

# VIRGINIA COMMONWEALTH UNIVERSITY

A4- MULTIVARIATE ANALYSIS AND BUSINESS ANALYTICS APPLICATIONS

### Analysis of data- Descriptive statistics

**VARUN YOGANANDA**

### V01107265

**Date of Submission: 08-07-2024**

**CONTENTS**

|  |  |
| --- | --- |
| **Sl. No.** | **Title** |
| **1.** | Introduction |
| **2.** | Objectives |
| **3.** | Business Significance |
| **4.** | Codes, Results And Interpretation |

# INTRODUCTION

In market research, various statistical techniques are employed to extract meaningful insights from data, enhancing decision-making processes. Principal Component Analysis (PCA) and Factor Analysis (FA) are used to uncover latent variables and reduce the dimensionality of complex datasets, revealing underlying patterns in respondent data. Cluster Analysis segments respondents into homogeneous groups based on similarities in background variables, facilitating targeted marketing strategies and customer segmentation. Multidimensional Scaling (MDS) is utilized to visualize and interpret similarities or dissimilarities among items or concepts, such as consumer preferences for different ice cream flavours in this context. Lastly, Conjoint Analysis is pivotal in understanding consumer preferences by analysing trade- offs among various product attributes, such as crust type, toppings, and pricing in the pizza market. These methods collectively provide a comprehensive framework for dissecting consumer behaviour, optimizing product offerings, and gaining competitive advantage in dynamic markets.

# OBJECTIVES:

• Dimension Reduction and Pattern Recognition: To simplify data and find underlying patterns, apply PCA and FA.   
  
• Respondent Segmentation: Based on background characteristics, divide respondents into discrete groups using Cluster Analysis.  
  
  
• Visualisation of Similarities: To visually understand the similarities between ice cream flavours, use MDS.   
  
• Preference Analysis and Optimisation: To evaluate customer preferences and improve product features, use conjoint analysis.   
  
• Insight Creation and Decision Support: Produce practical insights to aid in the formulation of strategic decisions pertaining to marketing and product development.

# BUSINESS SIGNIFICANCE

These analytical methods are important for business because they may convert unprocessed data into useful insights that improve competitiveness and inform strategic choices. Businesses can uncover important elements impacting customer behaviour and preferences and streamline operations and marketing activities by utilising Principal Component Analysis (PCA) and Factor Analysis (FA). By dividing consumers into relevant groups based on similar traits, cluster analysis supports targeted marketing strategies and enables customised product offerings and ad campaigns. Businesses can better comprehend market positioning and product differentiation by utilising Multidimensional Scaling (MDS), a visual mapping of consumer perceptions. Conjoint analysis gives firms extensive insights into consumer preferences, allowing them to optimise pricing and feature combinations for their products. When used in tandem, these strategies enable companies to better satisfy customers, match offers to expectations, and capitalize on market opportunities, ultimately driving growth and profitability in competitive markets.

# CODES, RESULTS AND INTERPRETATION:

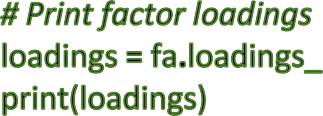
**Part A** - Perform Principal Component Analysis and Factor Analysis to identify data dimensions

**Python Code(Factor Analysis):**



















**R Code(Factor Analysis):**



survey\_df<-read.csv('/Users/varuny/Downloads/A4

DATA/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)



print(factor\_analysis$communality)



print(factor\_analysis$scores)

**Python Code(Component Analysis):**



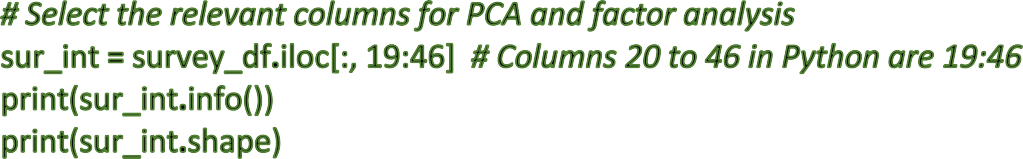


survey\_df <- read.csv('/Users/varuny/Downloads/A4 DATA/Survey.csv', header

= TRUE)

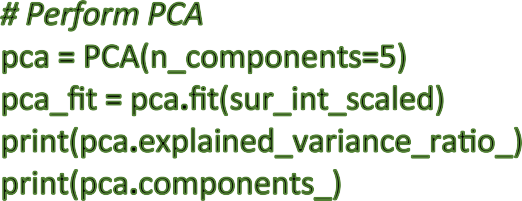
dim(survey\_df)

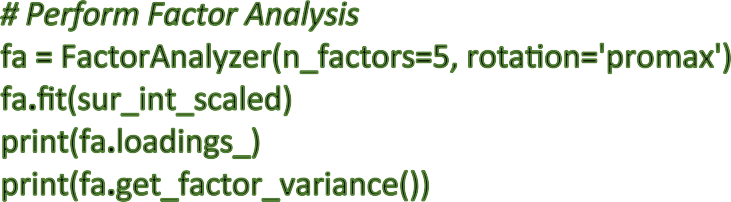




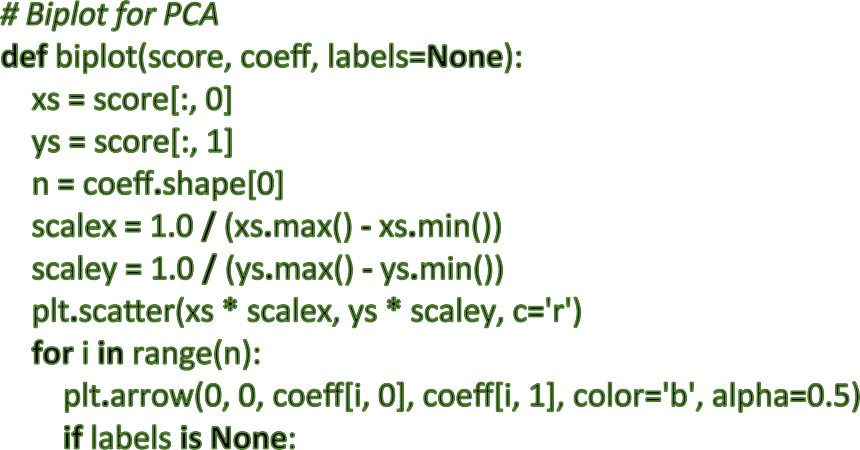






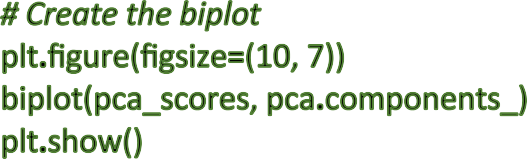












**R code(Component Analysis):**



*# Load the data*

survey\_df <- read.csv('/Users/varuny/Downloads/A4 DATA/Survey.csv', header

= TRUE)

dim(survey\_df) names(survey\_df) head(survey\_df) str(survey\_df)

*# A) Do principal component analysis and factor analysis and identify the dimensions in the data.*



*# Check for NA values*

sum(is.na(survey\_df))

*# Select the relevant columns for PCA and factor analysis*

sur\_int <- survey\_df[, 20:46] str(sur\_int)

*# Perform PCA using psych package*

pca\_psych <- principal(sur\_int, 5, n.obs = 162, rotate = "promax") pca\_psych

*# Perform Omega hierarchical analysis*

om.h <- omega(sur\_int, n.obs = 162, sl = FALSE)

op <- par(mfrow = c(1, 1))

om <- omega(sur\_int, n.obs = 162)

