VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A5- Multivariate Analysis and Business Analytics Applications

ALEENA MARY ABRAHAM

V01150203

Date of Submission: 05-07-2025

CONTENTS

INTRODUCTION

INTRODUCTION

This project applies four such techniques—Principal Component Analysis (PCA), Factor Analysis (FA), Cluster Analysis, Multidimensional Scaling (MDS), and Conjoint Analysis—to real-world marketing and consumer datasets. Each method serves a unique purpose: dimensionality reduction, segmentation, perceptual mapping, and preference modelling, respectively.

Together, these tools empower companies to make smarter decisions in market research, product development, positioning, and customer targeting.

OBJECTIVES

- To reduce and interpret high-dimensional survey data using PCA and FA.
- To segment respondents into meaningful groups using cluster analysis.
- To visualize brand perceptions and competitive positions using MDS.
- To estimate customer preferences and product trade-offs through conjoint analysis.

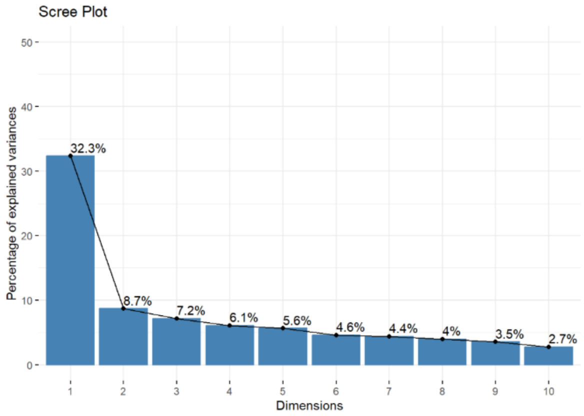
BUSINESS SIGNIFICANCE

- Improved Customer Understanding: Identifying underlying drivers of customer attitudes and behavior helps firms build better-targeted marketing strategies.
- Effective Segmentation: Cluster analysis enables the development of tailored offerings and personalized experiences for each segment.
- Smarter Positioning: MDS offers visual insights into how a brand is perceived in relation to competitors, guiding positioning decisions.
- Data-Driven Product Design: Conjoint analysis helps optimize product features, pricing, and bundles based on what customers truly value.

RESULTS AND INTERPRETATION – PART 1

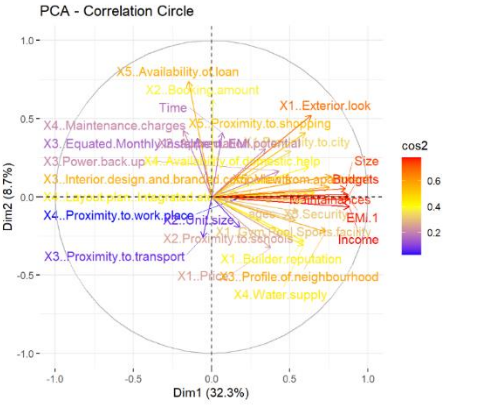
USING R

1. Scree plot



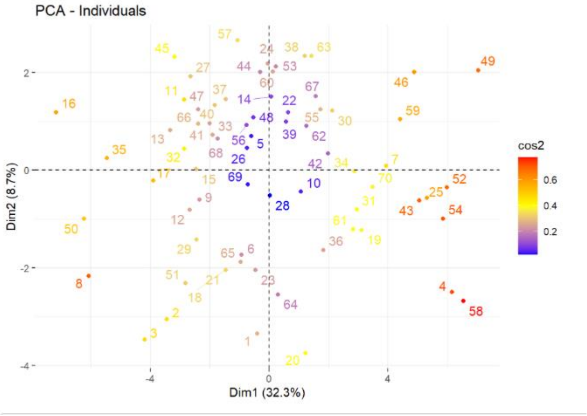
- Shows that the first principal component explains 32.3% of the variance, and the second explains 8.7%.
- Based on the "elbow" method, retaining the first few components is reasonable as the explained variance starts to level off after the third or fourth component.

