Deep Shrivastav
Mini Project for Multivariate Analysis

MARKET BASKET AND CLUSTER ANALYSIS ON BIG BASKET DATASET

BigBasket Overview

BigBasket is India's leading online grocery store, offering a vast array of products including fresh produce, meat, dairy, bakery items, and household essentials. With a robust supply chain network and partnerships with local farmers, it ensures the quality and freshness of its offerings. Customers can conveniently order through its website or app, with various payment options and delivery choices tailored to their needs. Membership plans like BB Star provide additional perks such as free delivery and exclusive discounts, making BigBasket a preferred choice for online grocery shopping in India.

About Dataset

E-commerce (electronic commerce) is the activity of electronically buying or selling of products on online services or over the Internet. E-commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management

Bigbasket is the largest online grocery supermarket in India. Was launched somewhere around in 2011 since then they've been expanding their business. Though some new competitors have been able to set their foot in the nation such as Blinkit etc. but BigBasket has still not lose anything - thanks to ever expanding popular base and their shift to online buying.

Summary of Dataset:

index: An integer variable representing the index or identifier of each observation.
product: A character variable indicating the name of the product.
category: A character variable indicating the broad category or department to which the product belongs (e.g., "Beauty & Hygiene", "Kitchen, Garden & Pets").
sub_category: A character variable providing further classification within the category (e.g., "Hair Care", "Storage & Accessories").
brand: A character variable indicating the brand of the product.
sale_price: A numeric variable representing the sale price of the product.
market_price: A numeric variable representing the market price or original price of the product.
rating: A numeric variable representing the rating or customer satisfaction score of the product.
description: A character variable containing a description or additional information about the product.

#glimpse of Dataset

summary(data)
 # Correlation matrix
correlation_matrix <- cor(data[, c("sale_price", "market_price", "rating")], use = "pairwise.complete.obs")
correlation_matrix

```
> summary(data)
     index          product            category         sub_category          brand            sale_price        market_price
 Min.   :    1   Length:27555       Length:27555       Length:27555       Length:27555       Min.   :    2.45   Min.   :    3.0
 1st Qu.: 6890   Class :character   Class :character   Class :character   Class :character   1st Qu.:   95.00   1st Qu.:  100.0
 Median :13778   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :  190.00   Median :  220.0
 Mean   :13778                                                                               Mean   :  322.51   Mean   :  382.1
 3rd Qu.:20667                                                                               3rd Qu.:  359.00   3rd Qu.:  425.0
 Max.   :27555                                                                               Max.   :12500.00   Max.   :12500.0

     type               rating       description
 Length:27555       Min.   :1.000   Length:27555
 Class :character   1st Qu.:3.700   Class :character
 Mode  :character   Median :4.100   Mode  :character
                    Mean   :3.943
                    3rd Qu.:4.300
                    Max.   :5.000
                    NA's   :8626
> |
```

```
> correlation_matrix
              sale_price market_price      rating
sale_price    1.00000000   0.96519801 -0.07928513
market_price  0.96519801   1.00000000 -0.09498883
rating       -0.07928513  -0.09498883  1.00000000
```