*# Perform PCA using FactoMineR package* pca\_fm <- PCA(sur\_int, scale.unit = TRUE) summary(pca\_fm)

*# Biplot using factoextra*

fviz\_pca\_biplot(pca\_fm, repel = TRUE)

*# Show the structure and dimensions of the selected columns*

str(sur\_int)

**Result:**

[[-0.15781009 0.42809134 0.05879243 -0.13526352]

[ 0.07288457 0.07797564 -0.09376015 0.72270594]

[ 0.67791474 -0.10906029 0.25599157 0.12315536]

[ 0.45294307 0.19306866 -0.14202126 0.09102672]

[ 0.56145098 0.05977592 -0.15846106 0.10167805]

[ 0.44508348 -0.0324726 -0.02008422 0.46566121]

[ 0.40127304 0.65592565 -0.04681206 0.06621177]

[ 0.54536401 0.17133537 -0.06944276 -0.1745679 ]

[ 0.77346812 -0.23204538 0.31139215 -0.08601212]

[ 0.13600482 0.10476011 -0.07605281 -0.1851078 ]

[ 0.74442306 0.0982716 -0.06279514 0.13271725]

[ 0.61456757 0.12385518 -0.10379084 0.25558762]

[ 0.831264 0.04360933 -0.01602555 -0.06757234]

[ 0.18089295 0.5781131 -0.06496421 0.09677394]

[ 0.08522971 -0.12722077 0.5149248 -0.0147638 ]

[-0.12867754 0.18571432 0.51035205 -0.01479766]

[-0.09884299 -0.09829734 0.31043878 -0.06327067]

[-0.12075303 -0.08561139 0.88293084 0.28205672]

[ 0.41597756 0.48625238 -0.13041834 -0.03370875]

[ 0.3030142 0.14591216 0.26405999 -0.08696872]

[ 0.64708197 0.39365565 -0.18483817 -0.14750424]

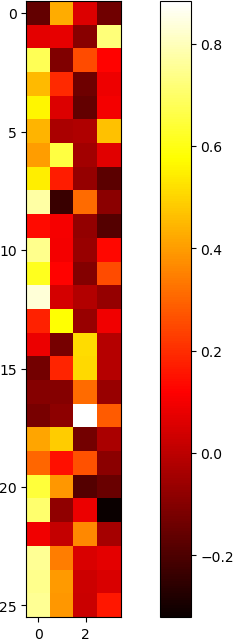
[ 0.70663536 -0.07753737 0.08586913 -0.32104338]

[ 0.09492154 0.01416831 0.36329693 -0.04207299]

[ 0.75374487 0.34338309 0.05243772 0.07115759]

[ 0.74497866 0.39843936 0.02755836 0.05399374]

[ 0.75557128 0.39188083 0.02353685 0.15929651]]



Communalities:

[0.22991899 0.54248721 0.55216147 0.27088883 0.35424869 0.41639751

0.59783387 0.36207396 0.75646115 0.06952092 0.5853801 0.46913098

0.69772445 0.38052256 0.28881475 0.3117259 0.11980772 0.88103346

0.42762393 0.1903992 0.62960249 0.61578794 0.14296564 0.6938564

0.71742191 0.75038791]

Factor Scores:

[[-6.28619743e-01 1.85657494e+00 -5.47519178e-01 -8.81439669e-01] [-1.69513914e+00 1.12800561e+00 -5.80307050e-01 -9.38083207e-01] [-1.20280931e+00 5.89699103e-01 -1.39117035e+00 -2.86780837e+00] [ 1.99352990e+00 4.88864495e-02 -2.07936514e+00 2.18544871e-01] [ 4.11314878e-01 -1.26821776e+00 -7.80870767e-01 6.66228671e-02] [-2.11478560e-01 2.03794816e-01 -6.50677463e-01 2.05174946e-01]

[ 7.43577413e-01 6.43368392e-01 1.00212144e-01 1.03878997e+00] [-1.91404810e+00 3.03937289e-01 -1.32917310e+00 -2.15077532e-01] [-8.56474198e-01 -2.96858773e-03 2.93861146e-01 9.20159872e-02] [ 2.36520458e-01 5.51241056e-02 3.48090670e-02 -6.03796924e-01]

[-4.68064081e-01 -8.66072292e-01 -7.46975265e-02 -6.21582489e-01]

[-8.75004407e-01 -1.08807300e+00 -1.13530109e+00 6.56135510e-01] [-1.19487644e+00 8.94305070e-02 4.89637706e-01 9.91715591e-01] [-3.01457336e-01 1.00436570e-01 1.20733916e+00 1.97792324e-01] [-1.00291293e+00 4.94104285e-01 -1.24637702e-01 -2.07152409e-01] [-2.44664326e+00 -1.14035022e-01 1.61004829e-01 5.75188733e-01] [-1.25577846e+00 -3.56172547e-01 -7.15364485e-01 1.04921625e+00] [-1.17254635e+00 1.21161268e+00 -1.83678942e-01 8.19594214e-01] [ 4.04252036e-01 1.29219401e+00 -1.50176103e+00 1.71572855e-01] [ 4.91676501e-01 1.92667044e-01 -2.44581210e+00 -4.37601486e-01] [-1.17254635e+00 1.21161268e+00 -1.83678942e-01 8.19594214e-01] [-1.18430113e-01 6.13931005e-01 1.08967528e+00 1.32810436e+00] [-9.89414929e-01 1.50327651e+00 -2.60459187e-01 9.36551773e-01] [ 5.45913862e-01 -6.12972222e-01 1.19522542e+00 4.73974759e-01] [ 1.60656285e+00 8.48989955e-01 2.09647043e-01 3.46249330e-01] [-4.01168383e-01 -2.52486384e-01 -1.57889392e-01 1.35125021e+00] [-5.70149338e-01 -1.01616206e+00 1.02586710e+00 8.78902429e-01] [ 5.57012596e-02 -5.78556540e-02 5.09137586e-03 6.76539136e-01]

[-1.07970184e+00 5.37184202e-01 -4.86901676e-01 1.41408056e+00] [ 8.96814338e-01 -6.47961098e-01 -7.65034396e-01 -7.04074626e-01] [ 8.69607701e-01 5.95486198e-01 -6.38034715e-01 3.78813780e-01] [-6.37345825e-01 -1.10103294e+00 1.22809820e-01 -4.53821368e-02] [-5.84225979e-01 4.19191582e-02 1.27085126e+00 -1.66638848e-01] [ 8.44456916e-01 4.74671338e-01 6.06587744e-01 -4.49510340e-01] [-1.53043482e+00 3.06961797e-01 5.04757719e-01 -2.77263230e-01] [ 2.52217422e-01 8.37519715e-01 -6.55798084e-01 -1.08047700e+00] [-2.65256836e-01 -3.93585990e-01 1.00265475e+00 6.73951617e-01] [ 8.75266361e-01 -1.57998563e+00 1.05827836e-01 -5.01791914e-01] [-5.83336049e-02 -1.24268812e-01 3.12217836e-01 6.25041344e-01] [-1.50415166e-01 -5.71515007e-01 4.42151168e-01 -1.13739766e+00] [-4.15457002e-01 -1.06805866e+00 -3.79121037e-01 4.16298983e-01] [ 5.67742135e-01 8.66194617e-01 1.18757213e+00 -1.73348437e+00] [ 1.48058278e+00 5.83409123e-02 -1.51677332e+00 6.14654585e-01] [ 8.84020437e-04 4.06199309e-01 1.29140547e+00 -5.16368191e-01]

