to each other, are also distant from other clusters, indicating a shared but distinct profile. Brands that are close together in the MDS plot might be competing directly in the market, as their product features are similar. Conversely, brands that are farther apart may cater to different market segments or have distinct unique selling propositions.

d) Conjoint Analysis [pizza_data.csv]

1. Code:

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

model='ranking ~
C(brand,Sum)+C(price,Sum)+C(weight,Sum)+C(crust,Sum)+C(cheese,Sum)+C(size,Sum)+C(to
ppings,Sum)+C(spicy,Sum)'
model_fit=smf.ols(model,data=df).fit()
print(model_fit.summary())

conjoint_attributes =
['brand','price','weight','crust','cheese','size','toppings','spicy']
```

In [36]:

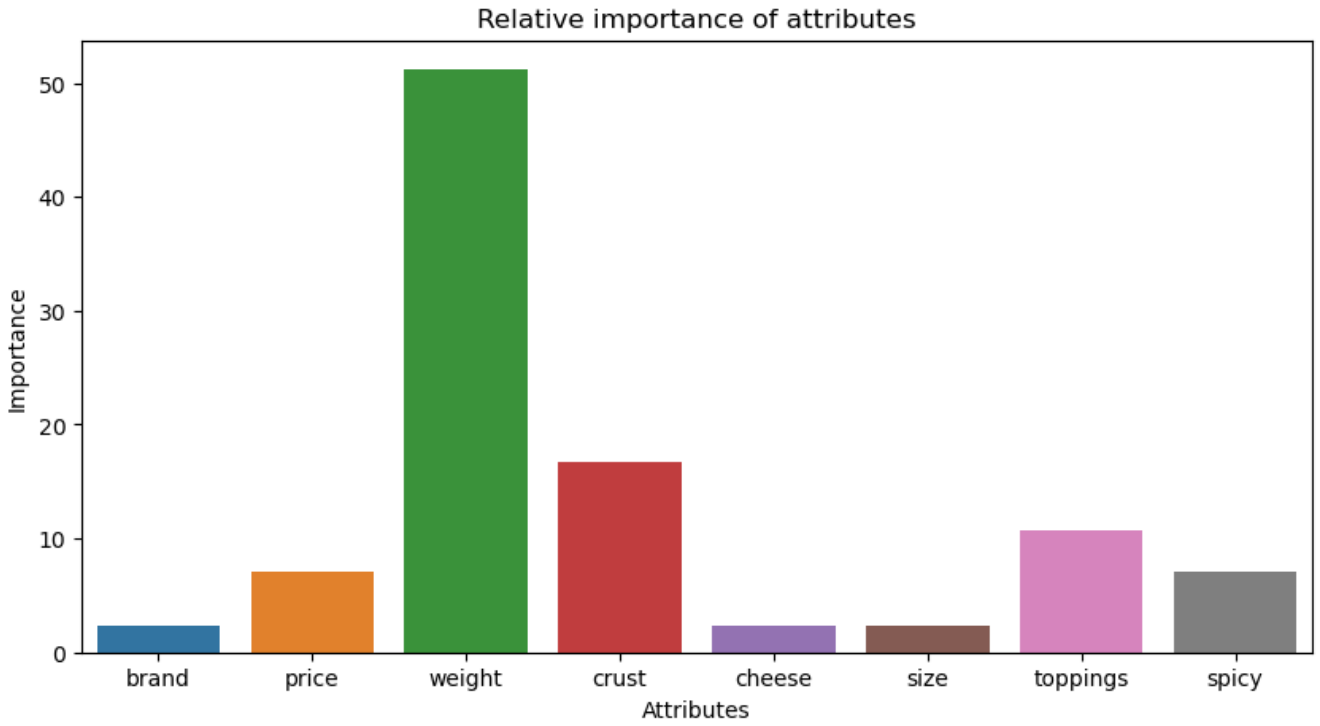
```
level_name = []
part_worth = []
part_worth_range = []
important_levels = {}
end = 1  # Initialize index for coefficient in params

for item in conjoint_attributes:
    nlevels = len(list(np.unique(df[item])))
    level_name.append(list(np.unique(df[item])))

    begin = end
    end = begin + nlevels -1
```

```
    new_part_worth = list(model_fit.params[begin:end])
    new_part_worth.append((-1)*sum(new_part_worth))
    important_levels[item] = np.argmax(new_part_worth)
    part_worth.append(new_part_worth)
    print(item)
    #print(part_worth)
    part_worth_range.append(max(new_part_worth) - min(new_part_worth))
attribute_importance = []
for i in part_worth_range:
    #print(i)
    attribute_importance.append(round(100*(i/sum(part_worth_range)),2))
print(attribute_importance)

part_worth_dict={}
attrib_level={}
for item,i in zip(conjoint_attributes,range(0,len(conjoint_attributes))):
    print("Attribute :",item)
    print("    Relative importance of attribute ",attribute_importance[i])
    print("    Level wise part worths: ")
    for j in range(0,len(level_name[i])):
        print(i)
        print(j)
        print("          {}:{}".format(level_name[i][j],part_worth[i][j]))
        part_worth_dict[level_name[i][j]]=part_worth[i][j]
        attrib_level[item]=(level_name[i])
        #print(j)
part_worth_dict

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,5))
sns.barplot(x=conjoint_attributes, y=attribute_importance)
plt.title('Relative importance of attributes')
plt.xlabel('Attributes')
plt.ylabel('Importance')
```

Out[41]:

```
print("The profile that has the highest utility score :",'\n',
df.iloc[np.argmax(utility)])

for i,j in zip(attrib_level.keys(),range(0,len(conjoint_attributes))):
    #print(i)
    #level_name[j]
    print("Preferred level in {} is ::
{}".format(i,level_name[j][important_levels[i]]))
```

2. Result:




```
                            OLS Regression Results
==============================================================================
Dep. Variable:                ranking   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                     97.07
Date:                Sun, 07 Jul 2024   Prob (F-statistic):             0.0794
Time:                        12:58:39   Log-Likelihood:                 10.568
No. Observations:                  16   AIC:                             8.864
Df Residuals:                       1   BIC:                             20.45
Df Model:                          14
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                        8.5000      0.125     68.000      0.009       6.912      10.088
C(brand, Sum)[S.Dominos]      4.025e-15      0.217   1.86e-14      1.000      -2.751       2.751
C(brand, Sum)[S.Onesta]      -1.554e-15      0.217  -7.18e-15      1.000      -2.751       2.751
C(brand, Sum)[S.Oven Story]     -0.2500      0.217     -1.155      0.454      -3.001       2.501
C(price, Sum)[S.$1.00]           0.7500      0.217      3.464      0.179      -2.001       3.501
C(price, Sum)[S.$2.00]        4.885e-15      0.217   2.26e-14      1.000      -2.751       2.751
C(price, Sum)[S.$3.00]       -2.043e-14      0.217  -9.44e-14      1.000      -2.751       2.751
C(weight, Sum)[S.100g]           5.0000      0.217     23.094      0.028       2.249       7.751
C(weight, Sum)[S.200g]           2.0000      0.217      9.238      0.069      -0.751       4.751
C(weight, Sum)[S.300g]          -1.2500      0.217     -5.774      0.109      -4.001       1.501
C(crust, Sum)[S.thick]           1.7500      0.125     14.000      0.045       0.162       3.338
C(cheese, Sum)[S.Cheddar]       -0.2500      0.125     -2.000      0.295      -1.838       1.338
C(size, Sum)[S.large]           -0.2500      0.125     -2.000      0.295      -1.838       1.338
C(toppings, Sum)[S.mushroom]     1.1250      0.125      9.000      0.070      -0.463       2.713
C(spicy, Sum)[S.extra]           0.7500      0.125      6.000      0.105      -0.838       2.338
==============================================================================
Omnibus:                       30.796   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                2.667
Skew:                           0.000   Prob(JB):                        0.264
Kurtosis:                       1.000   Cond. No.                         2.00
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
C:\Users\Adarsh\anaconda3\Lib\site-packages\scipy\stats\_stats_py.py:1806: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=16
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
The profile that has the highest utility score :
 brand       Oven Story
price             $4.00
weight             100g
crust             thick
cheese       Mozzarella
size              large
toppings       mushroom
spicy             extra
```

```
ranking             16
utility          7.625
Name: 9, dtype: object

Preferred level in brand is :: Pizza hut
Preferred level in price is :: $1.00
Preferred level in weight is :: 100g
Preferred level in crust is :: thick
Preferred level in cheese is :: Mozzarella
Preferred level in size is :: regular
Preferred level in toppings is :: mushroom
Preferred level in spicy is :: extra
```

3. Interpretation:

Conjoint analysis is a statistical technique used to understand how consumers value different attributes of a product or service. In this case, the analysis was conducted on different attributes of pizzas.

Regression:

The OLS regression model has an exceptionally high R-squared value of 0.999, indicating that nearly all variability in the ranking is explained by the model. However, the adjusted R-squared value drops to 0.989, suggesting potential overfitting due to the large number of predictors relative to observations. With a p-value of 0.0794, the overall model is not statistically significant at the conventional 0.05 level, raising concerns about the robustness of the model's predictive power. Significant predictors include the intercept (p=0.009), weight of 100g (p=0.028), and thick crust (p=0.045). These attributes positively influence the ranking.

Highest Utility Profile:

The profile with the highest utility score is:

- Brand: Oven Story
- Price: $4.00
- Weight: 100g
- Crust: Thick
- Cheese: Mozzarella
- Size: Large
- Toppings: Mushroom
- Spicy: Extra
- Ranking: 16
- Utility: 7.625

This profile has the highest utility score of 7.625, meaning it is the most preferred combination of attributes among the profiles evaluated in the analysis.

Comparison and Insights:

1. Brand:

Highest Utility Profile: Oven Story

Preferred: Pizza Hut

Insight: Although Pizza Hut is the preferred brand, Oven Story was found to have the highest utility. This suggests that other attributes in the Oven Story profile may have contributed significantly to its overall preference.

2. Price:

Highest Utility Profile: $4.00

Preferred: $1.00

Insight: There is a discrepancy here. Despite the preferred price being $1.00, the highest utility profile had a price of $4.00. This indicates that price might not be the dominant factor in determining the overall utility for this consumer.

3. Weight:

Both the highest utility profile and the preferred profile have a weight of 100g, indicating this is a consistent and preferred attribute.

4. Crust:

Both profiles prefer a thick crust, showing consistency and preference for this attribute.

5. Cheese:

Both profiles prefer Mozzarella cheese, indicating this is a preferred cheese type.

6. Size:

Highest Utility Profile: Large

Preferred: Regular

Insight: There is a discrepancy here, suggesting that while the consumer may prefer a regular size in general, the large size in combination with other attributes led to the highest utility.

7. Toppings:

Both profiles prefer mushroom toppings, showing a consistent preference.

8. Spicy:

Both profiles prefer extra spicy, indicating a preference for spiciness.

Conclusion:

The conjoint analysis reveals that the profile with the highest utility score (Oven Story, $4.00, 100g, thick crust, Mozzarella cheese, large size, mushroom toppings, extra spicy) is not entirely aligned with the preferred levels for some attributes (brand, price, and size). However, the consistency in preferences

for weight, crust, cheese, toppings, and spiciness suggests these attributes are highly valued by the consumer.

The discrepancy between the highest utility profile and the preferred levels of brand, price, and size indicates that the consumer values the overall combination of attributes more than individual preferred levels.

RESULTS AND INTERPRETATION

Perform Principal Component Analysis and Factor Analysis to identify data dimensions. [Survey.csv]

1. Code:

```
library(GPArotation)
pca <- principal(sur_int,5,n.obs =162, rotate ="promax")
pca
```

Factor Analysis:

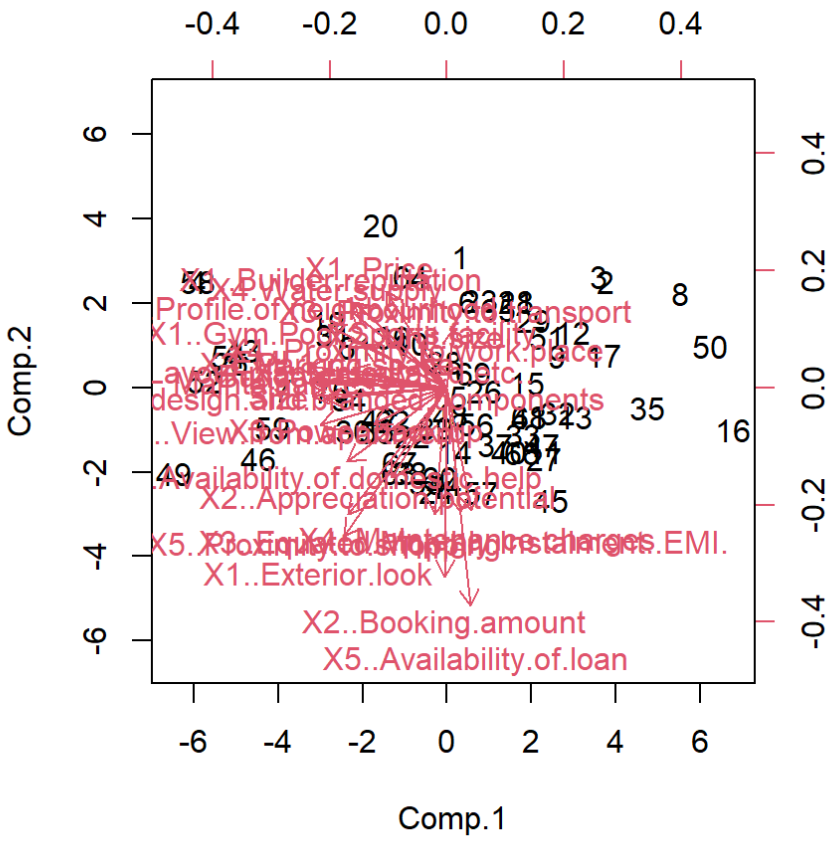
```
survey_df<-read.csv('Survey.csv',header=TRUE)
sur_int=survey_df[,20:46]


#Factor Analysis
factor_analysis<-fa(sur_int,nfactors = 4,rotate = "varimax")
names(factor_analysis)
print(factor_analysis$loadings,reorder=TRUE)
fa.diagram(factor_analysis)
```

2. Result:

```
## Principal Components Analysis
## Call: principal(r = sur_int, nfactors = 5, rotate = "promax", n.obs = 162)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                                        RC1   RC5   RC2   RC4   RC3   h2
## X3..Proximity.to.transport           -0.07  0.06  0.11 -0.17  0.77 0.58
## X4..Proximity.to.work.place           0.31 -0.46  0.11  0.82 -0.09 0.65
## X5..Proximity.to.shopping             0.06  0.64  0.25  0.19 -0.12 0.66
## X1..Gym.Pool.Sports.facility          0.05  0.49 -0.16  0.20  0.23 0.45
## X2..Parking.space                     0.13  0.50 -0.18  0.19 -0.01 0.46
## X3.Power.back.up                      0.06  0.23  0.11  0.69 -0.07 0.64
## X4.Water.supply                       0.38  0.24  0.01  0.10  0.63 0.72
```

```
## X5.Security                                   -0.16  0.91 -0.18 -0.14  0.33 0.74
## X1..Exterior.look                              0.31  0.53  0.24 -0.11 -0.36 0.78
## X2..Unit.size                                  0.49 -0.14 -0.17 -0.51 -0.15 0.45
## X3..Interior.design.and.branded.components     0.45  0.39 -0.06  0.12 -0.10 0.60
## X4..Layout.plan..Integrated.etc..              0.65  0.02 -0.04  0.24 -0.21 0.59
## X5..View.from.apartment                        0.33  0.64 -0.05 -0.07 -0.08 0.71
## X1..Price                                      0.61 -0.26  0.04  0.08  0.48 0.54
## X2..Booking.amount                             0.09  0.00  0.64 -0.06 -0.12 0.47
## X3..Equated.Monthly.Instalment..EMI.          -0.03 -0.05  0.68  0.01  0.42 0.53
## X4..Maintenance.charges                       -0.13  0.02  0.42 -0.09  0.01 0.22
## X5..Availability.of.loan                      -0.01 -0.20  0.89  0.24  0.00 0.76
## X1..Builder.reputation                         0.86 -0.18 -0.09 -0.17  0.18 0.67
## X2..Appreciation.potential                     0.41  0.08  0.37 -0.21  0.08 0.35
## X3..Profile.of.neighbourhood                   0.43  0.47 -0.21 -0.16  0.25 0.67
## X4..Availability.of.domestic.help              0.06  0.83 -0.05 -0.34 -0.11 0.71
## Time                                          -0.08  0.23  0.46 -0.05  0.16 0.27
## Size                                           0.74  0.20  0.07  0.04  0.02 0.76
## Budgets                                        0.81  0.16  0.05  0.03  0.05 0.81
## Maintainances                                  0.72  0.20  0.07  0.16  0.08 0.79
## EMI.1                                          0.77  0.13 -0.02  0.18 -0.04 0.81
##                                                 u2 com
## X3..Proximity.to.transport                    0.42 1.2
## X4..Proximity.to.work.place                   0.35 2.0
## X5..Proximity.to.shopping                     0.34 1.6
## X1..Gym.Pool.Sports.facility                  0.55 2.1
## X2..Parking.space                             0.54 1.7
## X3.Power.back.up                              0.36 1.3
## X4.Water.supply                               0.28 2.0
## X5.Security                                   0.26 1.5
## X1..Exterior.look                             0.22 3.1
## X2..Unit.size                                 0.55 2.6
## X3..Interior.design.and.branded.components    0.40 2.3
## X4..Layout.plan..Integrated.etc..             0.41 1.5
## X5..View.from.apartment                       0.29 1.6
## X1..Price                                     0.46 2.3
## X2..Booking.amount                            0.53 1.1
## X3..Equated.Monthly.Instalment..EMI.          0.47 1.7
## X4..Maintenance.charges                       0.78 1.3
```