2. Correlation Circle



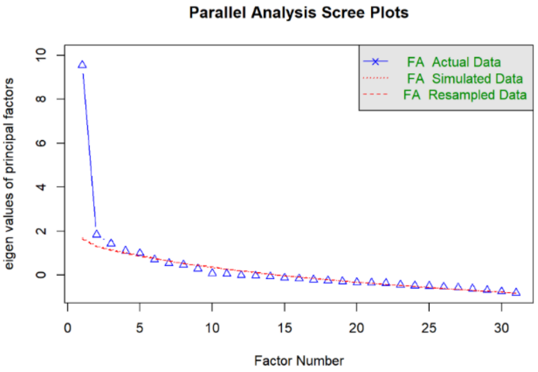
- Displays the correlation of each variable with the principal components.
- Variables closer to the circle's circumference are better represented on the factor map.
- For instance, Income, Size, Budgets, and Maintainances are strongly correlated with the first principal component.

3. PCA Biplot for Individuals



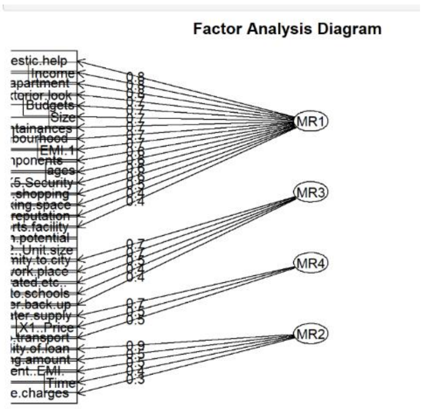
- Shows the positioning of individuals in the new component space.
- Observations close to each other are similar in terms of the variables measured.
- For example, individuals 4, 7, and 48 are similar based on their positions in the plot.

4. Parallel Analysis Scree Plot



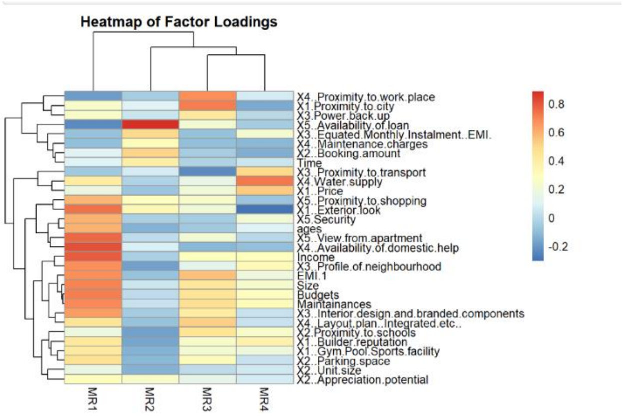
- Parallel analysis suggests: number of factors = 3 and the number of components = NA
- Suggests that three factors should be retained for FA.
- The scree plot from parallel analysis shows that the first three factors have eigenvalues significantly greater than those from random data, indicating their importance.

5. Factor Analysis



- This is a path diagram showing how 32 observed variables load onto 4 latent factors: MR1, MR2, MR3, MR4. Variables cluster logically around key decision areas: finances, amenities, aesthetics, proximity.
- Factor 1 (MR1): This factor represents people's financial capability and preference for well-designed, spacious, and secure housing with appealing views and surroundings. MR1 is the most dominant (explains 24% variance).
- Factor 2 (MR2): Reflects loan availability and repayment feasibility. Customers influenced by affordability, EMI, booking amount, and time flexibility.
- Factor 3 (MR3): Customers valuing location convenience, easy access to city, workplace, and layout integration.
- Factor 4 (MR4): This factor indicates concern for basic utilities and pricing, suggesting practical buyers focused on functionality.