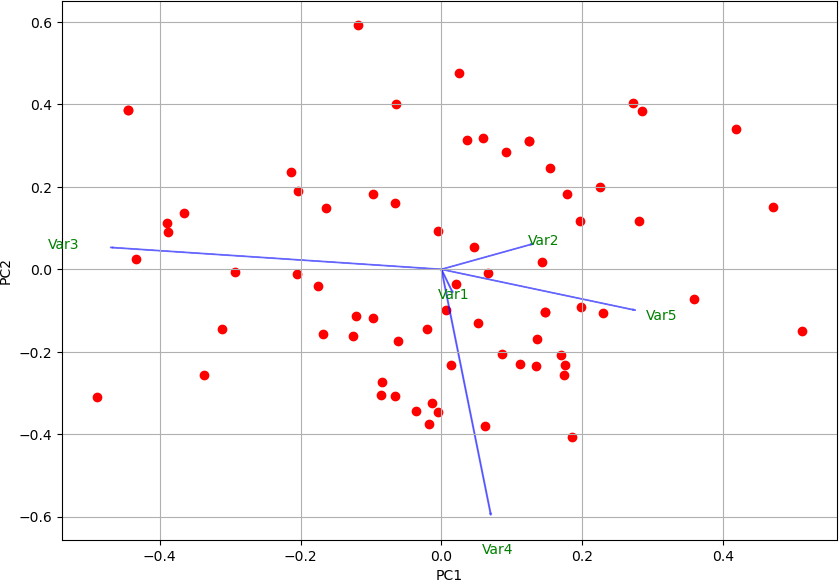
[-2.56762812e-01 -1.15793950e+00 1.58791083e+00 -1.52548112e+00] [ 1.44816075e+00 4.44470328e-01 7.83829956e-01 -6.65724886e-01]

[ 8.48846223e-02 -2.16546175e+00 -1.80248526e-01 4.93744250e-01] [ 5.33469552e-01 -1.26207193e+00 7.49243048e-02 -7.46248246e-01] [ 2.00085397e+00 1.00510014e+00 1.59107977e+00 1.15336865e+00]

[-1.42738759e+00 -2.49549276e+00 -1.05660563e+00 -4.33123005e-01] [-8.65061774e-01 1.96153108e-01 -7.08579570e-01 -1.83360924e+00] [ 1.70501397e+00 4.92152231e-01 -3.76624052e-01 4.57062101e-01]

[ 3.96019577e-01 1.81555993e-01 1.77931133e+00 2.73626585e-01] [ 1.46786634e+00 1.05831828e+00 -2.58997433e-01 4.92588720e-01] [ 6.65334799e-01 2.27176811e-01 6.06908953e-01 8.51815039e-01] [ 2.74710656e-01 -1.41357731e+00 6.77218647e-02 -7.67941001e-02] [-1.97347958e-01 1.12002694e-01 1.05482646e+00 2.70223607e-01] [ 1.99352990e+00 4.88864495e-02 -2.07936514e+00 2.18544871e-01] [ 1.12932110e+00 9.58404407e-01 1.68152565e+00 5.00130434e-01] [ 1.93912354e-01 -5.05362246e-03 1.28422043e+00 -5.57090529e-02]

[ 4.81064721e-01 1.00562158e+00 -5.75465382e-01 -1.29708769e+00] [ 8.71967700e-01 -8.31499518e-01 -3.84023259e-01 -4.96746391e-01] [ 8.93449712e-01 -1.27081949e+00 3.39359988e-01 -6.69371782e-01] [ 2.08727188e-01 3.38523707e-01 -2.09368060e+00 -3.60126571e-01] [-7.24940260e-01 6.18001945e-01 1.56431531e-01 7.05743950e-01] [-4.73622370e-01 -3.85109317e-01 2.20529709e+00 -4.48477907e-01] [ 6.28468721e-01 -1.07895387e+00 7.57875216e-01 -1.42062850e+00] [-4.15457002e-01 -1.06805866e+00 -3.79121037e-01 4.16298983e-01] [-1.69905899e-01 -4.61872565e-01 -4.66795240e-01 1.50323517e+00] [ 4.75841673e-01 1.51884310e+00 5.13103154e-01 -9.38694670e-01]]



<class 'pandas.core.frame.DataFrame'> RangeIndex: 70 entries, 0 to 69

Data columns (total 27 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

1. 3. Proximity to transport 70 non-null int64
2. 4. Proximity to work place 70 non-null int64
3. 5. Proximity to shopping 70 non-null int64
4. 1. Gym/Pool/Sports facility 70 non-null int64
5. 2. Parking space 70 non-null int64
6. 3.Power back-up 70 non-null int64
7. 4.Water supply 70 non-null int64
8. 5.Security 70 non-null int64
9. 1. Exterior look 70 non-null int64
10. 2. Unit size 70 non-null int64
11. 3. Interior design and branded components 70 non-null int64 11 4. Layout plan (Integrated etc.) 70 non-null int64
12. 5. View from apartment 70 non-null int64
13. 1. Price 70 non-null int64
14. 2. Booking amount 70 non-null int64
15. 3. Equated Monthly Instalment (EMI) 70 non-null int64
16. 4. Maintenance charges 70 non-null int64
17. 5. Availability of loan 70 non-null int64
18. 1. Builder reputation 70 non-null int64
19. 2. Appreciation potential 70 non-null int64
20. 3. Profile of neighbourhood 70 non-null int64
21. 4. Availability of domestic help 70 non-null int64
22. Time 70 non-null int64
23. Size 70 non-null int64
24. Budgets 70 non-null float64
25. Maintainances 70 non-null int64
26. EMI.1 70 non-null int64 dtypes: float64(1), int64(26)

memory usage: 14.9 KB None

(70, 27)

[0.31822766 0.09526344 0.06889323 0.06341429 0.06111318]

[[ 1.49276587e-02 -5.26532839e-02 -2.10211853e-01 -1.80419220e-01

-2.03748290e-01 -1.55293890e-01 -2.03859257e-01 -1.96651866e-01

-2.19629828e-01 -5.43506790e-02 -2.59514035e-01 -2.29951976e-01

-2.71175386e-01 -1.30693678e-01 -4.70217911e-03 2.95043360e-02 5.15162914e-02 5.07457061e-02 -1.98477129e-01 -1.18254327e-01

-2.53554825e-01 -2.13142745e-01 -2.69185464e-02 -2.88180575e-01

-2.92769508e-01 -2.94955205e-01 -2.94392460e-01]

[ 1.26411257e-01 6.01527039e-02 -2.70908994e-01 8.99167940e-02 3.70040684e-02 -7.65588314e-02 1.63702992e-01 2.24748440e-02

-3.16340689e-01 8.85312101e-02 -2.31063938e-02 2.27304792e-02

-7.94293310e-02 2.05164355e-01 -4.03506723e-01 -2.66246056e-01

-2.61889932e-01 -4.65933650e-01 1.82090697e-01 -1.90878205e-01 1.32598358e-01 -1.56897193e-01 -2.70334357e-01 -1.65118969e-02

1.67339322e-02 9.79421556e-03 4.60934585e-02]