```
## X5..Availability.of.loan                    0.24 1.3
## X1..Builder.reputation                      0.33 1.3
## X2..Appreciation.potential                  0.65 2.7
## X3..Profile.of.neighbourhood                0.33 3.2
## X4..Availability.of.domestic.help           0.29 1.4
## Time                                        0.73 1.9
## Size                                        0.24 1.2
## Budgets                                     0.19 1.1
## Maintainances                               0.21 1.3
## EMI.1                                       0.19 1.2
##
##                         RC1  RC5  RC2  RC4  RC3
## SS loadings           5.69 4.47 2.42 1.88 1.91
## Proportion Var        0.21 0.17 0.09 0.07 0.07
## Cumulative Var        0.21 0.38 0.47 0.54 0.61
## Proportion Explained  0.35 0.27 0.15 0.12 0.12
## Cumulative Proportion 0.35 0.62 0.77 0.88 1.00
##
##  With component correlations of
##       RC1   RC5   RC2   RC4   RC3
## RC1  1.00  0.50 -0.08  0.16  0.00
## RC5  0.50  1.00  0.08  0.29 -0.06
## RC2 -0.08  0.08  1.00 -0.16 -0.19
## RC4  0.16  0.29 -0.16  1.00  0.09
## RC3  0.00 -0.06 -0.19  0.09  1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  252.24  with prob <  0.11
##
## Fit based upon off diagonal values = 0.95
```

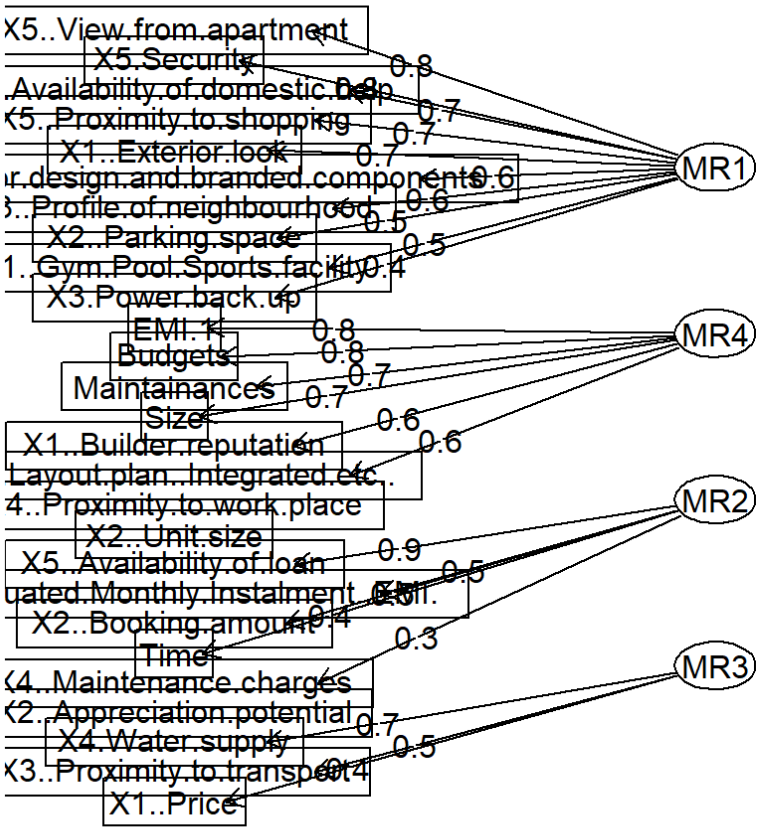


Factor Analysis:

```
Loadings:
##                                                MR1    MR4    MR2    MR3
## X3..Proximity.to.transport                                        0.539
## X4..Proximity.to.work.place                           0.282
## X5..Proximity.to.shopping                      0.691  0.143  0.288
## X1..Gym.Pool.Sports.facility                   0.467  0.164 -0.125  0.232
## X2..Parking.space                              0.520  0.249 -0.143
## X3.Power.back.up                               0.362  0.238
## X4.Water.supply                                0.347  0.361         0.660
## X5.Security                                    0.753 -0.101         0.385
## X1..Exterior.look                              0.671  0.294  0.302 -0.344
## X2..Unit.size                                         0.150 -0.108
## X3..Interior.design.and.branded.components     0.612  0.432
## X4..Layout.plan..Integrated.etc..              0.405  0.554
## X5..View.from.apartment                        0.756  0.329
## X1..Price                                             0.407         0.438
## X2..Booking.amount                                           0.516 -0.138
```