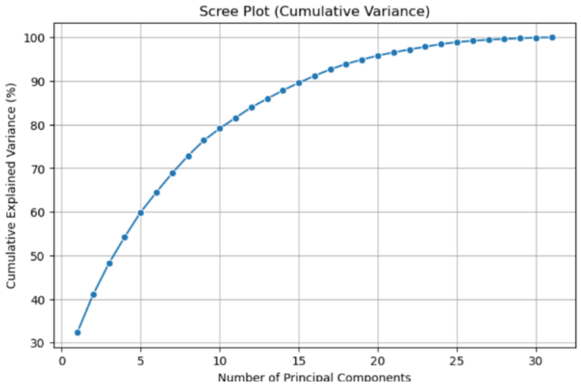
6. Heatmap



- The heatmap provides a visual representation of how variables load on different factors.
- Variables with high loadings on the same factor are grouped together, making it easier to see which variables contribute to each factor.
- For instance, Profile of neighbourhood, Water supply, and Builder reputation have high loadings on MR1, suggesting these variables are strongly associated with this factor.

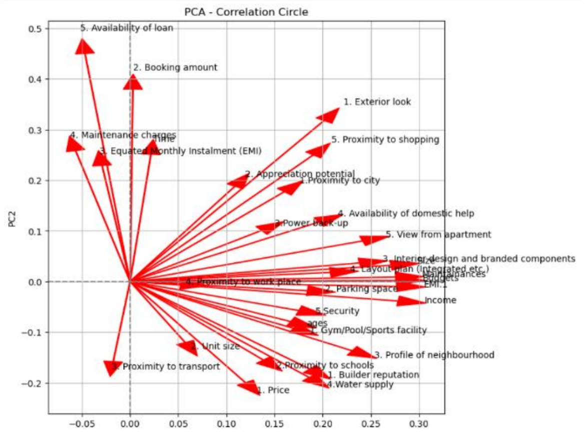
USING PYTHON

1. Scree plot



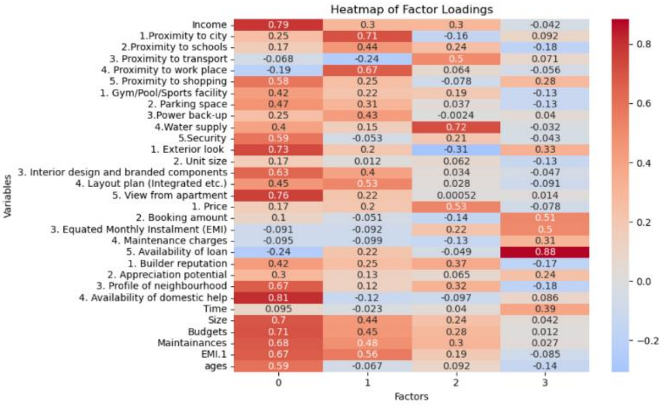
- This Scree Plot shows how much cumulative variance (%) is explained by the Principal Components (PCs) in a PCA (Principal Component Analysis).
- The first few components capture a large portion of variance.
- The elbow or bend in the curve indicates the point beyond which additional components contribute diminishing returns in terms of explained variance.
- The curve flattens as we approach 100% cumulative variance.

2. PCA – Correlation Circle



- The correlation circle (or PCA biplot) helps us understand how the original variables contribute to the first two principal components.
- Longer arrows indicate stronger contributions of the variable to PC1 and PC2
- Arrows pointing in the same direction are positively correlated. For example, Exterior look, Proximity to shopping, and Income likely move together.
- Arrows in opposite directions indicate negative correlation. For example, Proximity to transport vs Availability of loan may have a negative correlation.

3. Heatmap of Factor loadings

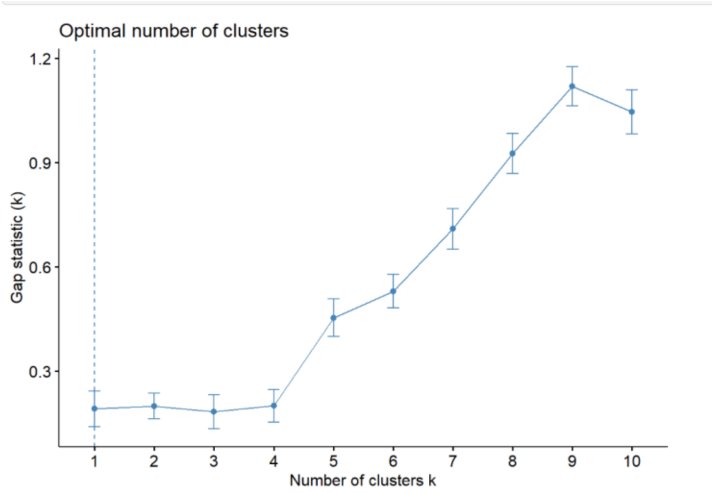


- This heatmap visualizes how each variable in the dataset loads onto each extracted factor (likely MR1 to MR4), from the factor analysis
- It helps in segmenting buyers, simplifying variables for modelling, and interpreting latent consumer decision drivers in real estate.