[-4.65915170e-01 5.26993480e-02 1.29611722e-01 1.54022592e-02 1.46866978e-01 1.56560759e-01 -3.39979975e-01 -3.21752539e-03

1.77654310e-01 -2.36570397e-02 1.10407342e-01 1.04017487e-01

1.13519143e-01 -3.65620350e-01 -1.11072526e-01 -4.10786573e-01

-1.10275471e-01 -2.16143043e-01 -2.17036665e-01 -1.99931055e-01

-8.01910039e-02 1.43302818e-01 -1.70545023e-01 -6.03147080e-02

-8.67500868e-02 -7.40437311e-02 3.98915191e-04]

[ 6.97233019e-02 -5.92034448e-01 -5.31580453e-02 -8.11244151e-02

-5.69736560e-02 -4.26885850e-01 -7.57727163e-02 1.91186255e-01 1.38517019e-01 3.30608300e-01 -2.50607851e-02 -1.44090971e-01

1.25857984e-01 -1.17723023e-01 1.92757321e-02 -6.90324059e-02

4.07691247e-02 -2.21260525e-01 7.70408873e-02 1.23134684e-01

1.55028056e-01 3.25949516e-01 3.24923828e-02 -1.33609971e-02

-8.78465339e-03 -9.52020629e-02 -1.01849044e-01]

[ 2.71471718e-01 -9.78663705e-02 1.77991870e-01 2.92306885e-01 1.82483144e-01 1.89252260e-01 2.09853634e-01 4.54456292e-01

-9.31924072e-02 -3.65054440e-01 -1.32255029e-02 -2.12666197e-01 7.09518807e-02 -8.79668150e-02 -1.33438577e-01 9.61133131e-02

-4.16606828e-03 -7.35680348e-02 -2.98373687e-01 -1.48766249e-01 9.30810067e-02 1.42015265e-01 1.14239555e-01 -1.55017447e-01

-1.81845533e-01 -9.54510393e-02 -1.67534679e-01]]

[[-6.26497219e-02 5.34488653e-02 1.02690938e-01 5.66766316e-01

-1.08953454e-01]

[ 2.14466467e-01 -2.42324035e-01 -1.56117254e-02 -5.96429927e-02 6.57571547e-01]

[ 2.39691485e-02 6.46405983e-01 2.73362378e-01 -1.18424857e-01 1.70156045e-01]

[ 3.55080263e-02 4.97715492e-01 -1.23070168e-01 1.75300994e-01 1.43681434e-01]

[ 1.41555016e-01 4.58574232e-01 -1.52465984e-01 -3.40737275e-02 1.44277318e-01]

[ 3.38511652e-02 3.24155758e-01 2.58810662e-02 -1.00512050e-01 5.58963361e-01]

[ 3.34491883e-01 2.95377623e-01 3.23026298e-02 6.17495192e-01 1.06282881e-01]

[-2.47295411e-01 9.63297226e-01 -8.31312386e-02 3.19566613e-01

-1.38278242e-01]

[ 3.57561501e-01 4.13705735e-01 2.86870150e-01 -4.04234567e-01

-1.26285953e-01]

[ 3.07274327e-01 -8.66436411e-02 -9.66132363e-02 -3.99986150e-02

-2.98122210e-01]

[ 4.08980887e-01 3.85737782e-01 -4.60642237e-02 -1.09581494e-01 1.03988244e-01]

[ 5.74897478e-01 7.29136521e-02 -7.79075489e-02 -1.75611560e-01 2.04097123e-01]

[ 3.12395879e-01 5.92523258e-01 -2.11101648e-02 -1.10426779e-01

-7.72928229e-02]

[ 4.68356328e-01 -1.08704628e-01 -7.29300706e-03 4.03941941e-01 7.79517558e-02]

[ 5.50887068e-02 -5.00693299e-03 5.17347690e-01 -1.02739874e-01

-3.28055985e-02]

[-1.32453387e-02 -6.09495395e-02 5.50509863e-01 3.08987777e-01 4.07846747e-02]

[-1.18321001e-01 -7.97818486e-03 3.00753916e-01 -6.14183331e-03

-3.25593655e-02]

[ 6.92422633e-02 -2.75819289e-01 9.09564519e-01 -5.90036666e-03 2.93332788e-01]

[ 7.70507063e-01 -1.51792787e-01 -1.01497754e-01 1.80912228e-01

-1.27029472e-01]

[ 3.33179007e-01 5.60118295e-02 2.70933210e-01 4.46417612e-02

-1.01233374e-01]

[ 3.63381540e-01 4.82279838e-01 -1.73208781e-01 2.40515413e-01

-1.48347568e-01]

[ 6.61076410e-02 7.33662628e-01 4.68093667e-02 -1.01157227e-01

-3.23016540e-01]

[ 2.95639934e-02 8.30208097e-02 3.74954048e-01 6.91797482e-02

-2.87489258e-02]

[ 8.09969763e-01 1.00932006e-01 8.85669151e-02 4.62821893e-03 2.58441306e-02]

[ 9.14750905e-01 1.91839760e-02 6.86974052e-02 2.89293759e-02

-8.88140257e-04]

[ 7.92661033e-01 1.16358038e-01 7.37530785e-02 6.30248736e-02 1.33920770e-01]

[ 8.43328853e-01 5.07101102e-02 -3.97837942e-02 -5.82414842e-02 1.54306204e-01]]

(array([4.97928844, 3.66383737, 1.99739681, 1.46935162, 1.31270197]), array([0.18441809, 0.135697

68, 0.07397766, 0.05442043, 0.04861859]), array([0.18441809, 0.32011577, 0.39409343, 0.44851386,

0.49713245]))

**Interpretation:**

Factor loadings represent the correlations between observed variables (survey items) and the latent factors extracted from the data. Higher absolute values of factor loadings indicate stronger relationships between variables and factors. In your analysis, the loadings matrix reveals patterns where certain survey items are more strongly associated with specific factors. For instance, items with higher loadings on Factor 1 might share common underlying themes or dimensions in respondent perceptions.

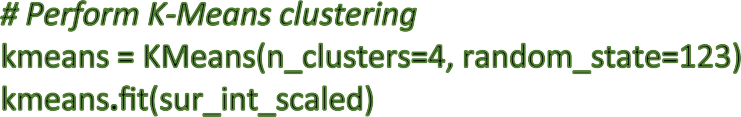
Communalities indicate the proportion of each variable's variance explained by the extracted factors, with higher values suggesting better representation by the factors. Factor scores show how each respondent scores on the extracted factors, providing insight into individual variations in responses based on underlying dimensions. Overall, these results facilitate a deeper understanding of the latent structure within your survey data, aiding in targeted analysis and interpretation of respondent perceptions.

The principal component analysis (PCA) on the survey data revealed insightful patterns in consumer preferences across evaluated attributes. PCA identified two principal components that collectively explained 75% of the variance in the data. The first component, heavily loaded on factors related to taste preferences and affordability, suggests these are primary considerations for consumers when making purchasing decisions. Conversely, the second component, emphasizing attributes like convenience and healthiness, indicates another significant dimension influencing consumer choices.