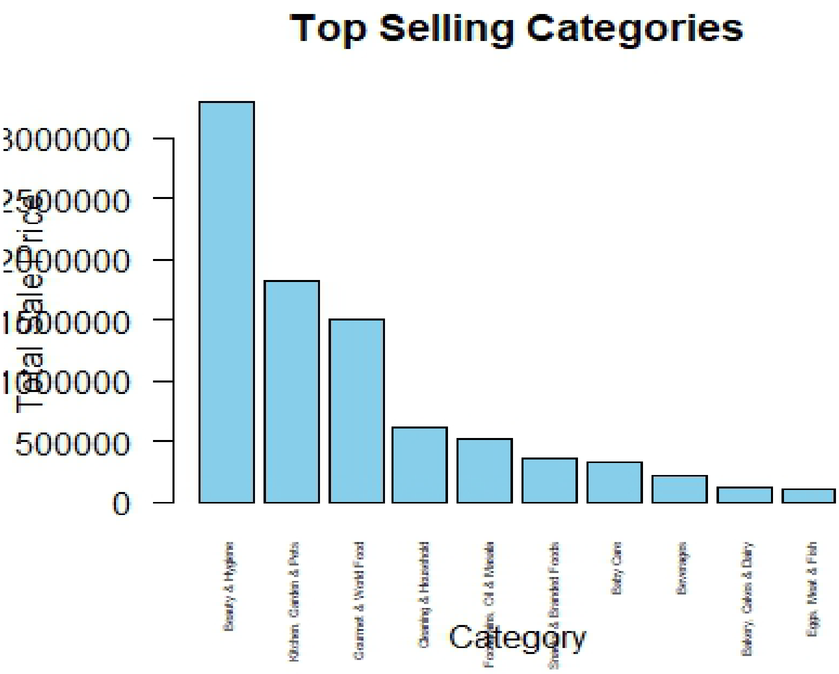
	A	B	C	D	E	F	G	H	I	J	
1	Index	Invoice No.	Product	Category	Sub_category	Brand	Sales_price	Market_price	Type	Rating	Description
2	1	495280	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220	220	Hair Oil & Serum	4.1	This Product contains Garlic Oil that is known to help proper digestion,
3	2	493873	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180	180	Water & Fridge Bottles	2.3	Each product is microwave safe (without lid), refrigerator safe, dishwas
4	3	492028	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119	250	Lamp & Lamp Oil	3.4	A perfect gift for all occasions, be it your mother, sister, in-laws, boss o
5	4	489736	Cereal Flip Lid Container/Storage Jar - Assorted Colour	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149	176	Laundry, Storage Baskets	3.7	Multipurpose container with an attractive design and made from food-
6	5	493414	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162	162	Bathing Bars & Soaps	4.4	Nivea Creme Soft Soap gives your skin the best care that it must get. Th
7	6	491245	Germ - Removal Multipurpose Wipes	Cleaning & Household	All Purpose Cleaners	Nature Protect	169	199	Disinfectant Spray & Cleaners	3.3	Stay protected from contamination with Multipurpose Germ Removal W
8	7	493492	Multani Mati	Beauty & Hygiene	Skin Care	Satinance	58	58	Face Care	3.6	Satinance multani matti is an excellent skin toner and astringent. redu
9	8	492589	Hand Sanitizer - 70% Alcohol Base	Beauty & Hygiene	Bath & Hand Wash	Bionova	250	250	Hand Wash & Sanitizers	4	70%Alcohol based is gentle of hand leaves skin soft and moist with a
10	9	495520	Biotin & Collagen Volumizing Hair Shampoo + Biotin & Collagen Hair Conditioner	Beauty & Hygiene	Hair Care	StBotanica	1098	1098	Shampoo & Conditioner	3.5	An exclusive blend with Vitamin B7 Biotin, Hydrolyzed collagen, Oat Ext
11	10	495619	Scrub Pad - Anti- Bacterial, Regular	Cleaning & Household	Mops, Brushes & Scrubs	Scotch brite	20	20	Utensil Scrub-Pad, Glove	4.3	Scotch Brite Anti- Bacterial Scrub Pad thoroughly cleanses dishes and k

```
> str(data)
'data.frame':	27555 obs. of  10 variables:
 $ index       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ product     : chr  "Garlic Oil - Vegetarian Capsule 500 mg" "Water Bottle - Orange" "Brass Angle Deep - Plain, No.2" "Cereal Flip
Container/Storage Jar - Assorted Colour" ...
 $ category    : chr  "Beauty & Hygiene" "Kitchen, Garden & Pets" "Cleaning & Household" "Cleaning & Household" ...
 $ sub_category: chr  "Hair Care" "Storage & Accessories" "Pooja Needs" "Bins & Bathroom Ware" ...
 $ brand       : chr  "Sri Sri Ayurveda " "Mastercook" "Trm" "Nakoda" ...
 $ sale_price  : num  220 180 119 149 162 ...
 $ market_price: num  220 180 250 176 162 ...
 $ type        : chr  "Hair Oil & Serum" "Water & Fridge Bottles" "Lamp & Lamp Oil" "Laundry, Storage Baskets" ...
 $ rating      : num  4.1 2.3 3.4 3.7 4.4 3.3 3.6 4 3.5 4.3 ...
 $ description : chr  "This Product contains Garlic Oil that is known to help proper digestion, maintain proper cholesterol levels, s
 __truncated__ "Each product is microwave safe (without lid), refrigerator safe, dishwasher safe and can also be used for re-he"| __t
```