RESULTS AND INTERPRETATION – PART 2

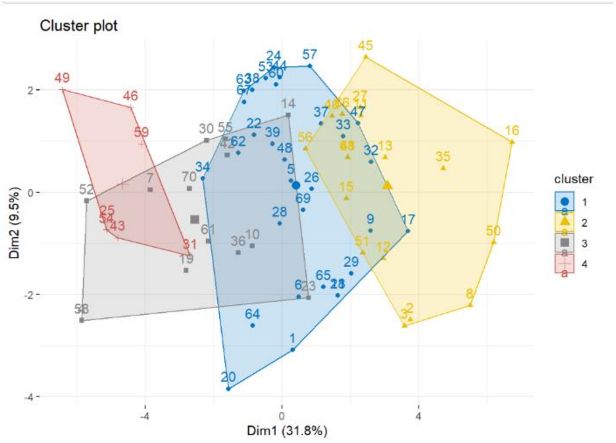
USING R

1. Optimal Number of Clusters – GAP Statistic



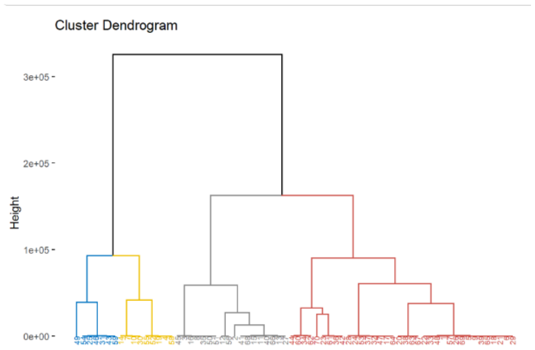
- The gap statistic is lowest at k = 1, meaning no clustering is better than random here.
- The gap statistic increases with more clusters and peaks around k = 9.
- The maximum gap is at k = 9, suggesting that 9 is the optimal number of clusters.

2. K-means Clustering



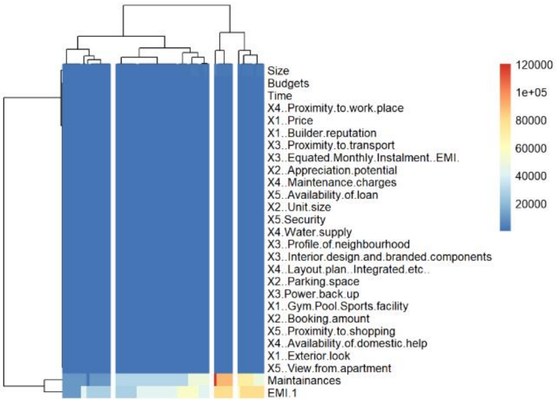
- 4-cluster solution shows clear structure, meaning k = 4 was a reasonable choice for exploratory analysis.
- Clusters can now be profiled further using original variables (e.g., income, proximity, features) to understand:
  - What defines each segment (e.g., budget-conscious, location-prioritizing, luxury-oriented).
  - How to target or recommend housing options to each cluster.

3. Hierarchical Clustering – Dendrogram



- 4-cluster hierarchical model is valid and consistent with the earlier K-Means cluster analysis.
- Each branch represents unique groupings based on housing preferences or related features.
- Dendrograms give a visual validation of the distance between clusters, making them useful for deciding the number of clusters when k is unknown.