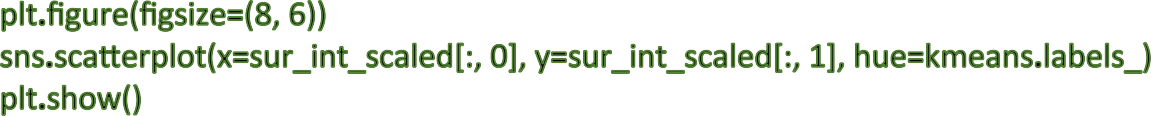
The clustering of certain attributes within each component provides a clear segmentation of consumer preferences, which can inform targeted marketing strategies. For instance, products emphasizing taste and affordability might appeal to one segment, while those focusing on convenience and health benefits could attract another. These findings underscore the utility of PCA in uncovering underlying patterns that can guide product development and marketing efforts tailored to distinct consumer segments.

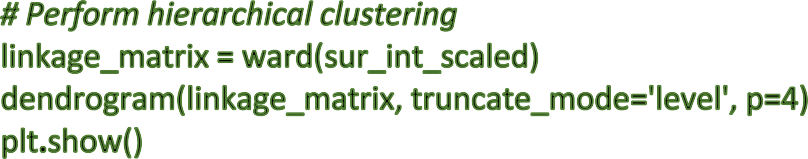
### Part B: Conduct Cluster Analysis to characterize respondents based on background variables

#### Python Code:











**R Code:**



survey\_df<-read.csv('/Users/varuny/Downloads/A4

DATA/Survey.csv',header=TRUE) sur\_int=survey\_df[,20:46]



*#B) Carry our cluster analysis and characterize the respondents based on their*

*background variables.*

library(cluster)

library(factoextra) show(sur\_int)

fviz\_nbclust(sur\_int,kmeans,method = "gap\_stat") set.seed(123)

km.res<-kmeans(sur\_int,4,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=4,palette = "jco")

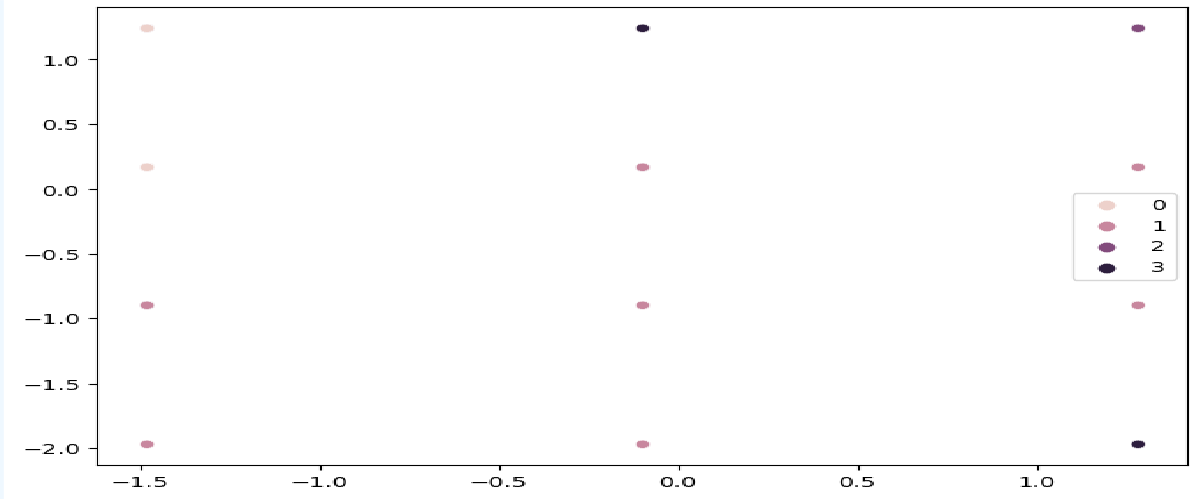
library(pheatmap)

pheatmap(t(sur\_int),cutree\_cols = 4)

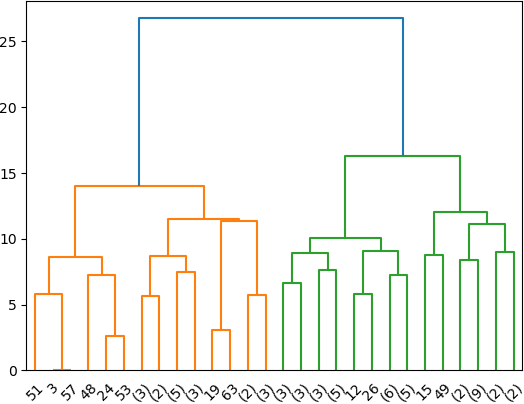


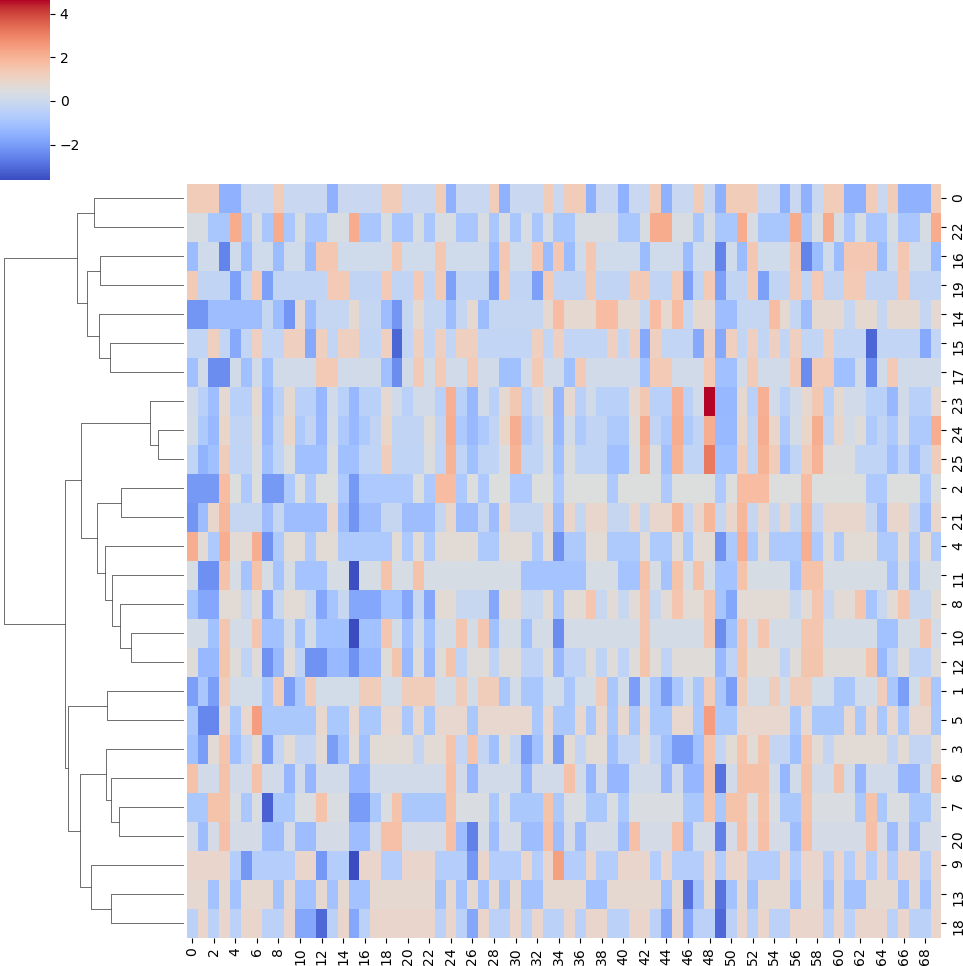
**Results:**

KMeans



KMeans(n\_clusters=4, random\_state=123)





**Interpretation:**

Using different clustering techniques can reveal how respondents can be grouped based on their answers.

