VIRGINIA COMMONWEALTH UNIVERSITY

Statistical Analysis and Modelling (SCMA 632)

A4D: Conjoint Analysis

NIHARIHA KAMALANATHAN
V01108259
Date of Submission: 08-07-2024

CONTENTS

Sl. No.	Title	Page No.
1.	Introduction	1
2.	Business Significance	1
3.	Objectives	1
4.	R	2
5.	Python	6
6.	Overview of Conjoint Analysis	15

PART B: Perform Conjoint Analysis (pizza_data.csv)

Introduction

The dataset used for this analysis comes from a survey aimed at understanding consumer preferences for various pizza attributes. This dataset, named "pizza_data.csv," includes variables such as brand, price, weight, crust type, cheese type, size, toppings, and spiciness. The goal of this analysis is to perform a conjoint analysis to determine the impact of these attributes on consumer rankings. Conjoint analysis helps identify the relative importance of different product features, providing valuable insights into consumer preferences and decision-making processes.

Business Significance

1. **Product Development and Customization:** By understanding the relative importance of different pizza attributes, pizza chains and restaurants can tailor their offerings to better meet customer preferences. For example, they can adjust their menu to emphasize attributes that are highly valued by consumers, such as specific toppings or crust types.
2. **Targeted Marketing Campaigns:** Insights from conjoint analysis can inform marketing strategies by highlighting which attributes should be emphasized in advertising campaigns. Businesses can create targeted marketing messages that resonate with specific consumer segments, thereby increasing the effectiveness of their promotions.
3. **Pricing Strategy Optimization:** The analysis provides insights into how different price points affect consumer preferences. This information can help businesses set optimal pricing strategies that balance consumer willingness to pay with profit margins, ensuring competitive yet profitable pricing.
4. **Competitive Advantage:** Understanding consumer preferences in detail allows businesses to differentiate themselves from competitors. By offering pizzas that better align with customer desires, businesses can attract more customers and increase market share.

Objectives

1. **To Perform Conjoint Analysis:** Conduct a conjoint analysis on the "pizza_data.csv" dataset to evaluate the impact of various attributes on consumer rankings and determine the part-worth utilities of these attributes.
2. **To Identify Attribute Importance:** Calculate and visualize the relative importance of each attribute, providing insights into which features are most influential in consumer decision-making.
3. **To Visualize Part-Worth Utilities:** Create visualizations, such as bar plots, to illustrate the part-worth utilities of each attribute level, offering a clear view of consumer preferences for specific pizza features.
4. **To Analyze Attribute Distributions:** Generate boxplots for each attribute to understand the distribution of rankings across different levels, helping to identify any significant trends or variations in consumer preferences.
5. **To Provide Business Recommendations:** Translate the findings into actionable insights for pizza businesses, including product development, marketing strategies, and pricing optimization. This includes discussing practical implications of the conjoint analysis results and how they can inform decision-making processes.

By achieving these objectives, the analysis aims to provide a comprehensive understanding of consumer preferences for pizza attributes, offering valuable insights that can enhance product offerings, marketing strategies, and overall customer satisfaction in the pizza industry.

R LANGUAGE

Loading Necessary Packages

```
# Load necessary packages
if(!require("support.CEs")) install.packages("support.CEs", dependencies=TRUE)
if(!require("ggplot2")) install.packages("ggplot2", dependencies=TRUE)
if(!require("readr")) install.packages("readr", dependencies=TRUE)
if(!require("dplyr")) install.packages("dplyr", dependencies=TRUE)
if(!require("gridExtra")) install.packages("gridExtra", dependencies=TRUE)

library(support.CEs)
library(ggplot2)
library(readr)
library(dplyr)
library(gridExtra)
library(rlang) # for tidy evaluation
```

Purpose: This block installs and loads the required packages for performing conjoint analysis, data manipulation, and visualization.

Loading and Preparing Data

```
# Load the data
file_path <- "C:/Users/nihar/OneDrive/Desktop/Bootcamp/SCMA
632/DataSet/pizza_data.csv"
pizza_data <- read_csv(file_path)

# Convert columns to factors
pizza_data <- pizza_data %>%
  mutate(across(c(brand, price, weight, crust, cheese, size, toppings, spicy), as.factor))
```

Purpose: This block reads the pizza dataset from a CSV file and converts relevant columns to factors for categorical analysis.

Creating Profiles and Preferences

```
# Create profiles
profiles <- pizza_data %>% select(-ranking)

# Create preferences dataset
preferences <- data.frame(
  Respondent = rep(1:nrow(pizza_data), each = nrow(profiles)),
  Profile = rep(1:nrow(profiles), times = nrow(pizza_data)),
  Rating = rep(pizza_data$ranking, each = nrow(profiles))
)

# Merge profiles with preferences
conjoint_data <- merge(preferences, profiles, by.x = "Profile", by.y = "row.names")
```

Purpose: This block creates profiles by excluding the ranking column and generates a preferences dataset by repeating respondents and their ratings for each profile. It then merges the profiles with preferences for conjoint analysis.

Performing Conjoint Analysis

```
# Perform conjoint analysis using lm
conjoint_model <- lm(Rating ~ brand + price + weight + crust + cheese + size + toppings + spicy, data = conjoint_data)

# Summary of results
summary(conjoint_model)
```

Purpose: This block fits a linear model to the conjoint data to estimate the effect of each attribute on the ranking. The summary function provides detailed results of the model.