4. Heatmap

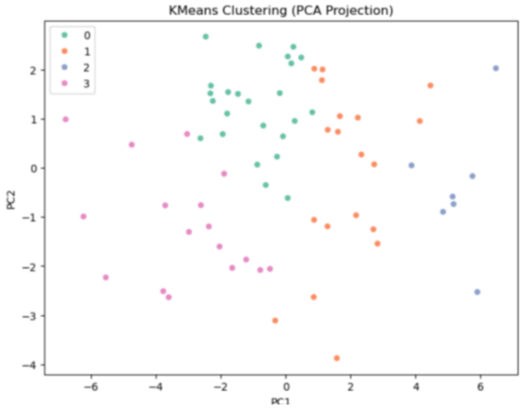


This heatmap complements the PCA and clustering results by:

- Highlighting which features drive cluster separation.
- Showing clear distinctions in preferences/cost sensitivity.
- Helping the segment respondents (e.g., affordability-focused, aesthetics-driven, location-prioritizes).

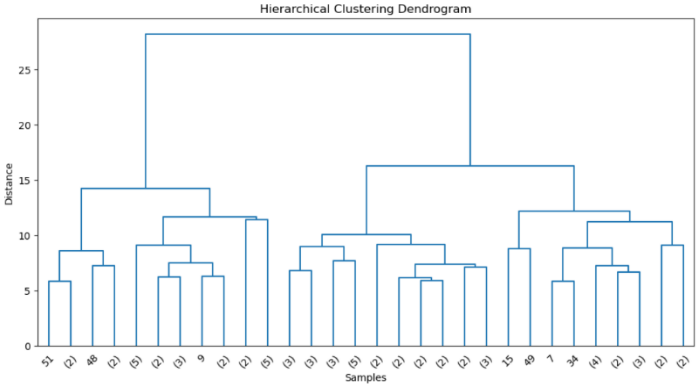
USING PYTHON

1) K-means Clustering



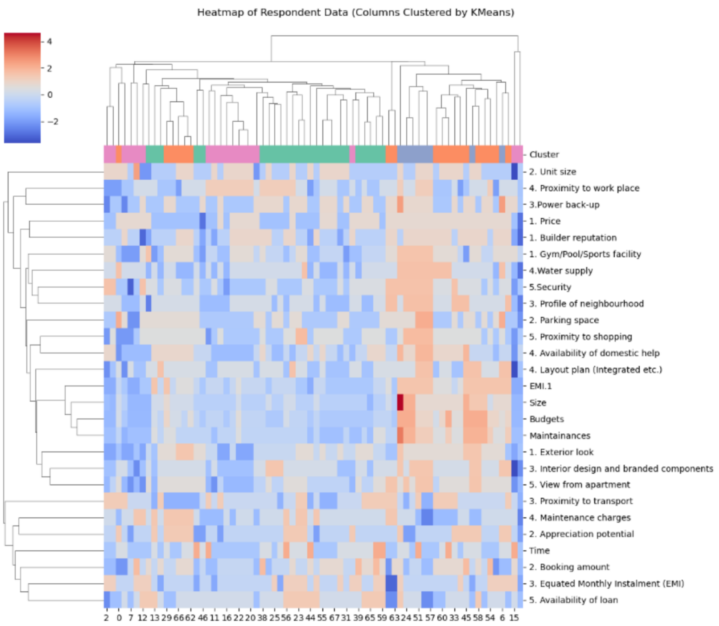
- This PCA + K-means clustering result visualizes market segments effectively.
- The spread of Cluster 1 and Cluster 2 suggests those segments have more within-group variation.
- Cluster 3 is compact, which may indicate a very specific preference profile (e.g., high-end lifestyle seekers).

2) Hierarchical Clustering – Dendrogram



- 4-cluster hierarchical model is valid and consistent with the earlier K-Means cluster analysis.
- Dendrograms give a visual validation of the distance between clusters, making them useful for deciding the number of clusters when k is unknown.