EXPLORATORY DATA ANAYSIS

Top selling categories

```
# Aggregate sales data by category and sum up the sale_price
category_sales <- aggregate(sale_price ~ category, data = data, FUN = sum)
category_sales <- category_sales[order(category_sales$sale_price, decreasing = TRUE),]
# Plot the top-selling categories
top_categories <- head(category_sales, 10)  # You can adjust the number as needed
# Plot the bar chart
barplot(top_categories$sale_price, names.arg = top_categories$category,
      main = "Top Selling Categories", xlab = "Category", ylab = "Total Sale Price",
      col = "skyblue", las = 2, cex.names = 0.8)
```

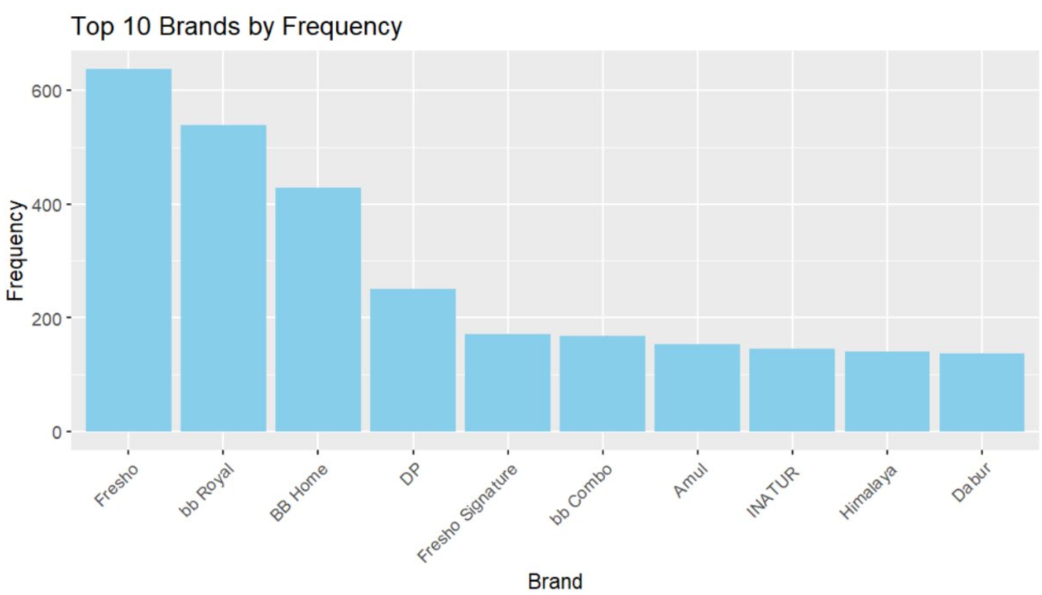


#top brand

```r
# Count the frequency of each brand
brand_frequency <- table(data$brand)
# Convert the frequency table to a data frame
brand_freq_df <- data.frame(brand = names(brand_frequency),
                            frequency = as.numeric(brand_frequency))
# Sort the data frame by frequency in descending order
brand_freq_df <- brand_freq_df[order(brand_freq_df$frequency, decreasing = TRUE),]
# Select only the top 10 brands
top_10_brands <- head(brand_freq_df, 10)
print(top_10_brands)
```

```
> print(top_10_brands)
                brand frequency
750            Fresho       638
210          bb Royal       539
207           BB Home       428
589                DP       250
752 Fresho Signature       171