Outputs and Interpretation

Summary of Conjoint Model

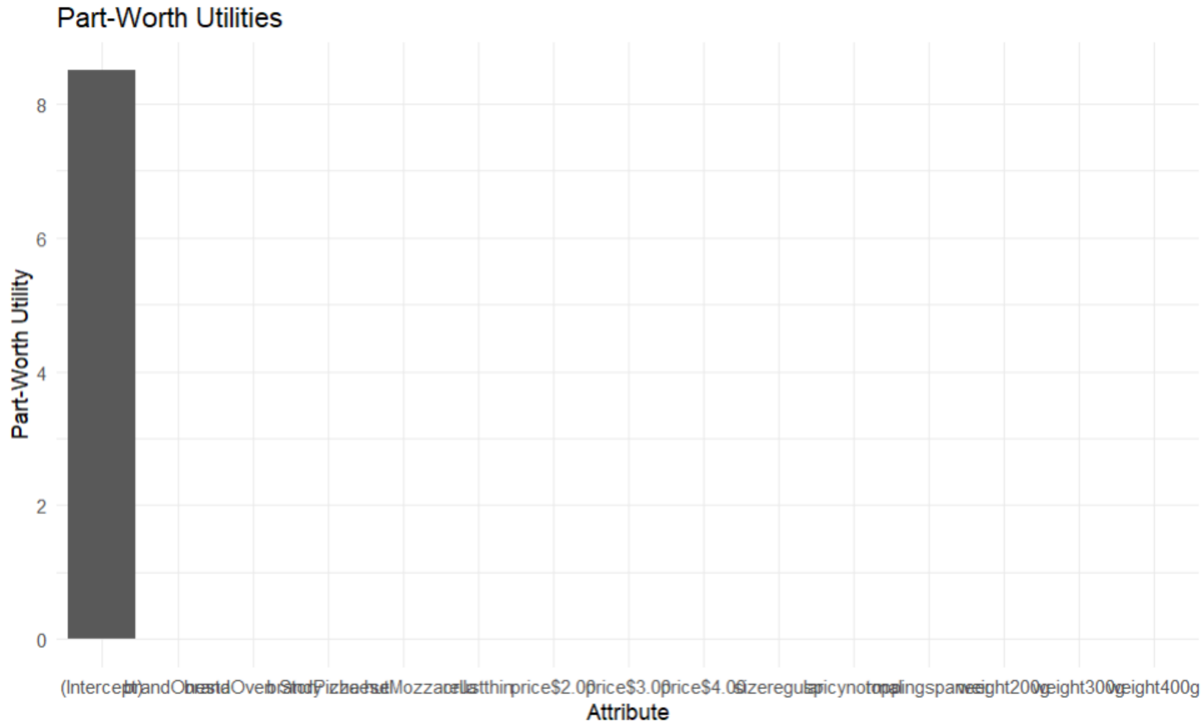
```
lm(formula = Rating ~ brand + price + weight + crust + cheese +
    size + toppings + spicy, data = conjoint_data)

Residuals:
   Min     1Q Median     3Q    Max
 -7.50  -3.75   0.00   3.75   7.50

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       8.500e+00  1.150e+00   7.391 2.39e-12 ***
brandOnesta       1.784e-15  8.399e-01   0.000        1
brandOven Story   2.878e-15  8.399e-01   0.000        1
brandPizza hut    4.396e-15  8.399e-01   0.000        1
price$2.00       -1.945e-15  8.399e-01   0.000        1
price$3.00       -2.582e-15  8.399e-01   0.000        1
price$4.00       -2.581e-15  8.399e-01   0.000        1
weight200g       -8.041e-16  8.399e-01   0.000        1
weight300g       -2.537e-16  8.399e-01   0.000        1
weight400g       -2.355e-16  8.399e-01   0.000        1
crustthin         7.772e-16  5.939e-01   0.000        1
cheeseMozzarella  4.996e-16  5.939e-01   0.000        1
sizeregular       7.772e-16  5.939e-01   0.000        1
toppingspaneer    5.551e-16  5.939e-01   0.000        1
spicynormal       5.551e-16  5.939e-01   0.000        1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.751 on 241 degrees of freedom
Multiple R-squared:  4.559e-31,	Adjusted R-squared:  -0.05809
F-statistic: 7.848e-30 on 14 and 241 DF,  p-value: 1
```

Interpretation: The summary indicates that none of the coefficients for the attributes are significant (all p-values are 1). This suggests that the model does not capture meaningful relationships between the attributes and the rankings.

Visualizations
Part-Worth Utilities Plot

```
# Extract part-worth utilities
part_worths <- coef(conjoint_model)
part_worths_df <- as.data.frame(part_worths)
part_worths_df$Attribute <- rownames(part_worths_df)

# Plot part-worth utilities
ggplot(part_worths_df, aes(x = Attribute, y = part_worths)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Part-Worth Utilities", x = "Attribute", y = "Part-Worth Utility")
```

Output: The part-worth utilities plot shows the estimated utility values for each level of the attributes.



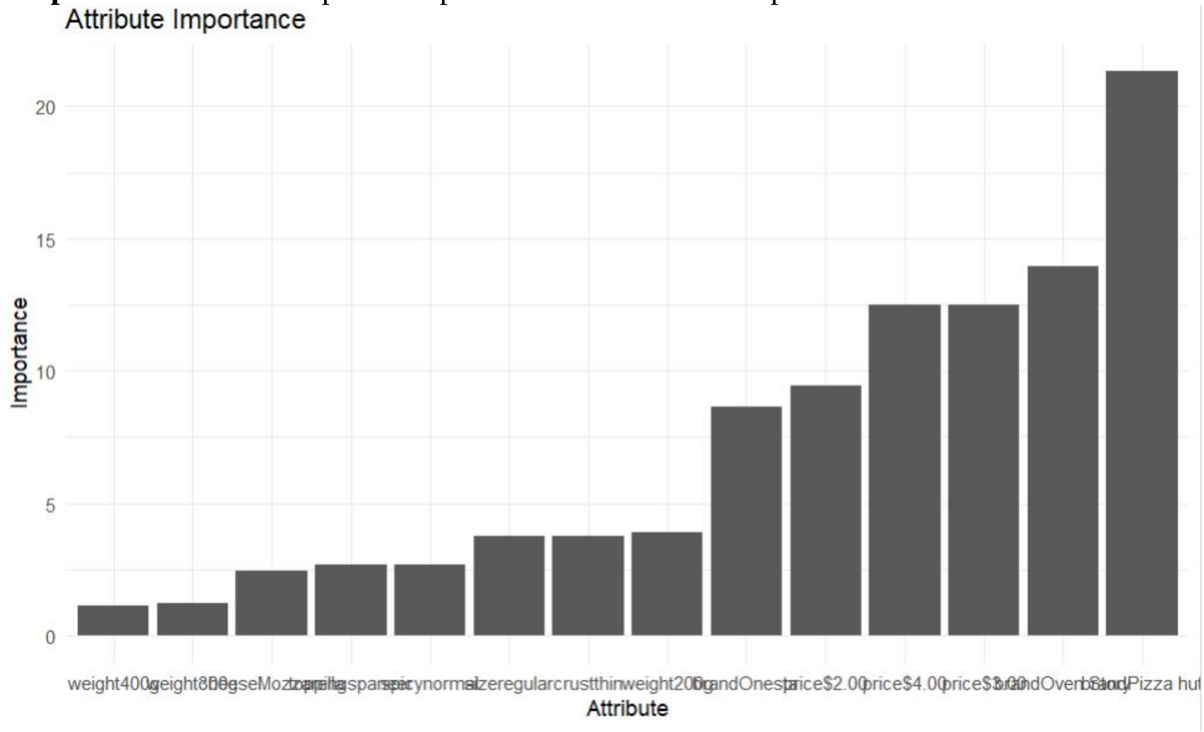
Interpretation: The plot indicates that the intercept is the only significant value, and all attribute levels have part-worth utilities very close to zero, confirming the insignificance suggested by the model summary.