3) Heatmap

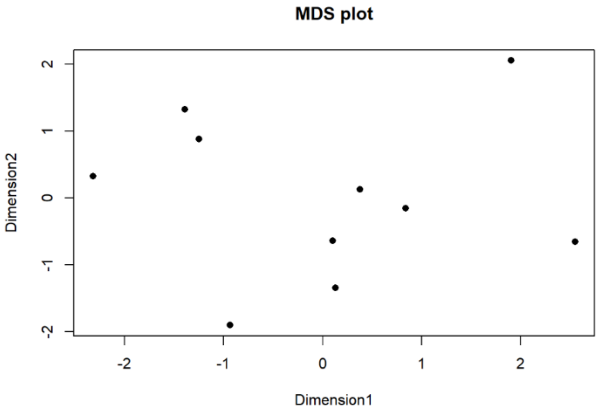


This heatmap complements the PCA and clustering results by:

- Highlighting which features drive cluster separation.
- Showing clear distinctions in preferences/cost sensitivity.

RESULTS AND INTERPRETATION – PART 3

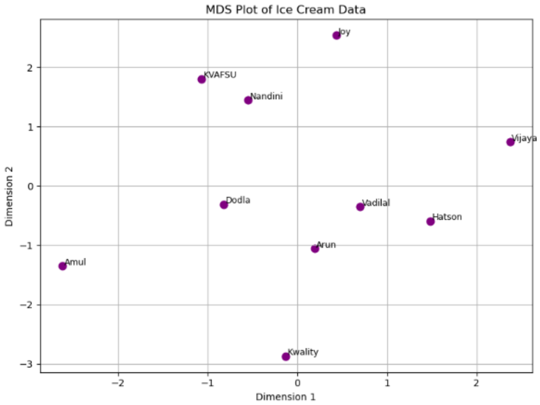
USING R



The MDS plot helps visualize how different ice cream brands compare based on their attributes.

- Brands that are closer together in the plot are more similar based on the 6 attributes (Price, Availability, etc.).
- Axes Interpretation
  - Dimension1 might correspond to a dominant factor (e.g., Price vs. Quality).
  - Dimension2 could represent another key factor (e.g., Taste vs. Shelflife).

USING PYTHON



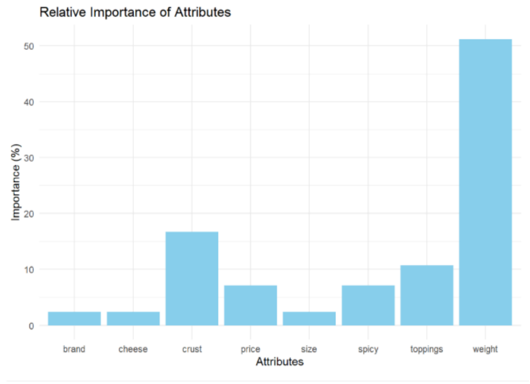
This MDS plot visualizes the ice cream data, revealing:

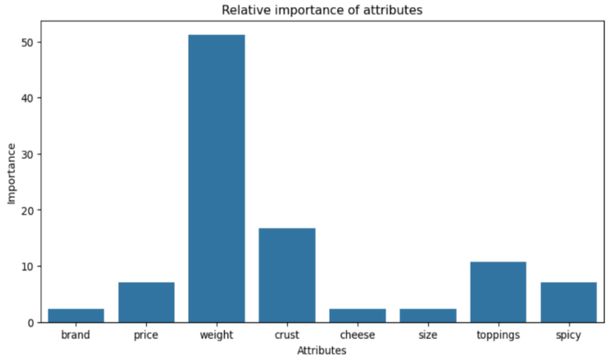
- Clusters of similar brands.
- Outliers with unique attributes.
- Potential market segments (budget vs. premium).

RESULTS AND INTERPRETATION – PART 4

USING R & PYTHON

1. Attribute importance





The bar chart visualizes the relative importance (%) of different in influencing a target variable—likely consumer preferences, sales, or product performance in the context of pizza or food products.

2. Conjoint Analysis

- Price sensitivity is high: $1.00 is most preferred, showing cost plays a large role in decision-making.
- Flavor and texture preferences lean toward thick crust, extra spicy, and mozzarella cheese, reflecting a desire for richer, more intense eating experiences.
- Portion control: Lighter weight (100g) but regular size suggests people may prefer the perception of a "normal" portion but not an overloaded one.
- Brand trust: Pizza Hut may benefit from strong brand recognition or past positive experiences.