```
## X3..Equated.Monthly.Instalment..EMI.                      0.520  0.249
## X4..Maintenance.charges                           -0.141  0.303
## X5..Availability.of.loan                  -0.146          0.872
## X1..Builder.reputation                     0.204  0.578 -0.157  0.234
## X2..Appreciation.potential                 0.231  0.228  0.244
## X3..Profile.of.neighbourhood               0.590  0.352 -0.204  0.322
## X4..Availability.of.domestic.help          0.741
## Time                                       0.111          0.362
## Size                                       0.510  0.701
## Budgets                                    0.476  0.769          0.109
## Maintainances                              0.509  0.728          0.146
## EMI.1                                      0.488  0.775
##
##                  MR1   MR4   MR2   MR3
## SS loadings    5.386 4.022 1.908 1.554
## Proportion Var 0.199 0.149 0.071 0.058
## Cumulative Var 0.199 0.348 0.419 0.477
```

Factor Analysis



3. Interpretation:

- RC1: The variables X1 (Builder reputation), Size, Budgets, Maintenance, and EMI 1 have high loadings, indicating a strong relationship with the first principal component.
- RC2: The variables X2 (Booking amount), X3 (Equated Monthly Installment - EMI), and X5 (Availability of loan) have high loadings, suggesting this component is primarily associated with financial aspects.
- RC3: The variables X3 (Proximity to transport) and X4 (Water supply) have high loadings, indicating a focus on infrastructure and utilities in this component.
- RC4: The variables X4 (Proximity to workplace) and X3 (Power backup) have high loadings, pointing to a relationship with proximity to work and backup facilities.
- RC5: The variables X5 (Security) and X1 (Exterior look) have high loadings, suggesting this component is related to security and aesthetics.

Factor Analysis:

Factor Loadings:

Factor loadings reflect the correlation between the observed variables and the underlying factors. The four factors (MR1, MR2, MR3, MR4) can be interpreted as follows based on their loadings:

MR1:

- High loadings on "X5..Security," "X1..Exterior look," "X5..View from apartment," "X4..Availability of domestic help," "Size," "Budgets," and "Maintenance."
- This factor can be interpreted as representing "Safety and Aesthetic Appeal."

MR2:

- High loadings on "X5..Availability of loan," "X2..Booking amount," and "X3..Equated Monthly Installment (EMI)."
- This factor appears to capture "Financial Aspects."

MR3:

- High loadings on "X4..Water supply," "X5..Security," and "X1..Price."
- This factor could be interpreted as representing "Basic Amenities and Security."

MR4:

- High loadings on "Size," "Budgets," "Maintenance," and "EMI.1."
- This factor might represent "Space and Maintenance Costs."

Communalities: Communalities indicate how much of the variance in each variable is explained by the factors. Higher communalities (closer to 1) suggest that the variable is well explained by the factors.

- Variables such as "X5..Security," "X1..Exterior look," "Size," "Budgets," "Maintenance," and "EMI.1" have high communalities, indicating they are well explained by the factors.
- Variables such as "X4..Proximity to workplace," "X2..Unit size," and "X3..Power backup" have lower communalities, suggesting they are not as well explained by the factors.

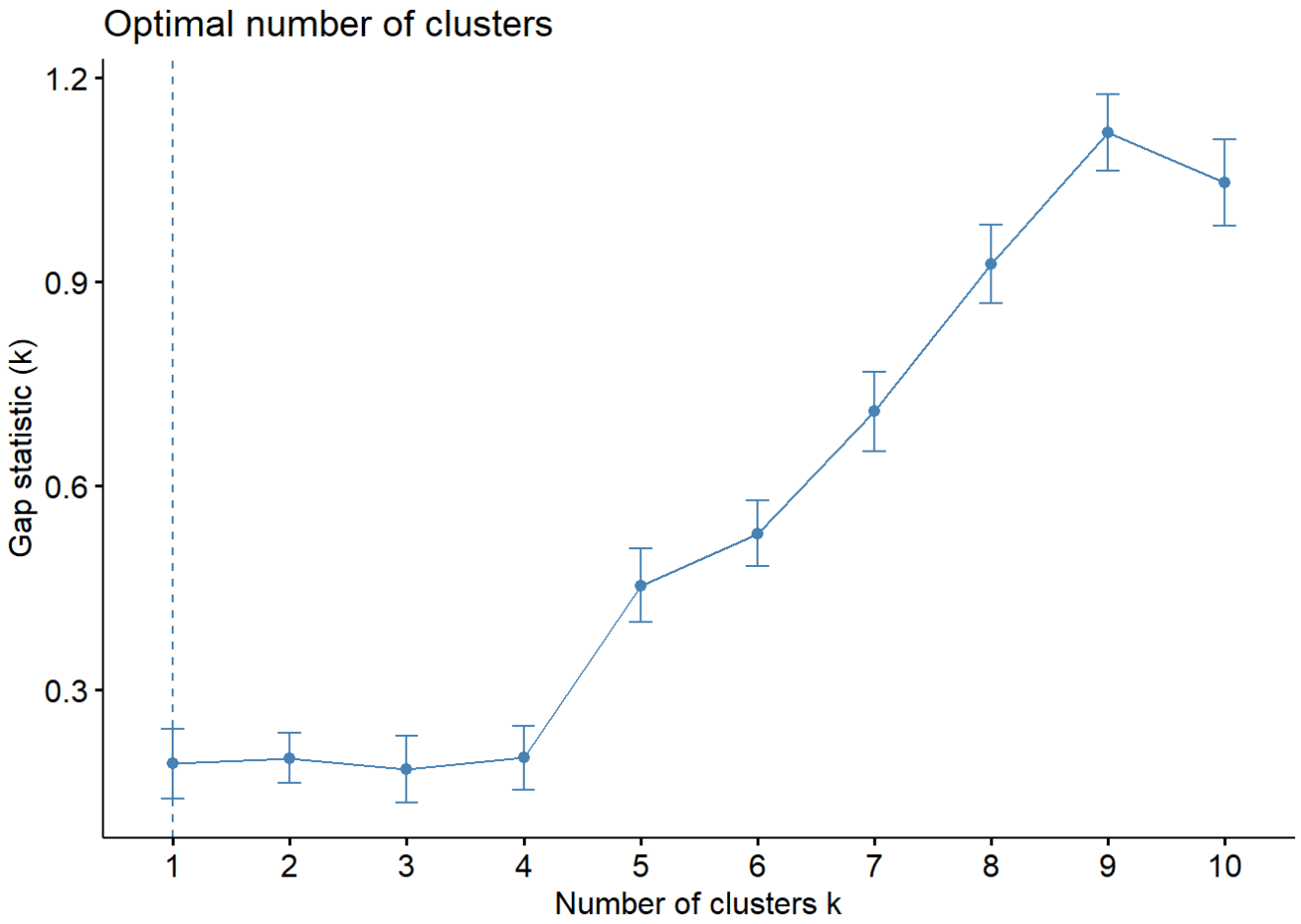
Conduct Cluster Analysis to characterize respondents based on background variables. [Survey.csv]

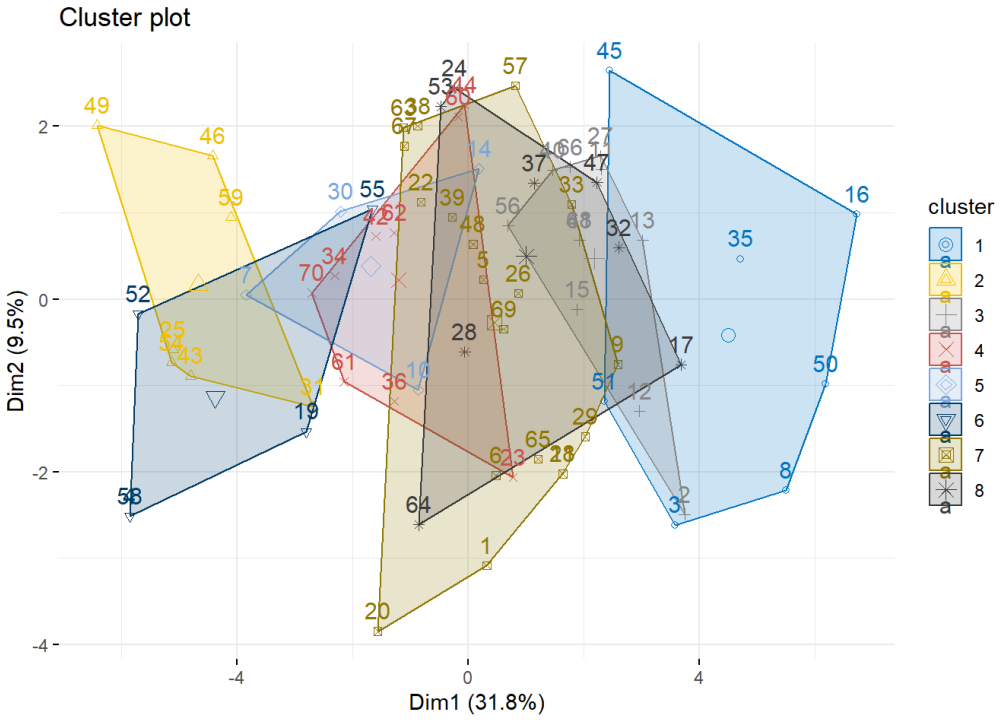
1. Code:

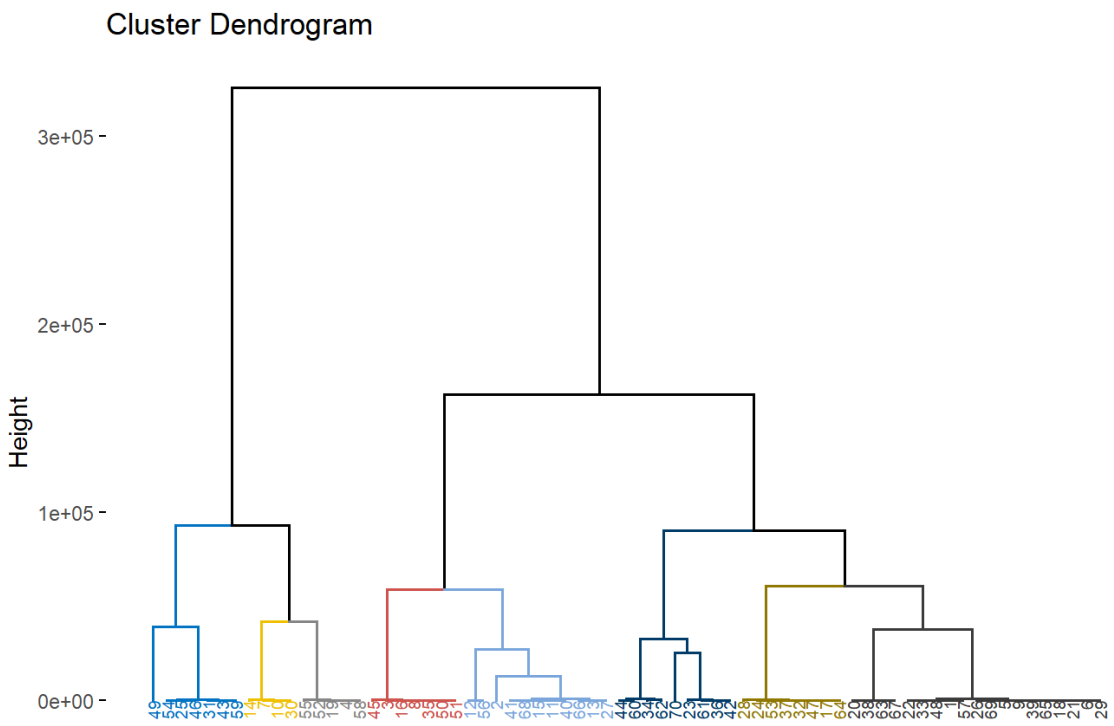
```
fviz_nbclust(sur_int,kmeans,method = "gap_stat")
set.seed(123)
km.res<-kmeans(sur_int,8,nstart = 25)
fviz_cluster(km.res,data=sur_int,palette="jco",
             ggtheme = theme_minimal())
res.hc <- hclust(dist(sur_int), method = "ward.D2")
fviz_dend(res.hc,cex=0.5,k=8,palette = "jco")
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" inst
ead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