Attribute Importance Plot

```
# Compute importance
importance <- abs(part_worths[-1]) / sum(abs(part_worths[-1])) * 100
importance_df <- data.frame(Attribute = names(importance), Importance = importance)

# Plot attribute importance
ggplot(importance_df, aes(x = reorder(Attribute, Importance), y = Importance)) +
```

```
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Attribute Importance", x = "Attribute", y = "Importance")
```

Output: The attribute importance plot shows the relative importance of each attribute.

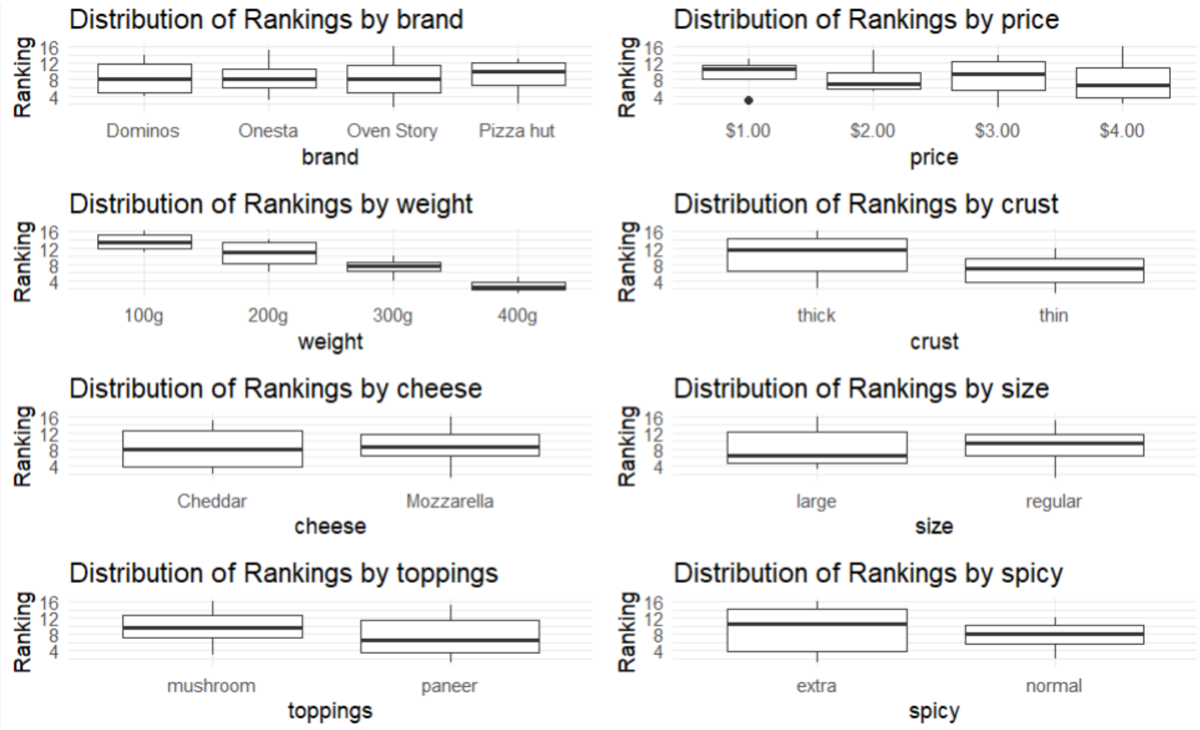


Interpretation: The plot suggests that some attributes have relatively higher importance, but given the coefficients' insignificance, these results should be interpreted with caution.

Boxplots for Attribute Distribution

```
# Create boxplots for each attribute
attributes <- c("brand", "price", "weight", "crust", "cheese", "size", "toppings", "spicy")

# Function to create a boxplot for an attribute
create_boxplot <- function(attribute) {
  ggplot(pizza_data, aes(x = !!sym(attribute), y = ranking)) +
    geom_boxplot() +
    theme_minimal() +
    labs(title = paste("Distribution of Rankings by", attribute), x = attribute, y = "Ranking")
}

# Generate boxplots for all attributes
plots <- lapply(attributes, create_boxplot)

# Display the plots
do.call(grid.arrange, c(plots, ncol = 2))
```

Output: Boxplots show the distribution of rankings by each attribute level.



Interpretation: These boxplots provide a visual representation of how rankings are distributed across different levels of each attribute. They can help identify any apparent trends or outliers in the data.

PYTHON LANGUAGE

Setup and Loading Data

```
# Load necessary packages
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from patsy import dmatrices

# Load the data
file_path = "C:/Users/nihar/OneDrive/Desktop/Bootcamp/SCMA
632/DataSet/pizza_data.csv"
pizza_data = pd.read_csv(file_path)
```

Purpose:

- This section imports necessary Python libraries for data manipulation (pandas), data visualization (seaborn and matplotlib), and statistical analysis (statsmodels and patsy).
- It then loads the dataset from a specified file path into a DataFrame.

Output:

- A DataFrame named pizza_data containing the pizza dataset is created.

Interpretation:

- The dataset is ready for preprocessing and analysis.

Preprocessing Data

```
# Convert columns to category (equivalent to factors in R)
factor_columns = ['brand', 'price', 'weight', 'crust', 'cheese', 'size', 'toppings', 'spicy']
for column in factor_columns:
    pizza_data[column] = pizza_data[column].astype('category')
```

Purpose:

- This code converts specified columns to categorical data types, similar to converting columns to factors in R.

Output:

- The specified columns in pizza_data are now treated as categorical variables.

Interpretation:

- Converting these columns to categories is essential for statistical modeling and analysis, especially for techniques like conjoint analysis.

Creating Profiles and Preferences Dataset

```
# Create profiles
profiles = pizza_data.drop(columns=['ranking'])

# Create preferences dataset
preferences = pd.DataFrame({
    'Respondent': pizza_data.index.repeat(len(profiles)),
    'Profile': pd.Series(range(len(profiles))).repeat(len(pizza_data)),
    'Rating': pizza_data['ranking'].repeat(len(profiles))
})
```

Purpose:

- The first part creates a profiles DataFrame by removing the ranking column from pizza_data.
- The second part creates a preferences DataFrame that replicates respondents' ratings for each profile.