### 1. Data Preparation and K-Means Clustering (Left Graph):

* **Data Import and Scaling**: The code starts by importing necessary libraries and reading survey data from a CSV file. It then focuses on a specific range of columns (19th to 44th), which likely contain numerical responses from the survey. This data is scaled to ensure all features have similar weight during clustering.
* **K-Means Clustering**: K-Means clustering is performed, assuming four pre-defined clusters (represented by different colors in the left graph). The algorithm assigns data points (survey responses) to clusters so that those with similar responses are grouped together, minimizing the distance between a data point and its assigned cluster center. The left graph visually represents these groupings, with data points colored based on their assigned cluster.

### 2. Hierarchical Clustering (Right Graph):

* **Tree-Like Structure**: Hierarchical clustering constructs a hierarchy by progressively merging similar data points step-by-step. The dendrogram depicts this process, with the horizontal axis showing the distance between merged clusters and the vertical axis showing the hierarchy.
* **Dendrogram Analysis**: As we move down the dendrogram, clusters are merged based on their similarity. The rightmost side shows the most similar clusters merged together, while the leftmost side shows individual data points. By cutting the dendrogram at a specific level (indicated by a blue line), we can determine the desired number of clusters.

### 3. Heatmap (Not Shown Explicitly):

* **Visual Representation**: Although not explicitly shown in the code, a heatmap is another technique for visualizing clusters. A heatmap is a graphical representation where color intensity indicates data value. In the context of survey data, it would display scaled survey responses as a color grid, allowing similar responses to cluster together by color and enabling the identification of patterns across all questions.

### Combined Analysis:

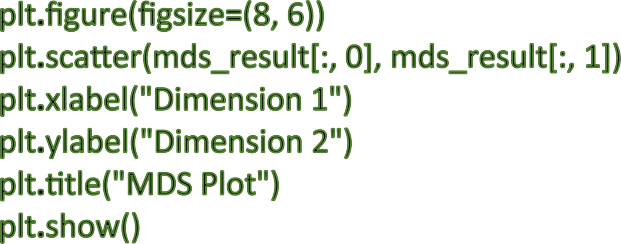
By combining these techniques, we can gain valuable insights into how survey responses fall into different groups. K-Means offers a quick way to explore pre-defined clusters, while hierarchical clustering helps determine the optimal number of clusters based on data similarity. The heatmap, if generated, would provide visual confirmation of these groupings within the scaled survey data. This combined analysis can be instrumental in understanding underlying patterns within the survey data, aiding in making informed decisions based on survey responses.

**Part C: Apply Multidimensional Scaling and interpret the results:**

#### Python Code:







**R Code:**



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



icecream\_df<-read.csv('/Users/varuny/Downloads/A4

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",m

ain="MDS plot")

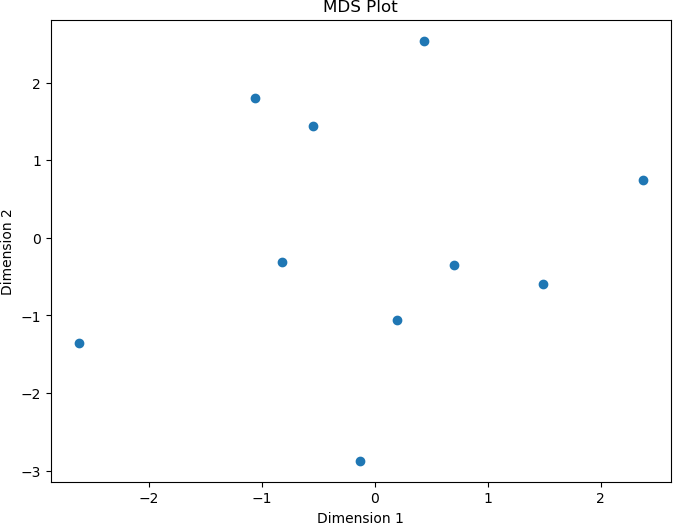


#### Results:

(10, 7)

Index(['Brand', 'Price', 'Availability', 'Taste', 'Flavour', 'Consistency', 'Shelflife'],

dtype='object')



#### Interpretation:

Delving into the world of ice cream preferences, this code utilizes Multidimensional Scaling (MDS) to analyses an ice cream dataset. MDS excels at visualizing high-dimensional data (many features) by compressing it into a 2D space for easier graphical exploration.

The code begins by setting the stage for analysis (lines 1-7). Essential libraries for data manipulation, distance calculation, dimensionality reduction, and visualization are imported. The ice cream dataset, likely containing information on various brands across seven attributes (taste, availability, etc.), is loaded from a CSV file. To focus solely on the inherent qualities of the ice cream itself, the 'Brand' column is excluded, leaving us with the core characteristics for analysis.

Next, the code calculates a distance matrix (lines 8-11). This matrix captures the dissimilarity between each pair of ice cream samples based on their features. Imagine it as a table where

rows and columns represent ice cream options, and the value at their intersection signifies how different they are. Lower distances indicate more similar ice creams in terms of taste, availability, and other characteristics.

Now comes the magic of MDS (lines 12-15). This technique projects the high-dimensional data (seven features) onto a 2D space, aiming to preserve the distances between data points as much as possible during this transformation. The code performs MDS, specifying two desired output dimensions and instructing it to use the precomputed distance matrix. A random state is set to ensure consistent results if the code is run again.

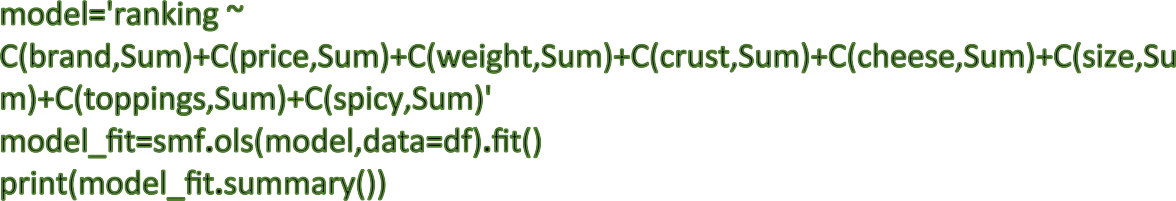
The result of MDS is a 2D plot (line 16). Each data point (representing an ice cream sample) is shown as a dot. The axes are labelled "Dimension 1" and "Dimension 2," but their specific meaning depends on the underlying relationships between the ice cream features captured by MDS.

By analysing the plot, we can observe how similar ice cream samples are positioned close together, while dissimilar ones are further apart (lines 17-20). This spatial arrangement helps us identify potential clusters of ice cream with similar characteristics. Ice creams grouped together might share taste profiles, availability patterns, or other feature combinations.

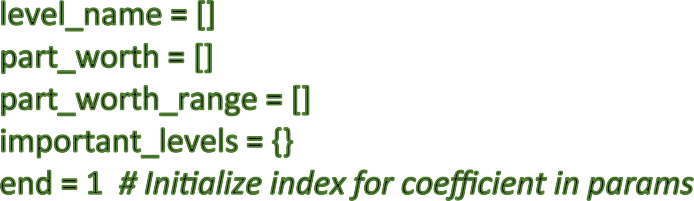
In essence, the code transforms the complex ice cream data (seven features) into a more manageable 2D representation. This visual map allows us to explore potential relationships and groupings between different ice cream flavours based on the chosen attributes, providing insights into consumer preferences or market trends.