2. Result:


Optimal number of clusters
Gap statistic (k)
1.2
0.9
0.6
0.3
1
2
3
4
5
6
7
8
9
10
Number of clusters k


Cluster plot
Dim2 (9.5%)
2
0
-2
-4
-4
0
4
Dim1 (31.8%)
cluster
1
2
3
4
5
6
7
8



3. Interpretation:

This plot is used to determine the optimal number of clusters (k) for k-means clustering. The x-axis represents the number of clusters, while the y-axis shows the gap statistic. The optimal number of clusters is typically the one with the highest gap statistic value, which in this case seems to be k = 8. This plot combines a heatmap and a dendrogram to show the hierarchical clustering results. The dendrogram on the left indicates the hierarchical structure of the observations, grouped into eight clusters. The heatmap shows the values of different variables for each observation, with color intensity representing the magnitude of the values.

- Optimal number of clusters (k): 8
- Cluster visualization: The clusters are well-separated in the cluster plot, and the heatmap/dendrogram provides detailed hierarchical structure.

The plot shows the clusters in different colors and shapes. Each cluster is represented by a polygon that encloses the data points belonging to that cluster. Here's a breakdown:

- Cluster 1: Represented in dark blue with circles. This cluster is quite distinct and separated from the others, indicating well-defined data points.
- Cluster 2: Represented in yellow with triangles. It spans a large area, indicating variability within the cluster.

- Cluster 3: Represented in light grey with squares. This cluster overlaps significantly with other clusters, suggesting less distinct separation.
- Cluster 4: Represented in red with crosses. This cluster is relatively small and overlaps with clusters 3, 5, and 6.
- Cluster 5: Represented in light blue with diamonds. This cluster also overlaps significantly with other clusters.
- Cluster 6: Represented in dark grey with upward triangles. This cluster is similar to cluster 4 in size and position.
- Cluster 7: Represented in black with X marks. This cluster overlaps with clusters 3 and 5, indicating less distinct boundaries.
- Cluster 8: Represented in light grey with asterisks. This cluster overlaps significantly with others, showing less distinct separation.