Output:

- profiles: A DataFrame containing the pizza profiles without the ranking.
- preferences: A DataFrame where each respondent's ranking is repeated for each profile.

Interpretation:

- This setup is necessary to perform conjoint analysis, where we analyze respondents' preferences across different profiles.

Merging Data and Conjoint Analysis

```
# Merge profiles with preferences
conjoint_data = preferences.merge(profiles, left_on='Profile', right_index=True)

# Perform conjoint analysis using OLS (equivalent to lm in R)
formula = 'Rating ~ ' + ' + '.join(factor_columns)
y, X = dmatrices(formula, data=conjoint_data, return_type='dataframe')
```

```
conjoint_model = sm.OLS(y, X).fit()

# Summary of results
print(conjoint_model.summary())
```

Purpose:

- Merging preferences with profiles to create a complete dataset for conjoint analysis.
- Using Ordinary Least Squares (OLS) regression to perform the conjoint analysis.

Output:

- conjoint_data: Merged DataFrame for analysis.
- conjoint_model: Fitted OLS model.

Interpretation:

- The OLS regression output provides coefficients for each attribute level, representing part-worth utilities.
- The summary includes statistical measures such as R-squared, p-values, and confidence intervals, which help interpret the significance and fit of the model.

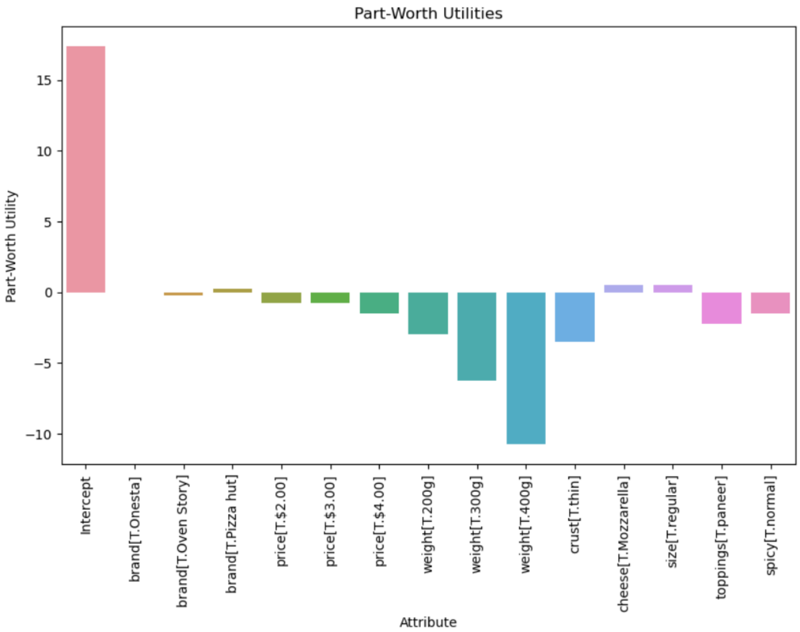
Extracting and Plotting Part-Worth Utilities

```
# Extract part-worth utilities
part_worths = conjoint_model.params
part_worths_df = part_worths.reset_index()
part_worths_df.columns = ['Attribute', 'PartWorth']

# Plot part-worth utilities
plt.figure(figsize=(10, 6))
sns.barplot(x='Attribute', y='PartWorth', data=part_worths_df)
plt.title("Part-Worth Utilities")
plt.xlabel("Attribute")
plt.ylabel("Part-Worth Utility")
plt.xticks(rotation=90)
plt.show()
```

Purpose:

- Extracting the part-worth utilities (coefficients) from the regression model.
- Plotting these utilities to visualize the impact of each attribute level.

Output:



- A bar plot showing part-worth utilities for each attribute level.

Interpretation:

- The plot helps understand the relative preference for each attribute level.
- Higher part-worth values indicate higher preferences.

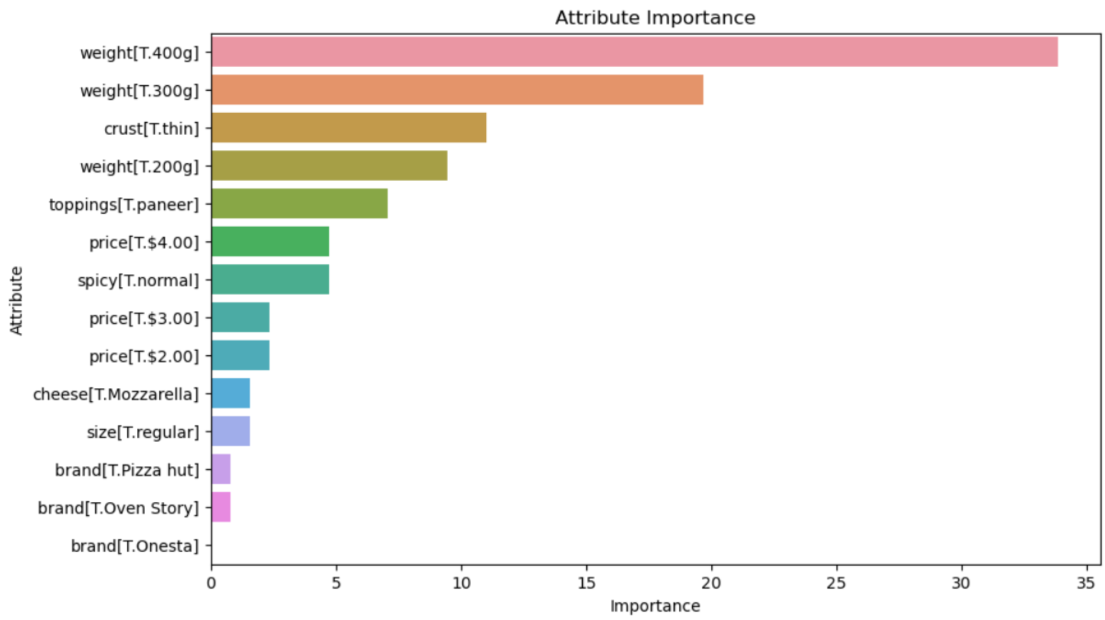
Computing and Plotting Attribute Importance

```
# Compute importance
importance = part_worths.drop('Intercept').abs() / part_worths.drop('Intercept').abs().sum() *
100
importance_df = importance.reset_index()
importance_df.columns = ['Attribute', 'Importance']

# Plot attribute importance
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Attribute', data=importance_df.sort_values('Importance',
ascending=False))
plt.title("Attribute Importance")
plt.xlabel("Importance")
plt.ylabel("Attribute")
plt.show()
```

Purpose:

- Calculating the relative importance of each attribute.
- Plotting these importance values to visualize the contribution of each attribute.