RECOMMENDATIONS

This project applies four such techniques—Principal Component Analysis (PCA), Factor Analysis (FA), Cluster Analysis, Multidimensional Scaling (MDS), and Conjoint Analysis—to real-world marketing and consumer datasets.

Summary of Key Findings:

1. Using PCA, factor analysis, and clustering, customers were grouped into four key segments based on preferences for financial options, location, aesthetics, and practical utilities.
2. Price ($1.00), EMI, and budget features ranked highest across analyses, showing that financial factors are dominant decision drivers.
3. Consumers showed consistent preference for Pizza Hut, mozzarella cheese, thick crust, and extra spicy flavours, indicating a tilt toward bold, rich, and trusted choices.
4. Conjoint analysis revealed that crust type, cheese, spiciness, and price are the most influential factors shaping consumer preferences

Recommendations:

1. Tailor marketing messages and offerings to the 4 identified segments—for example, finance-driven buyers vs. lifestyle-focused buyers.
2. Maintain a strong low-price offering (like $1.00) and flexible payment options to appeal to budget-conscious customers.
3. Highlight attributes like Pizza Hut branding, mozzarella, and spicy/thick crust options in advertising and product packaging to resonate with consumer preferences.
4. Use insights from attribute importance to guide product development, pricing models, and promotional campaigns around top-rated features.

CODES

R Codes:

Part 1:

```
# Function to auto-install and load packages
install_and_load <- function(packages) {
  for (package in packages) {
    if (!require(package, character.only = TRUE)) {
      install.packages(package, dependencies = TRUE)
      library(package, character.only = TRUE)
    }
  }
}

# List of packages to install and load
packages <- c("dplyr", "psych", "tidyr", "GPArotation", "FactoMineR", "factoextra",
"pheatmap")

# Install and load necessary packages
install_and_load(packages)

# Load the necessary libraries
library(dplyr)
library(psych)
library(tidyr)
library(GPArotation)
library(FactoMineR)
library(factoextra)
library(pheatmap)

# Load the dataset
```

```
dataset_path <- "C:/Users/Aleena Mary Abraham/OneDrive/Desktop/SCMA632_2025/DATA/Survey.csv"

survey_data <- read.csv(dataset_path)


# Inspect the dataset

str(survey_data)

summary(survey_data)


# Dataset contains both categorical and numerical variables, we will focus on the numerical variables for PCA and FA

# Select only the numerical variables for PCA and FA

# Assuming numerical variables are those that are integers or numeric

numerical_data <- survey_data %>% select(where(is.numeric))


# Standardize the data

survey_data_scaled <- scale(numerical_data)


# Perform PCA using FactoMineR

pca_result <- FactoMineR::PCA(survey_data_scaled, graph = FALSE)


# Summary of PCA results

print(summary(pca_result))


# Visualize the scree plot

factoextra::fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 50), main = "Scree Plot")


# Visualize the variables on the principal component map (Correlation Circle)

factoextra::fviz_pca_var(pca_result, col.var = "cos2",

                    gradient.cols = c("blue", "yellow", "red"),

                    repel = TRUE, title = "PCA - Correlation Circle")
```

```
# Visualize individuals on the principal component map
factoextra::fviz_pca_ind(pca_result, col.ind = "cos2",
                         gradient.cols = c("blue", "yellow", "red"),
                         repel = TRUE, title = "PCA - Individuals")

# Determine the number of factors for FA using parallel analysis
fa_parallel <- psych::fa.parallel(survey_data_scaled, fa = "fa")
print(fa_parallel)

# Perform Factor Analysis with the chosen number of factors
fa_result <- psych::fa(survey_data_scaled, nfactors = 4, rotate = "varimax")

# Print FA results
print(fa_result)

# Factor Loadings
fa_loadings <- fa_result$loadings
print(fa_loadings)

# Plot Factor Analysis results
psych::fa.diagram(fa_result, main = "Factor Analysis Diagram")

# Heatmap of Factor Loadings using pheatmap
loadings_matrix <- as.matrix(fa_loadings)
pheatmap::pheatmap(loadings_matrix, cluster_rows = TRUE, cluster_cols = TRUE, main =
"Heatmap of Factor Loadings")
```