Overlapping Clusters

- Clusters 3, 4, 5, 6, and 7 have significant overlap, indicating that these data points are not well-separated and may share similarities.
- Cluster 1 is the most distinct with minimal overlap, indicating well-defined separation from other clusters.
- Cluster 2 has some overlap but still maintains a relatively distinct boundary.

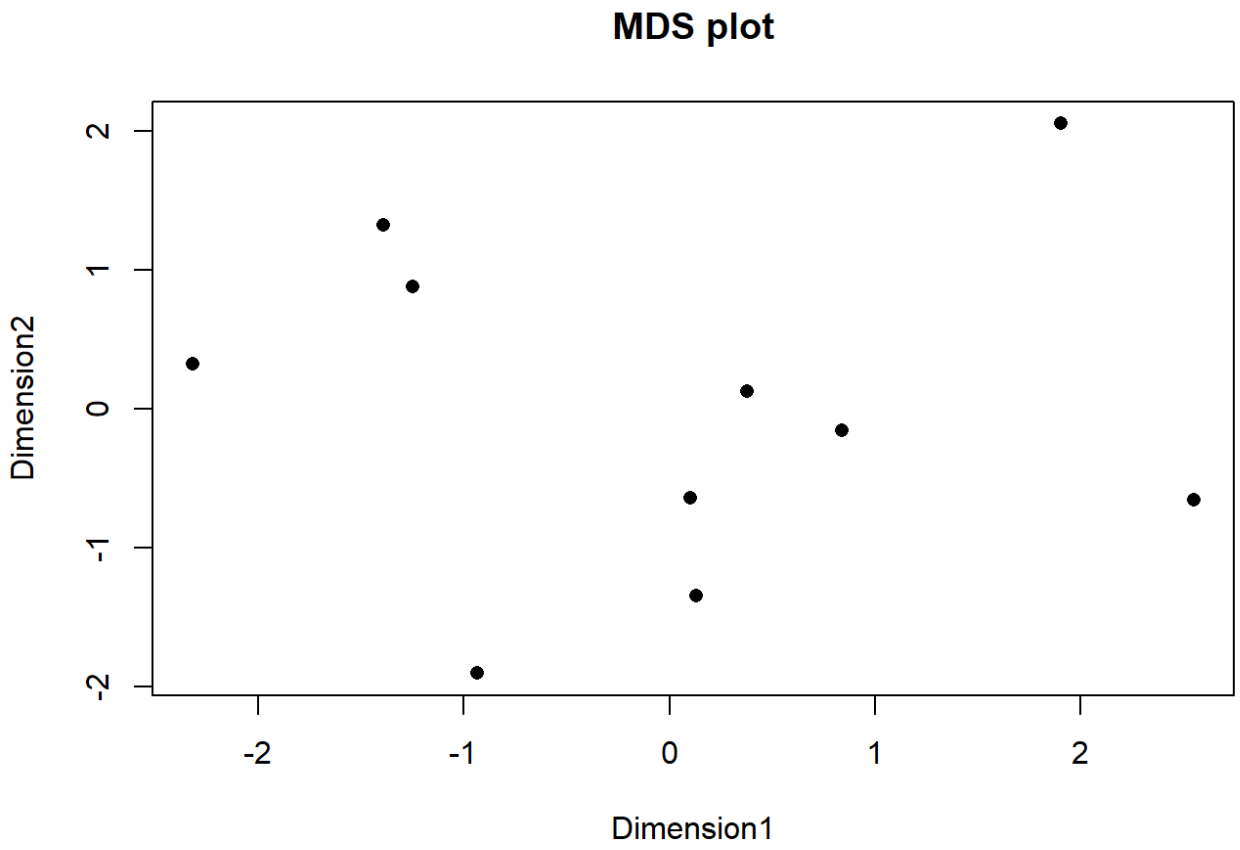
The plot provides insights into the clustering structure of the data. Cluster 1 is the most well-separated, while other clusters show significant overlap, indicating potential challenges in clearly distinguishing these clusters based on the first two principal components alone.

Apply Multidimensional Scaling and interpret the results [icecream.csv]

1. Code:

```
#C) Do multidimensional scaling and interpret the results.


icecream_df<-read.csv('Icecream.csv',header=TRUE)
ice<-subset(icecream_df,select = -c(Brand))
distance_matrix<-dist(ice)
mds_result<-cmdscale(distance_matrix,k=2)


plot(mds_result[,1],mds_result[,2],pch=16,xlab="Dimension1",ylab="Dimension2",main=
"MDS plot")
```

2. Result:



3. Interpretation:

The MDS plot provides a visual representation of the similarities and differences among ice cream brands based on scaled numerical features.

- Bottom Left Quadrant: There is a small cluster of points in the bottom left quadrant. These brands have similar feature profiles, suggesting they may compete closely in the market or share common characteristics.
- Upper Left Quadrant: There is a single point in the upper left quadrant, indicating a brand with a unique feature profile compared to others. This brand is quite distinct and may cater to a niche market.
- Center to Center-Right Quadrant: A spread of points from the center to the right indicates several brands with somewhat similar but still distinct features. These brands are not closely clustered, suggesting moderate differences in their features.
- Upper Right Quadrant: There is a single point in the upper right quadrant, indicating another

brand with unique characteristics that set it apart from the others.

- Bottom Right Quadrant: There is a small cluster of points in the bottom right quadrant, indicating brands with similar features that may compete directly with each other.

Brands positioned away from the clusters, such as those in the upper left and upper right quadrants, indicate unique feature sets. These brands are likely differentiating themselves in the market with distinct unique selling propositions. The spread of points across the plot suggests that brands are catering to different market segments or emphasizing different features to appeal to various customer preferences.