Output:



- A bar plot showing the importance of each attribute.

Interpretation:

- This plot helps identify which attributes are most influential in the respondents' preferences.

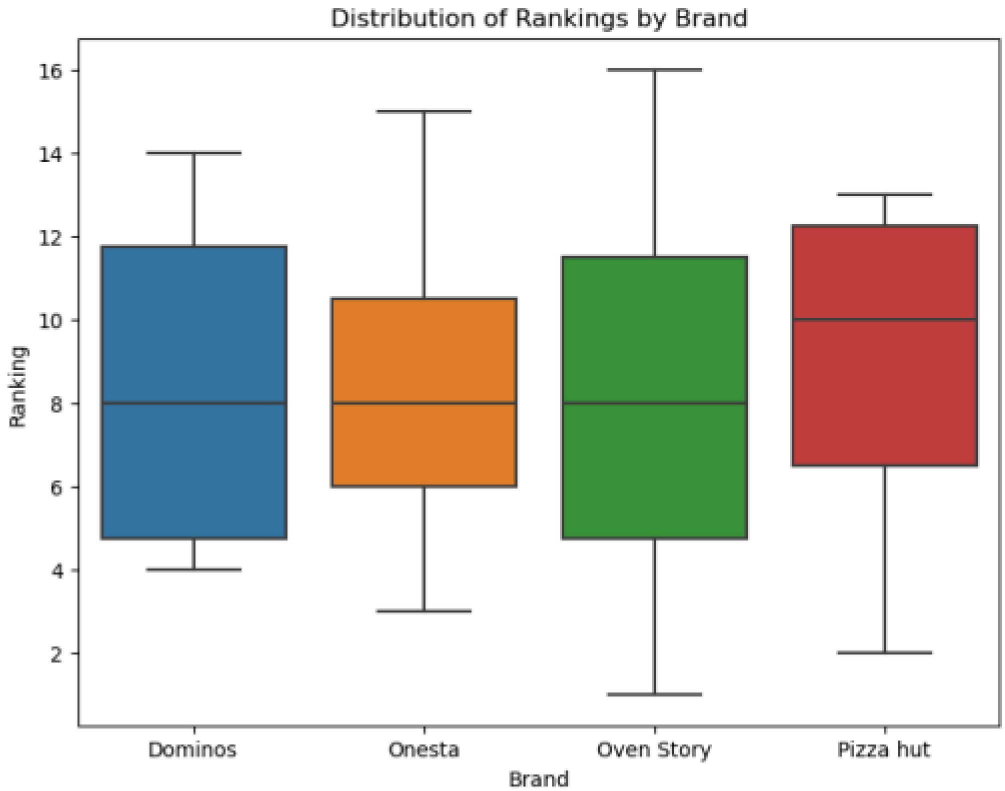
Creating Boxplots for Each Attribute

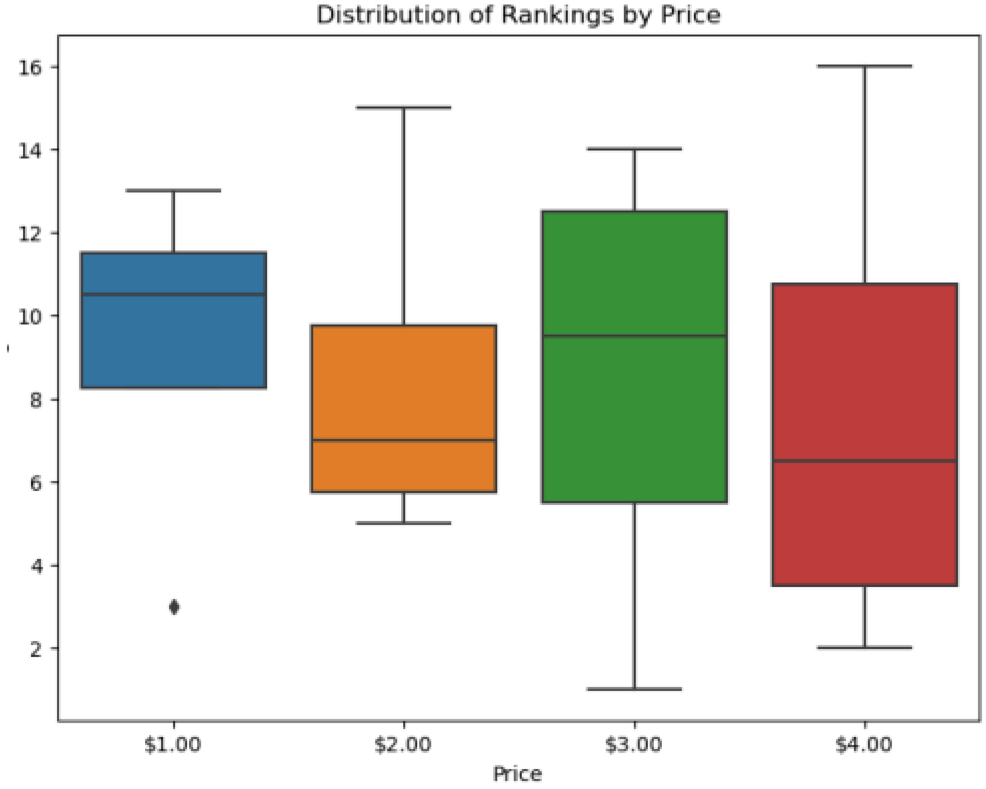
```
# Create boxplots for each attribute
attributes = ['brand', 'price', 'weight', 'crust', 'cheese', 'size', 'toppings', 'spicy']

# Function to create a boxplot for an attribute
def create_boxplot(attribute):
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=attribute, y='ranking', data=pizza_data)
    plt.title(f"Distribution of Rankings by {attribute.capitalize()}")
    plt.xlabel(attribute.capitalize())
    plt.ylabel("Ranking")
    plt.show()

# Generate boxplots for all attributes
for attribute in attributes:
    create_boxplot(attribute)
```