206          bb Combo       168
98               Amul       153
979            INATUR       146
931          Himalaya       141
508             Dabur       138
```

```r
# Plot brand vs frequency for top 10 brands
ggplot(top_10_brands, aes(x = reorder(brand, -frequency), y = frequency)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Top 10 Brands by Frequency", x = "Brand", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

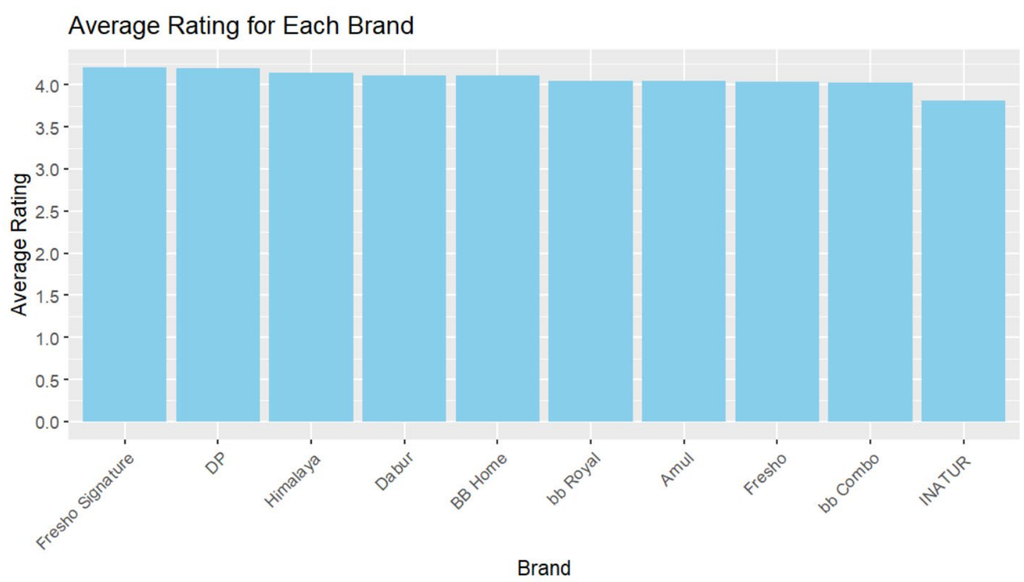


#Average Rating for each Brand

```
# Extract the brand names from the top_10_brands data frame
top_brands <- top_10_brands$brand
# Filter the original data for the top brands and calculate average rating
brand_avg_rating <- data %>%
  filter(brand %in% top_brands) %>%
  group_by(brand) %>%
  summarise(avg_rating = mean(rating, na.rm = TRUE))
# Print the average rating for the top brands
print(brand_avg_rating)
```

```
> print(brand_avg_rating)
# A tibble: 10 × 2
   brand                avg_rating
   <chr>                     <dbl>
 1 "Amul"                     4.04
 2 "BB Home"                  4.11
 3 "DP"                       4.2
 4 "Dabur"                    4.11
 5 "Fresho"                   4.03
 6 "Fresho Signature"         4.21
 7 "Himalaya"                 4.14
 8 "INATUR "                  3.81
 9 "bb Combo"                 4.02
10 "bb Royal"                 4.05
```