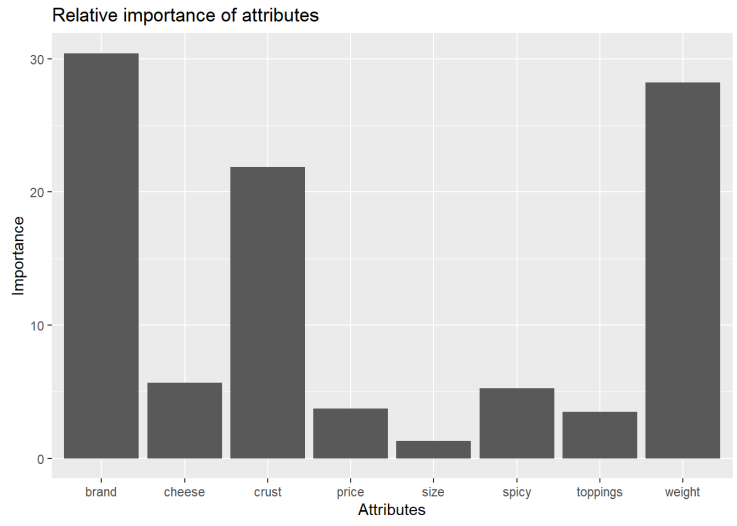
Conjoint Analysis [pizza_data.csv]

1. Code:

```
model <- 'ranking ~ brand + price + weight + crust + cheese + size + toppings + spicy'
model_fit <- lm(model, data=df)
summary(model_fit)
conjoint_attributes <- c('brand', 'price', 'weight', 'crust', 'cheese', 'size', 'toppings', 'spicy')

level_name <- list()
part_worth <- list()
part_worth_range <- c()
important_levels <- list()
end <- 1

for (item in conjoint_attributes) {
  nlevels <- length(unique(df[[item]]))
  level_name[[item]] <- unique(df[[item]])

  begin <- end
  end <- begin + nlevels - 1

  new_part_worth <- coef(model_fit)[begin:end]
  new_part_worth <- c(new_part_worth, (-1) * sum(new_part_worth))
  important_levels[[item]] <- which.max(new_part_worth)
  part_worth[[item]] <- new_part_worth
```

```
  part_worth_range <- c(part_worth_range, max(new_part_worth) - min(new_part_worth)
)
}

cat("-----------------------------------------------------------------\n")
print(part_worth)
print(important_levels)
attribute_importance <- round(100 * part_worth_range / sum(part_worth_range), 2)
print(attribute_importance)
part_worth_dict <- list()
attrib_level <- list()
for (i in seq_along(conjoint_attributes)) {
  item <- conjoint_attributes[i]
  cat("Attribute :", item, "\n")
  cat("    Relative importance of attribute ", attribute_importance[i], "\n")
  cat("    Level wise part worths: \n")
  for (j in seq_along(level_name[[item]])) {
    cat("          ", level_name[[item]][j], ":", part_worth[[item]][j], "\n")
    part_worth_dict[[level_name[[item]][j]]] <- part_worth[[item]][j]
    attrib_level[[item]] <- level_name[[item]]
  }
}
print(df[which.max(utility),])
```

2. Result:



```
brand price weight crust  cheese    size toppings spicy ranking utility
## 16 Dominos $3.00   200g thick Cheddar regular mushroom extra      14   7.375
## $Dominos
## (Intercept)
##      17.375
##
## $`Pizza hut`
## brandOnesta
## 1.02558e-15
##
## $Onesta
## brandOven Story
##           -0.25
##
## $`Oven Story`
## brandPizza hut
##           0.25
##
## $`$1.00`
## brandPizza hut
##           0.25
##
## $`$3.00`
## price$2.00
##      -0.75
##
## $`$4.00`
## price$3.00
##      -0.75
##
## $`$2.00`
## price$4.00
##       -1.5
##
## $`100g`
## price$4.00
##       -1.5
```

```
##
## $`200g`
## weight200g
##         -3
##
## $`400g`
## weight300g
##      -6.25
##
## $`300g`
## weight400g
##     -10.75
##
## $thin
## weight400g
##     -10.75
##
## $thick
## crustthin
##      -3.5
##
## $Mozzarella
## crustthin
##      -3.5
##
## $Cheddar
## cheeseMozzarella
##              0.5
##
## $regular
## cheeseMozzarella
##              0.5
##
## $large
## sizeregular
##         0.5
##
## $paneer
```

```
## sizeregular
##         0.5
##
## $mushroom
## toppingspaneer
##          -2.25
##
## $normal
## toppingspaneer
##          -2.25
##
## $extra
## spicynormal
##        -1.5
```

3. Interpretation:

- Brand: The most important attribute, with an importance value of 30. This indicates that the brand of the pizza significantly influences consumer choice.
- Weight: The second most important attribute, with an importance value close to 30. This suggests that the weight of the pizza is almost as crucial as the brand.
- Crust: The third most important attribute, with an importance value of around 20. The type of crust is also a significant factor in consumer preference.
- Cheese: This attribute has a moderate importance value of around 10. The type of cheese used affects consumer choice, but not as much as brand, weight, or crust.
- Size, Spicy, and Toppings: These attributes have relatively low importance values, indicating they are less critical in determining consumer preferences compared to the other attributes.
- Price: Surprisingly, price has the lowest importance value, suggesting that consumers may prioritize other attributes over the cost when choosing a pizza.

The part-worth utilities (also known as attribute levels) reflect the relative preferences for different levels of each attribute. In conjoint analysis, these utilities help in understanding the impact of each attribute level on the overall preference for a product.

- Dominos has the highest positive utility (17.375), indicating a strong preference.
- Pizza Hut serves as the base level for brand comparison.
- Onesta and Oven Story have negative utilities, indicating a lesser preference compared to the

base level (Pizza Hut).

- The lowest price level ($1.00) has a positive utility, showing a preference for cheaper options.
- Heavier pizzas (300g and 400g) have significantly negative utilities, indicating a strong preference for lighter pizzas.
- Thin crust has a negative utility, suggesting a preference for thick crust (since thick crust serves as the base level).
- Mozzarella cheese, regular size, paneer topping, and normal spiciness levels have specific utilities indicating varying preferences among these attributes.

The utilities provide insights into which attributes and levels are most and least preferred by consumers, guiding product development and marketing strategies.