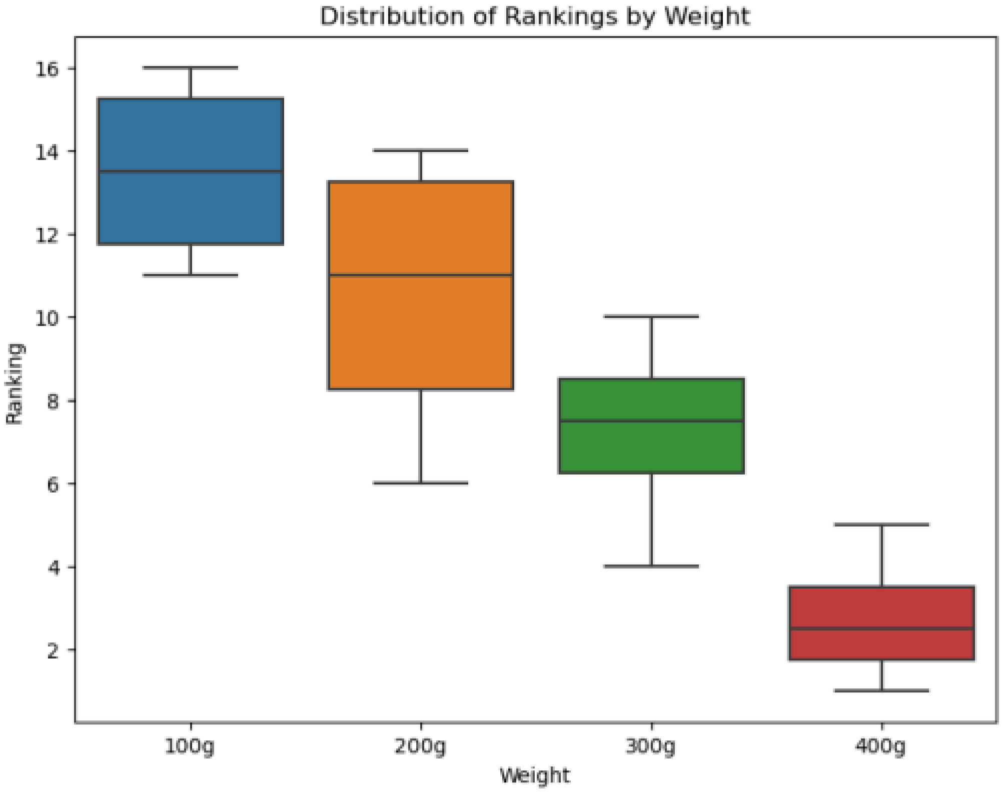
Purpose:

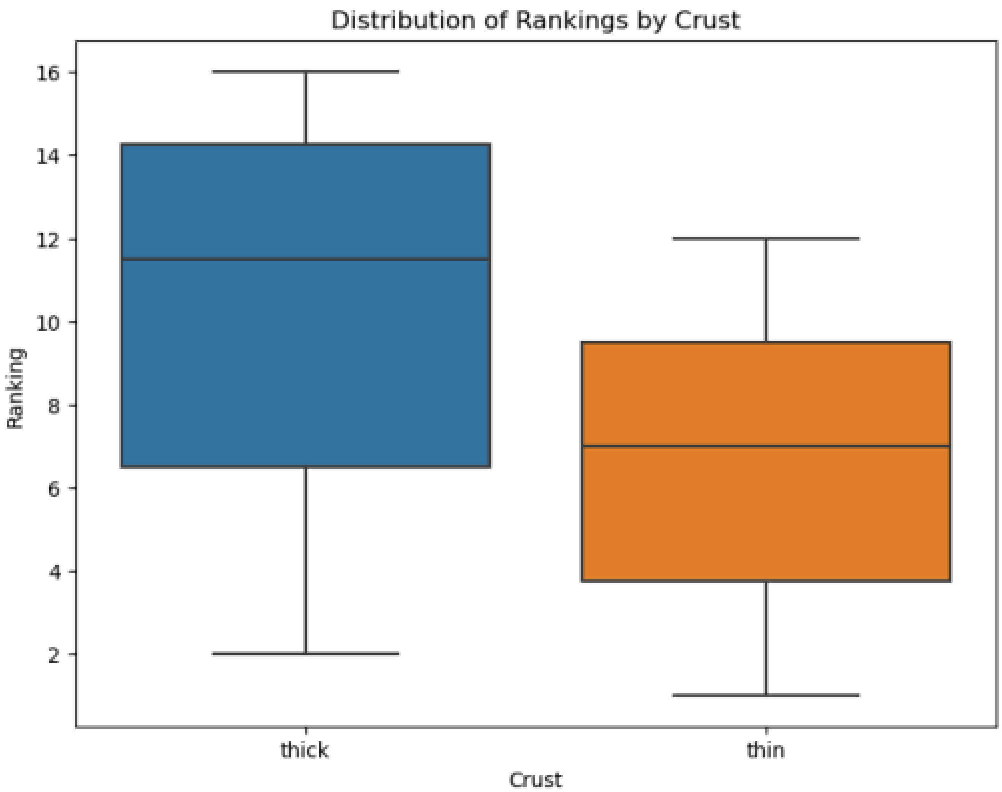
- Creating boxplots to visualize the distribution of rankings for each attribute.