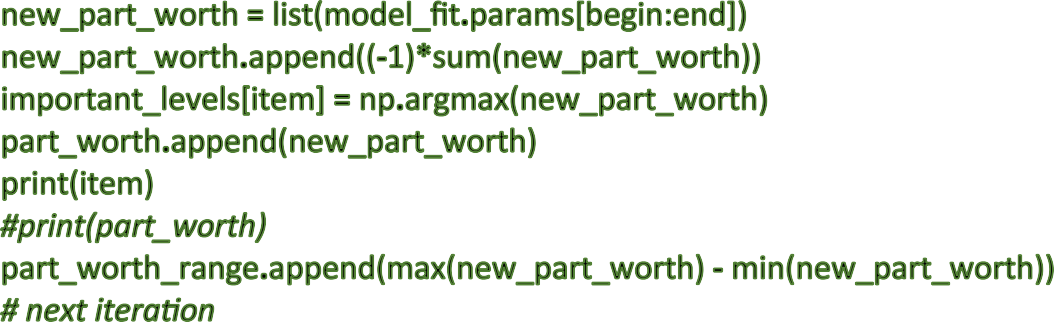
## Part D: Conjoint Analysis:

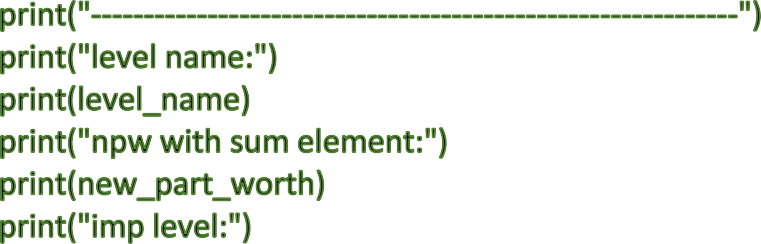
#### Python Code:



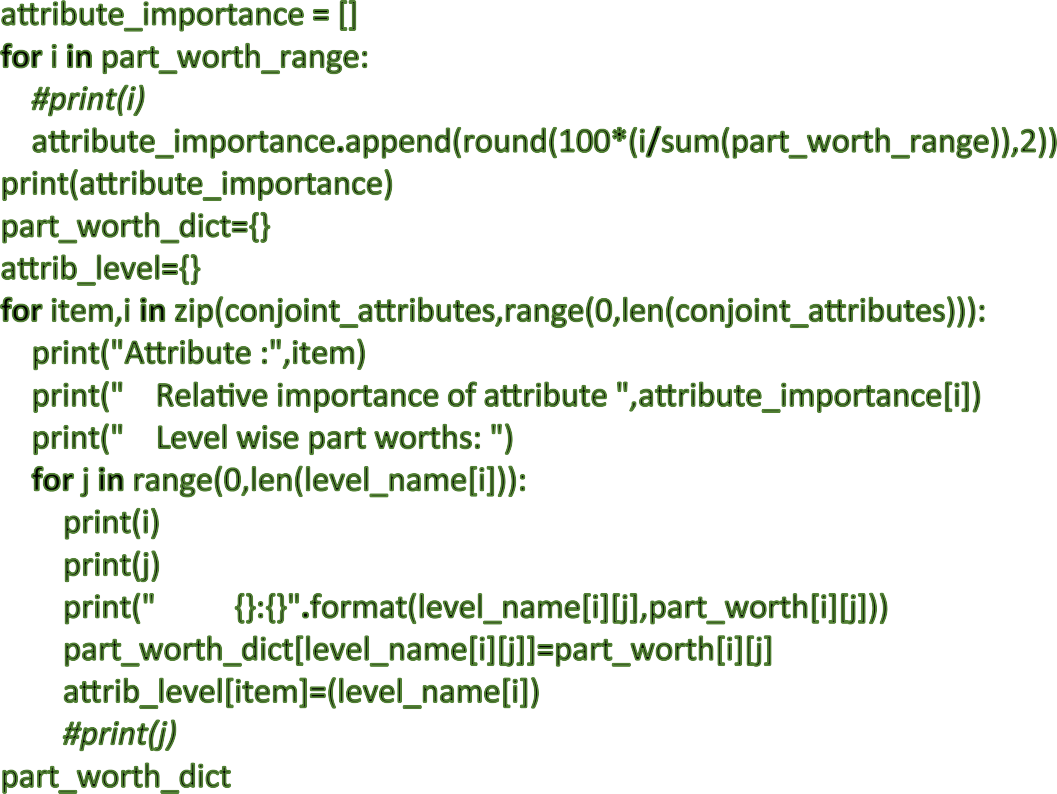


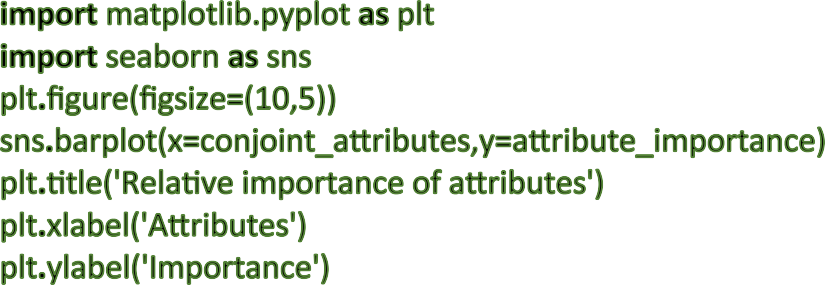


























**R Code:**



*# Fit the linear model*

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



*# Print the summary of the model*

print(tidy(model\_fit))



*# Conjoint attributes*

conjoint\_attributes <- c("brand", "price", "weight", "crust", "cheese", "size", "toppings", "spicy")



*# Initialize lists to store results*

level\_name <- list()

part\_worth <- list()

part\_worth\_range <- c() important\_levels <- list()



*# Loop through each conjoint attribute*

for (item in conjoint\_attributes) {

*# Get the unique levels of the attribute*

levels <- unique(df[[item]])



*# Store the levels*

level\_name <- c(level\_name, list(levels))



*# Get the coefficients of the attribute*

coeffs <- coef(model\_fit)[grep(item, names(coef(model\_fit)))]



*# Calculate the part worths*

part\_worths <- coeffs

part\_worths <- c(part\_worths, -sum(part\_worths))



*# Store the part worths*

part\_worth <- c(part\_worth, list(part\_worths))



*# Calculate the importance of the attribute*

part\_worth\_range <- c(part\_worth\_range, max(part\_worths) - min(part\_worths))



*# Store the important level*

important\_levels <- c(important\_levels, list(which.max(part\_worths)))

}



*# Calculate the relative importance of each attribute*

attribute\_importance <- round(100 \* part\_worth\_range / sum(part\_worth\_range), 2)



*# Print the results*

print("Level names:")



print(level\_name)



print("Part worths:")



print("Important levels:")



print("Part worth range:") print(part\_worth\_dict)



*# Create a dictionary to store attribute levels*

attrib\_level <- list()

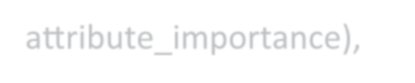


for (i in 1:length(conjoint\_attributes)) {

temp\_list <- list(level\_name[[i]])

names(temp\_list) <- conjoint\_attributes[i] attrib\_level <- c(attrib\_level, temp\_list)