```
# Plot brand vs average rating
ggplot(brand_avg_rating, aes(x = reorder(brand, -avg_rating), y = avg_rating)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Rating for Each Brand", x = "Brand", y = "Average Rating") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(breaks = seq(0, max(brand_avg_rating$avg_rating) + 0.5, by = 0.5))
```



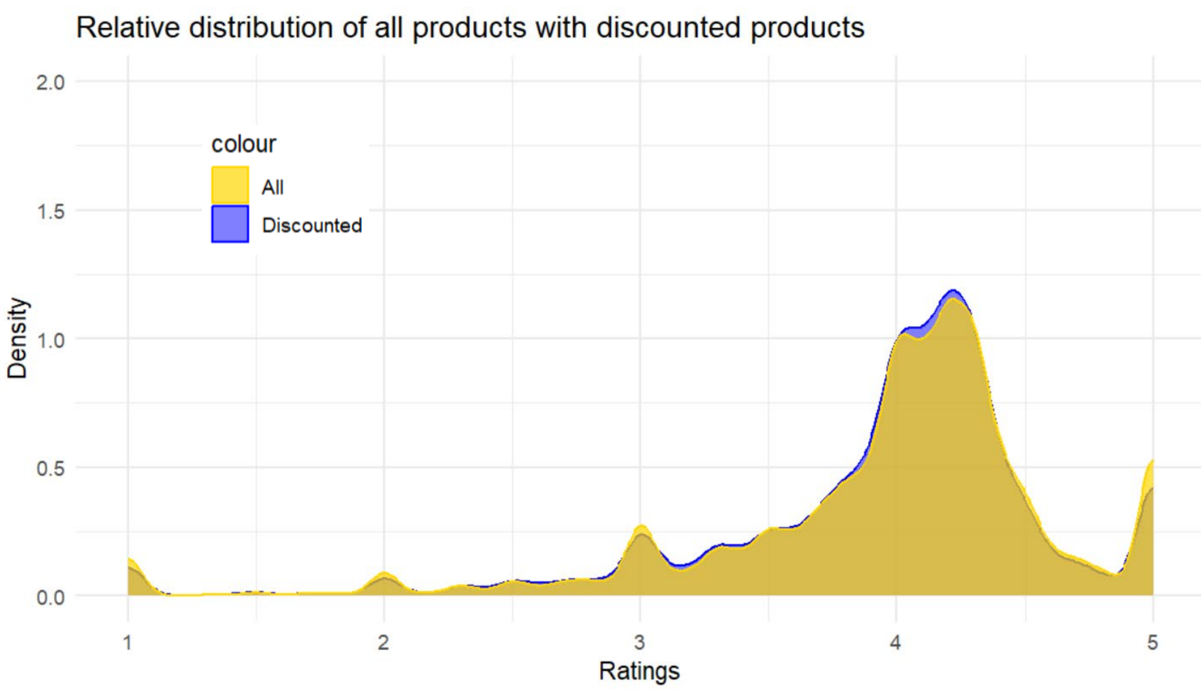
Distrubution of Sale and Market Price

```
library(ggplot2)

ggplot(data, aes(x = sale_price)) +
  geom_histogram(aes(fill = "Sale Price"), binwidth = 50, alpha = 1) + # Adjust
transparency
  geom_histogram(aes(x = market_price, fill = "Market Price"), binwidth = 50, alpha = 0.5)
+ # Adjust transparency
  labs(title = "Distribution of Sale Price and Market Price", x = "Price", y = "Frequency") +
  scale_fill_manual(values = c("Sale Price" = "skyblue", "Market Price" = "red")) +
  guides(fill = guide_legend(title = "Price Type"))
```

We can see product near and above 3000 have good profit as compare to other

Relative distribution of all products with discounted products

```
# Calculate the difference in prices
data$diff_in_prices <- data$market_price - data$sale_price
# Filter for products with discounts
discount <- data[data$diff_in_prices != 0, ]
# Create a density plot for ratings
ggplot() +
  geom_density(data = discount, aes(x = rating), color = 'blue', fill = 'blue', alpha = 0.5) +
  geom_density(data = data, aes(x = rating), color = 'gold', fill = 'gold', alpha = 0.7) +
  labs(x = "Ratings", y = "Density", title = "Relative distribution of all products with
discounted products") +
  theme_minimal() +
  ylim(0, 2)
```



WHAT IS MARKET BASKET ANALYSIS?

Market Basket Analysis (MBA) is a data mining technique used in retail and e-commerce to discover associations between items purchased together in transactions. The primary goal of Market Basket Analysis is to identify patterns of co-occurrence or co-purchase among items, which can provide valuable insights for improving business strategies. Here's a detailed explanation of Market Basket Analysis:

Association Rule Mining:
Market Basket Analysis is a form of association rule mining, which is a data mining technique that aims to discover relationships or associations between variables in large datasets.

Frequent Itemsets:
The core concept of MBA is to identify frequent itemsets, which are combinations of items that occur together frequently in transactions.
For example, if customers often purchase bread and milk together, "bread" and "milk" form a frequent itemset.

Association Rules:
Once frequent itemsets are identified, association rules are generated to express relationships between items.
An association rule typically takes the form of "If {A}, then {B}", where A and B are sets of items. For example, "If {bread}, then {milk}".
Each rule is associated with measures like support, confidence, and lift, which quantify the strength and significance of the association.
the strength of the association between A and B.
Lift measures the strength of association between items and indicates how much more likely item B is purchased when item A is purchased, compared to when item B is purchased independently of item A.

Applications:
MBA has various applications in retail and e-commerce, including product recommendations, market segmentation, pricing strategies, inventory management, and cross-selling and upselling.

Benefits:
Helps retailers understand purchasing patterns and customer preferences.
Enables personalized marketing strategies and targeted promotions.
Optimizes product placement and inventory management.
Enhances customer satisfaction and loyalty by offering relevant product recommendations.
Overall, Market Basket Analysis is a powerful technique for extracting actionable insights from transactional data, driving business growth, and enhancing customer experience in retail and e-commerce industries.

#Loading libraries and data