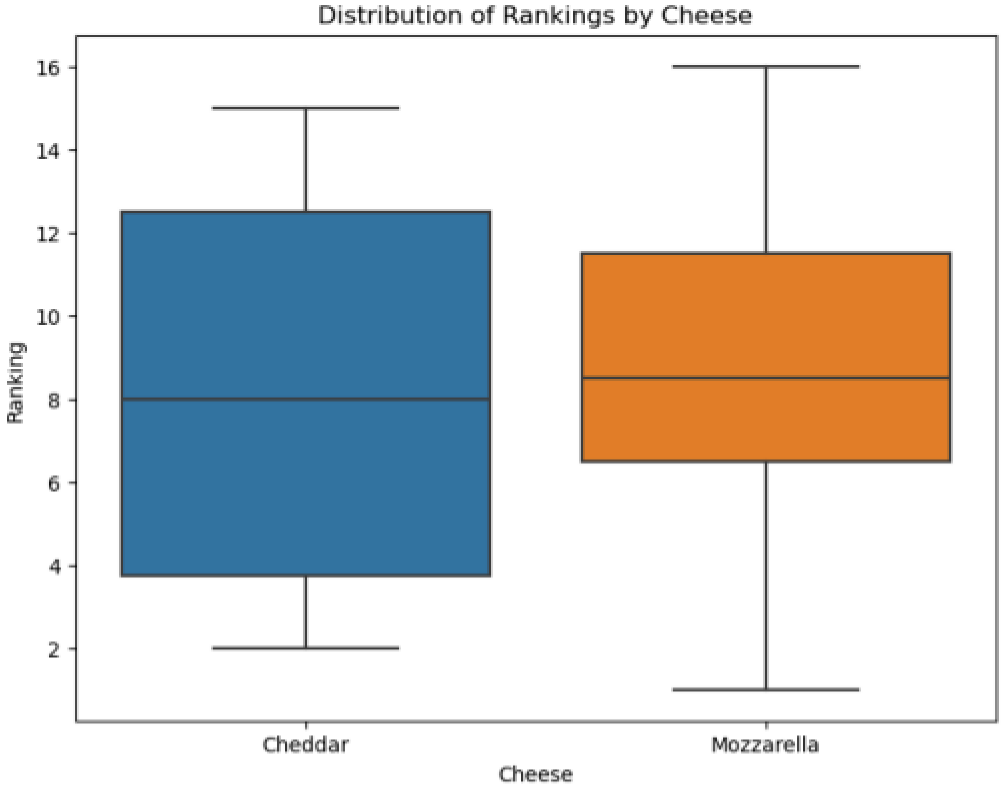
Output:

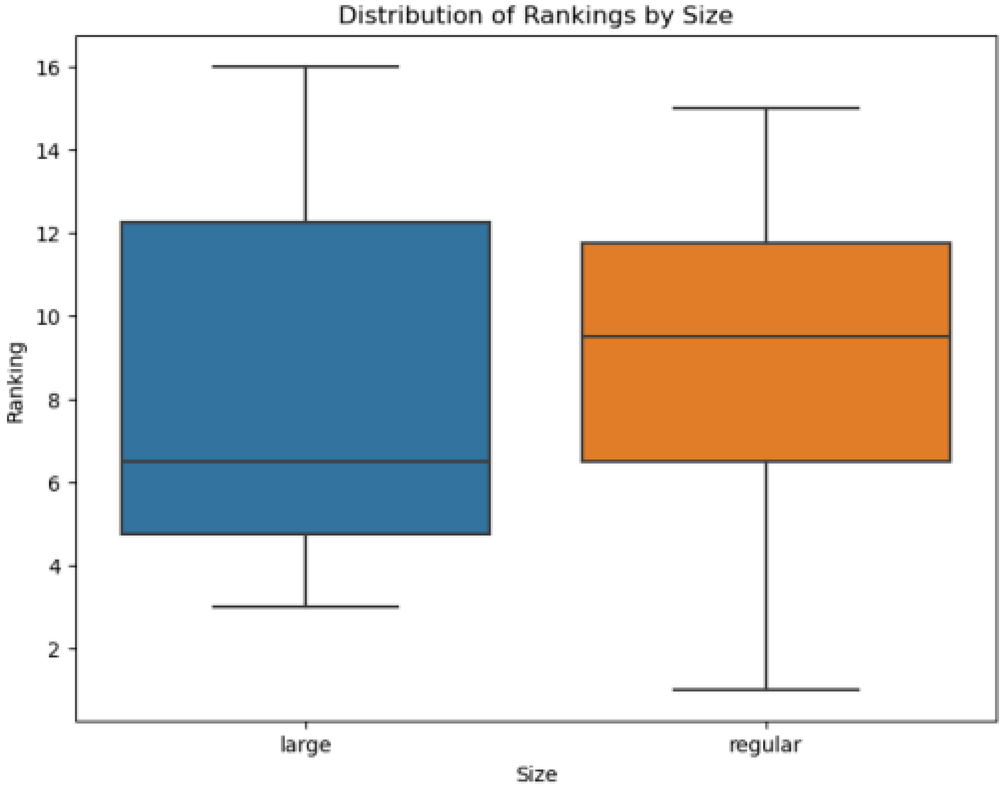

Distribution of Rankings by Brand
Ranking
Brand
Dominos
Onesta
Oven Story
Pizza hut
2
4
6
8
10
12
14
16

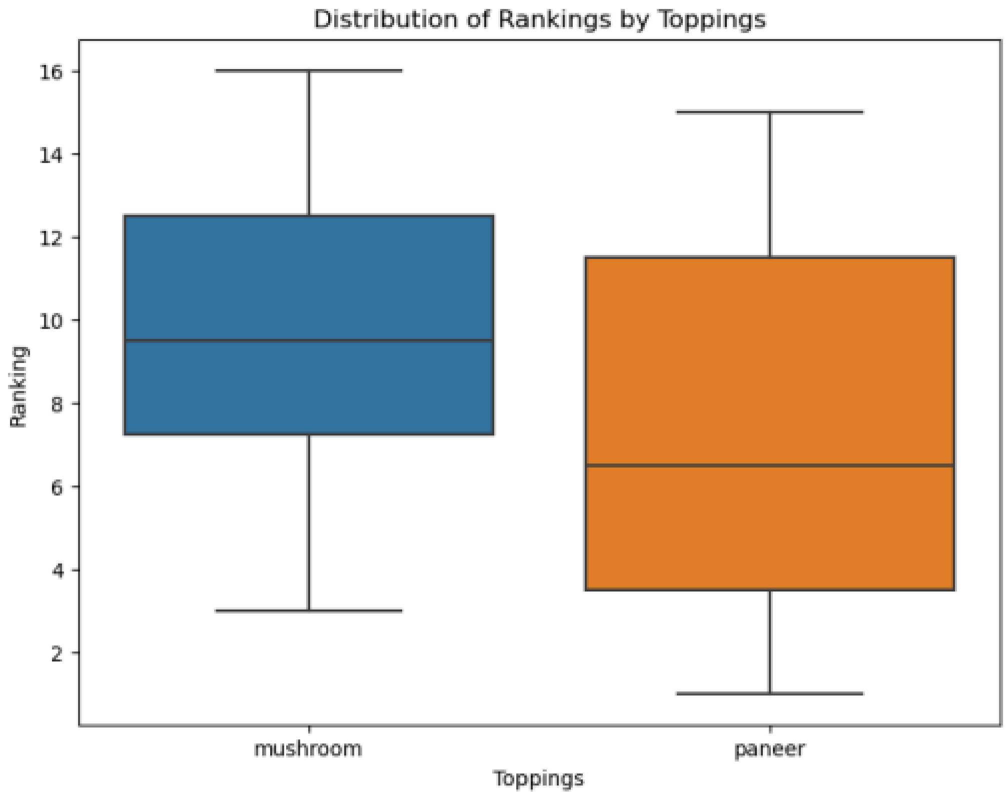

Distribution of Rankings by Price
Price
$1.00
$2.00
$3.00
$4.00
2
4
6
8
10
12
14
16