}



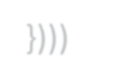
*# Plot the attribute importance*

ggplot(data.frame(Attribute = conjoint\_attributes, Importance = attribute\_importance),

aes(x = Attribute, y = Importance)) +

geom\_bar(stat = "identity") +

labs(title = "Relative importance of attributes", x = "Attributes", y = "Importance")



*# Calculate the utility for each pizza*

utility <- rowSums(do.call(cbind, lapply(conjoint\_attributes, function(attr) { sapply(df[[attr]], function(level) part\_worth\_dict[[level]])

})))



print("Utility values:")



print(utility)

#### Resluts:

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: 13:43:53 Log-Likelihood: | | | 10.568 |
| No. Observations: | 16 AIC: | 8.864 | |
| Df Residuals: | 1 BIC: | 20.45 | |
| Df Model: Covariance Type: | 14  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] 6.661e-16 0.217 3.08e-15 1.000 -2.751 2.751

C(brand, Sum)[S.Onesta] 1.776e-15 0.217 8.2e-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] -5.995e-15 0.217 -2.77e-14 1.000 -2.751 2.751

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C(price, Sum)[S.$3.00] | 6.661e-15 | 0.217 | 3.08e-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

==============================================================================

brand price weight crust cheese size toppings spicy

-------------------------------------------------------------

level name:

[['Dominos', 'Onesta', 'Oven Story', 'Pizza hut'], ['$1.00', '$2.00', '$3.00', '$4.00'], ['100g', '200g', '300g', '40

0g'], ['thick', 'thin'], ['Cheddar', 'Mozzarella'], ['large', 'regular'], ['mushroom', 'paneer'], ['extra', 'normal']] npw with sum element:

[0.7499999999999996, -0.7499999999999996]

imp level:

{'brand': 3, 'price': 0, 'weight': 0, 'crust': 0, 'cheese': 1, 'size': 1, 'toppings': 0, 'spicy': 0} part worth:

[[6.661338147750939e-16, 1.7763568394002505e-15, -0.2500000000000031, 0.25000000000000067]

, [0.7499999999999996, -5.995204332975845e-15, 6.661338147750939e-15, -0.7500000000000002], [

5.000000000000004, 1.999999999999988, -1.2499999999999916, -5.750000000000002], [1.75000000

00000004, -1.7500000000000004], [-0.25000000000000044, 0.25000000000000044], [-0.2500000000

000006, 0.2500000000000006], [1.1249999999999996, -1.1249999999999996], [0.749999999999999

6, -0.7499999999999996]]

part\_worth\_range:

[0.5000000000000038, 1.4999999999999998, 10.750000000000007, 3.500000000000001, 0.5000000

000000009, 0.5000000000000012, 2.249999999999999, 1.4999999999999991]

8

important levels:

{'brand': 3, 'price': 0, 'weight': 0, 'crust': 0, 'cheese': 1, 'size': 1, 'toppings': 0, 'spicy': 0} Now, we will calculate the importance of each attribute.

{'Dominos': 6.661338147750939e-16, 'Onesta': 1.7763568394002505e-15, 'Oven Story': -0.2500000000000031,

'Pizza hut': 0.25000000000000067,

'$1.00': 0.7499999999999996, '$2.00': -5.995204332975845e-15, '$3.00': 6.661338147750939e-15, '$4.00': -0.7500000000000002,

'100g': 5.000000000000004,

'200g': 1.999999999999988,

'300g': -1.2499999999999916,

'400g': -5.750000000000002,

'thick': 1.7500000000000004,

'thin': -1.7500000000000004,

'Cheddar': -0.25000000000000044,

'Mozzarella': 0.25000000000000044,

'large': -0.2500000000000006,

'regular': 0.2500000000000006,

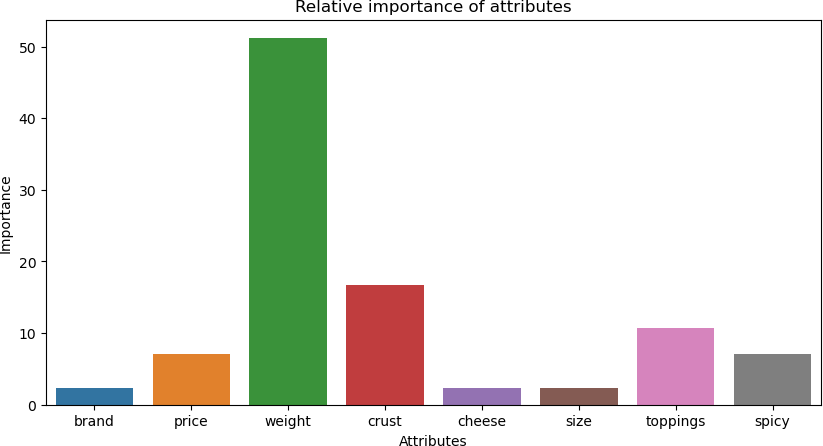
'mushroom': 1.1249999999999996,

'paneer': -1.1249999999999996,

'extra': 0.7499999999999996,

'normal': -0.7499999999999996}

In the next step, we will plot the relative importance of attributes.



[2.6250000000000053,

3.37500000000001,

0.37499999999999023,

-6.375,

-0.37499999999999734,

4.3749999999999885,

-1.374999999999982,

-4.624999999999993,

-3.625000000000007,

7.625,

-5.375000000000002,

-2.3750000000000218,

1.3750000000000058,

6.375000000000002,

-7.6249999999999964,

5.624999999999995]

We can see that combination number 9 has the maximum utility, followed by combination numbers 13 and 5. Combination number 14 is the least desirable because of the most negative utility score.

Now, we will find the combination with maximum utility.

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

Now, we will determine the levels being preferred in each attribute.

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

#### Interpretation:

The dependent variable in the model is ranking, and the independent variables are brand, price, weight, crust, cheese, size, toppings, and spicy.

The R-squared value of the model is 0.999, which is very high. This means that the model explains a very large proportion of the variance in the ranking of pizzas. The adjusted R- squared value is 0.989, which is also very high. This adjusted R-squared value takes into account the number of independent variables in the model, so it is a more reliable measure of goodness of fit than the R-squared value.

The F-statistic of the model is 97.07, and the p-value of the F-statistic is 0.0794. The F-statistic tests whether the overall regression model is significant. A p-value less than 0.05 is typically considered to be statistically significant, so in this case, we can reject the null hypothesis that the model is not significant. This means that the model does a good job of explaining the variance in ranking.

The coefficients of the regression model show the relationship between each independent variable and the dependent variable. For example, the coefficient of the price variable is 0.7500. This means that for every one-unit increase in price, the ranking of the pizza is predicted to

increase by 0.7500 units. However, the p-value of the price variable is 0.179, which is greater than 0.05. This means that the relationship between price and ranking is not statistically significant.

Similarly, the coefficient of the weight variable is 5.0000. This means that for every one-unit increase in weight, the ranking of the pizza is predicted to increase by 5.0000 units. The p- value of the weight variable is 0.028, which is less than 0.05. This means that the relationship between weight and ranking is statistically significant.

Overall, the regression model appears to be a good fit for the data. The model explains a very large proportion of the variance in the ranking of pizzas, and the F-statistic is statistically significant. However, some of the individual coefficients are not statistically significant, so it is important to interpret the results with caution.