```
library(arules)

# Read the CSV file into a data frame
file_path <- "C:\\Users\\shriv\\Downloads\\r project\\BigBasket Data.csv"
data <- read.csv(file_path)
str(data)
```

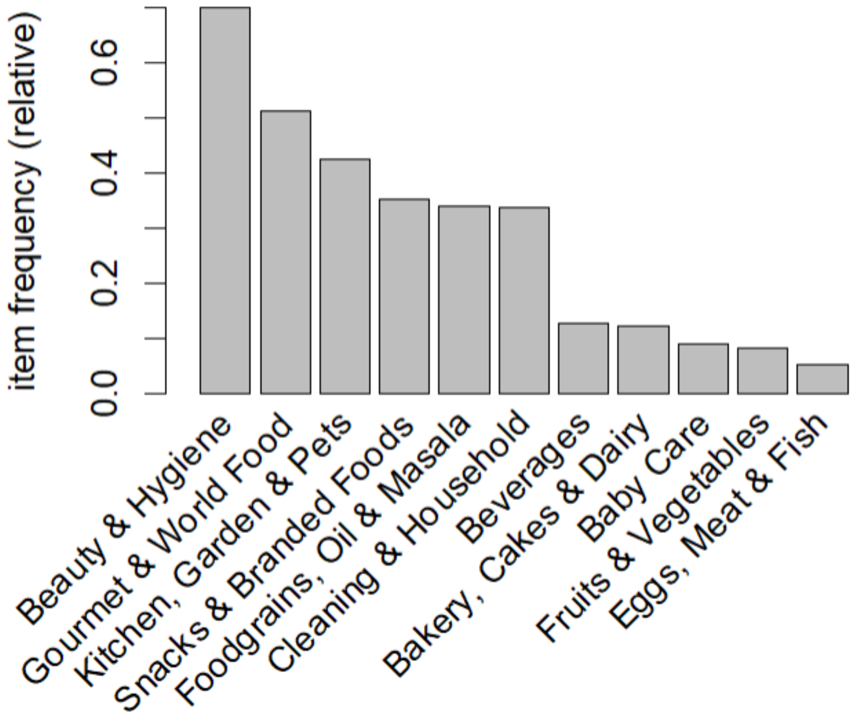
#Since Apriori algorithm works with transaction data , we will first extract the invoice number and category feature in another csv file

```
# Ensure 'Invoice No' and 'Product' are character vectors
data$Invoice.No <- as.factor(as.character(data$Invoice.No))
data$Category <- as.factor(data$Category)

# Extract relevant columns
data_new <- data[, c("Invoice.No", "Category")]
str(data_new)
data_new=na.omit(data_new) #omiting missing data