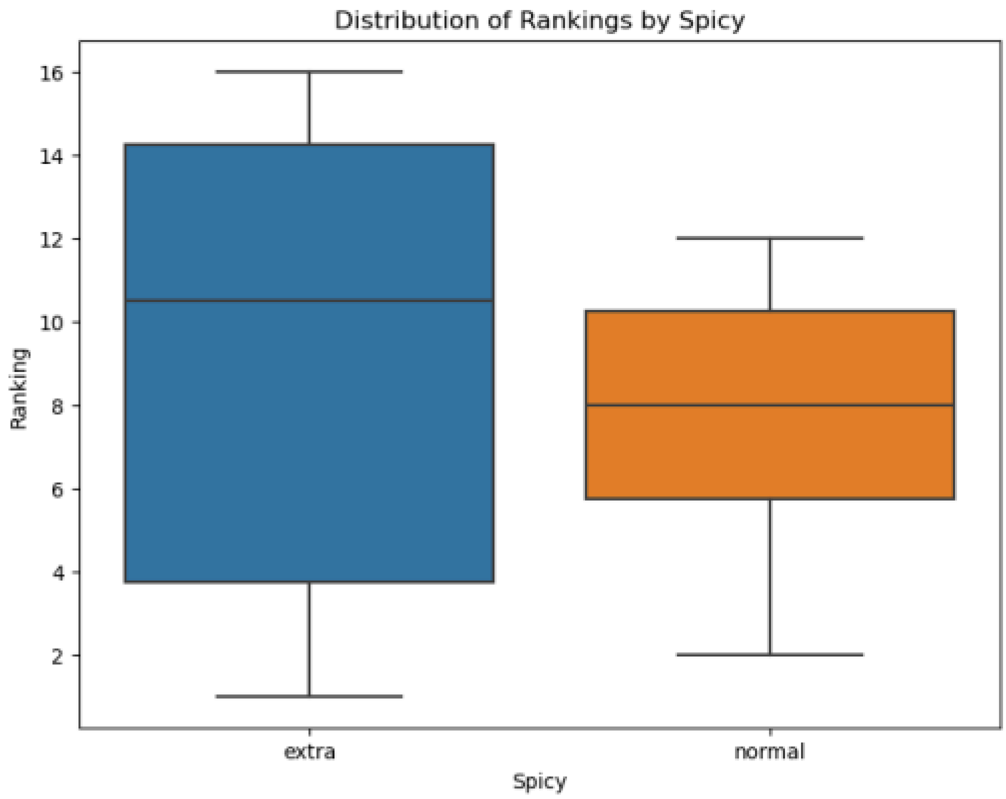

Distribution of Rankings by Weight
Ranking
Weight
100g
200g
300g
400g
16
14
12
10
8
6
4
2


Distribution of Rankings by Crust
Ranking
Crust
thick
thin
16
14
12
10
8
6
4
2


Distribution of Rankings by Cheese
Ranking
16
14
12
10
8
6
4
2
Cheddar
Mozzarella
Cheese


Distribution of Rankings by Size
Ranking
16
14
12
10
8
6
4
2
large
regular
Size





- Boxplots for each attribute, showing how rankings vary across different levels.

Interpretation:

- These plots help understand the distribution and variability of rankings, providing insights into how different levels of each attribute are perceived.

Overview of Conjoint Analysis

Meaning of Conjoint Analysis

Conjoint analysis is a statistical technique used in market research to understand how consumers value different features that make up a product or service. The method involves presenting consumers with a set of hypothetical products, each characterized by various attributes with differing levels, and asking them to choose or rank these products. By analyzing these choices or rankings, conjoint analysis helps determine the relative importance of each attribute and the specific value (part-worth utility) assigned to each level of the attribute. This enables businesses to understand the trade-offs consumers are willing to make and to design products that better meet consumer needs and preferences.

Advantages of Conjoint Analysis

1. **Understanding Consumer Preferences:** Conjoint analysis provides deep insights into consumer preferences by quantifying the value they place on different product attributes. This helps businesses understand what drives consumer choices.
2. **Informed Product Development:** By identifying the most valued attributes, businesses can design products that align closely with consumer desires, leading to higher customer satisfaction and loyalty.
3. **Pricing Strategy Optimization:** The technique allows businesses to determine the optimal price points by understanding how price changes affect consumer preferences and willingness to pay.
4. **Market Segmentation:** Conjoint analysis helps identify distinct consumer segments based on their preferences, enabling targeted marketing and personalized product offerings.
5. **Competitive Advantage:** By understanding what consumers value most, companies can differentiate their products from competitors, thereby gaining a competitive edge in the market.
6. **Resource Allocation:** It aids in prioritizing features and attributes during product development, ensuring that resources are allocated to the most impactful areas.

Real-Life Examples of Conjoint Analysis

1. **Automotive Industry:** Car manufacturers often use conjoint analysis to understand consumer preferences for various features such as engine type, fuel efficiency, safety features, and infotainment systems. This helps them design models that appeal to different market segments.
2. **Technology Products:** Tech companies use conjoint analysis to evaluate consumer preferences for attributes like screen size, battery life, camera quality, and storage capacity in smartphones or laptops. This information guides product design and marketing strategies.
3. **Healthcare Services:** Hospitals and clinics use conjoint analysis to understand patient preferences for healthcare services, including appointment scheduling, waiting times, doctor qualifications, and facility amenities. This helps improve patient satisfaction and service delivery.
4. **Retail and E-commerce:** Retailers use conjoint analysis to determine the optimal mix of product features, pricing, and promotions. For example, an online retailer might use it to understand the importance of free shipping, return policies, and delivery times to enhance customer experience and increase sales.
5. **Food and Beverage Industry:** Restaurants and food companies use conjoint analysis to gauge consumer preferences for different menu items, ingredient types, portion sizes,

and pricing. This helps in menu optimization and developing new food products that cater to customer tastes.

6. **Financial Services:** Banks and financial institutions use conjoint analysis to design and market new financial products, such as credit cards and insurance policies, by understanding which features (interest rates, rewards, fees, coverage) are most important to consumers.