Part 2:

```
# Function to auto-install and load packages
install_and_load <- function(packages) {
  for (package in packages) {
    if (!require(package, character.only = TRUE)) {
      install.packages(package, dependencies = TRUE)
    }
    library(package, character.only = TRUE)
  }
}

# List of packages to install and load
packages <- c("cluster", "FactoMineR", "factoextra", "pheatmap")
install_and_load(packages)

#C) Do multidimensional scaling and interpret the results.

icecream_df<-read.csv('C:/Users/Aleena Mary Abraham/OneDrive/Desktop/SCMA632_2025/DATA/icecream.csv',header=TRUE)
dim(icecream_df)

names(icecream_df)

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")
```

Part 3:

```
# Function to auto-install and load packages
install_and_load <- function(packages) {
  for (package in packages) {
    if (!require(package, character.only = TRUE)) {
      install.packages(package, dependencies = TRUE)
    }
    library(package, character.only = TRUE)
  }
}

# List of packages to install and load
packages <- c("cluster", "FactoMineR", "factoextra", "pheatmap")
install_and_load(packages)

#C) Do multidimensional scaling and interpret the results.

icecream_df<-read.csv('C:/Users/Aleena Mary Abraham/OneDrive/Desktop/SCMA632_2025/DATA/icecream.csv',header=TRUE)
dim(icecream_df)

names(icecream_df)

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")
```

Part 4:

```
# Load necessary packages
library(readr)
library(dplyr)
library(ggplot2)
library(tidyr)
```

```
# Read CSV file
df <- read_csv("C:/Users/Aleena Mary Abraham/OneDrive/Desktop/SCMA632_2025/DATA/pizza_data.csv")


# Convert relevant variables to factors
conjoint_attributes <- c("brand","price","weight","crust","cheese","size","toppings","spicy")
df[conjoint_attributes] <- lapply(df[conjoint_attributes], as.factor)


# Fit linear model with effects coding (sum contrasts)
options(contrasts = c("contr.sum", "contr.poly"))


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


part_worth_list <- list()
level_names <- list()
part_worth_ranges <- c()
important_levels <- list()


start_index <- 2  # Skip intercept


for (attr in conjoint_attributes) {
  levels_attr <- levels(df[[attr]])
  k <- length(levels_attr)


  # Extract effect-coded coefficients
  coefs <- coef(model)[start_index:(start_index + k - 2)]
  last_coef <- -sum(coefs)
  part_worths <- c(coefs, last_coef)
  part_worth_list[[attr]] <- part_worths
  level_names[[attr]] <- levels_attr

  # Identify most preferred level
  important_levels[[attr]] <- which.max(part_worths)
  part_worth_ranges <- c(part_worth_ranges, max(part_worths) - min(part_worths))

  start_index <- start_index + k - 1
}

attribute_importance <- round(100 * part_worth_ranges / sum(part_worth_ranges), 2)

# Create importance data frame
importance_df <- data.frame(
  Attribute = conjoint_attributes,
  Importance = attribute_importance
)

# Bar plot of attribute importance
ggplot(importance_df, aes(x = Attribute, y = Importance)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  ggtitle("Relative Importance of Attributes") +
  xlab("Attributes") + ylab("Importance (%)") +
  theme_minimal()

# Create lookup table
part_worth_dict <- unlist(part_worth_list)
names(part_worth_dict) <- unlist(lapply(names(part_worth_list), function(attr) {
  paste(attr, level_names[[attr]], sep = ":")
}))
```

```
# Compute utility for each row
df$utility <- apply(df, 1, function(row) {
  sum(sapply(conjoint_attributes, function(attr) {
    part_worth_dict[[paste(attr, row[[attr]], sep = ":")]]
  }))
})

# Show profile with highest utility
df[which.max(df$utility), ]

for (attr in conjoint_attributes) {
  cat("Preferred level in", attr, "is:", level_names[[attr]][important_levels[[attr]]], "\n")
}
```