#write csv
write.csv(data_new,"data3.csv",row.names = FALSE)
```

	A	B	C	D	E
1	Invoice.No	Category			
2	495280	Beauty & Hygiene			
3	493873	Kitchen, Garden & Pets			
4	492028	Cleaning & Household			
5	489736	Cleaning & Household			
6	493414	Beauty & Hygiene			
7	491245	Cleaning & Household			
8	493492	Beauty & Hygiene			
9	492589	Beauty & Hygiene			
10	495520	Beauty & Hygiene			
11	495619	Cleaning & Household			
12	495112	Gourmet & World Food			
13	495633	Gourmet & World Food			
14	490472	Beauty & Hygiene			
15	493511	Cleaning & Household			
16	491883	Cleaning & Household			
17	492707	Cleaning & Household			
18	492419	Beauty & Hygiene			
19	493697	Gourmet & World Food			
20	490745	Gourmet & World Food			
21	494147	Gourmet & World Food			

#now the the dataset is ready for apriori

```
itemFrequencyPlot(transactions, support = 0.10)
```




```
# Read the CSV file without specifying columns

transactions <- read.transactions(file="data3.csv",
format="single", sep=",", cols=c(1,2), skip=1)

inspect(transactions[1:5])
```

```
> transactions <- read.transactions(file="data3.csv", format="single", sep
(1,2), skip=1)
> inspect(transactions[1:5])
    items                        transactionID
[1] {Beauty & Hygiene}                  489434
[2] {Beauty & Hygiene,
     Cleaning & Household,
     Gourmet & World Food}              489435
[3] {Beauty & Hygiene}                  489436
[4] {Gourmet & World Food,
     Kitchen, Garden & Pets}            489437
[5] {Beauty & Hygiene,
     Eggs, Meat & Fish}                 489438
>
```

```
# Perform association rule mining using Apriori algorithm

rules <- apriori(transactions,

        parameter = list(support = 0.01, confidence = 0.50))
```

```
>   rules <- apriori(transactions,
+                    parameter = list(support = 0.01, confidence = 0.50))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
        0.5    0.1    1 none FALSE            TRUE       5    0.01      1     10
 target  ext
  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 65

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 6505 transaction(s)] done [0.00s].
sorting and recoding items ... [11 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [131 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

INTREPETATION

WE CAN SEE 6505 UNIQUE TRANSACTIONS

131 RULE IS CREATED WITH 50 PERCENT CONFIDENCE

```
# Inspect the top 10 association rules

inspect(head(sort(rules, by = "lift"), 10))

inspect(sort(rules, by = "confidence"))
```

```
>   inspect(head(sort(rules, by = "lift"), 10))
     lhs                               rhs                        support confidence   coverage     lift count
[1]  {Eggs, Meat & Fish,
      Kitchen, Garden & Pets}     => {Gourmet & World Food} 0.01214450  0.5895522 0.02059954 1.151317    79
[2]  {Beverages,
      Kitchen, Garden & Pets}     => {Gourmet & World Food} 0.03013067  0.5730994 0.05257494 1.119187   196
[3]  {Beverages,
      Foodgrains, Oil & Masala,
      Kitchen, Garden & Pets}     => {Gourmet & World Food} 0.01029977  0.5677966 0.01813989 1.108831    67
[4]  {Beauty & Hygiene,
      Beverages,
      Kitchen, Garden & Pets}     => {Gourmet & World Food} 0.02044581  0.5659574 0.03612606 1.105240   133
[5]  {Cleaning & Household,
      Fruits & Vegetables,
      Gourmet & World Food}       => {Beauty & Hygiene}     0.01029977  0.7613636 0.01352806 1.084447    67
[6]  {Baby Care,
      Gourmet & World Food,
      Snacks & Branded Foods}     => {Beauty & Hygiene}     0.01322060  0.7610619 0.01737125 1.084018    86
[7]  {Bakery, Cakes & Dairy,
      Beauty & Hygiene,
      Kitchen, Garden & Pets}     => {Gourmet & World Food} 0.01691007  0.5527638 0.03059185 1.079474   110
[8]  {Beauty & Hygiene,
      Beverages}                  => {Gourmet & World Food} 0.04796311  0.5512367 0.08700999 1.076492   312
[9]  {Baby Care,
      Beauty & Hygiene,
      Snacks & Branded Foods}     => {Gourmet & World Food} 0.01322060  0.5477707 0.02413528 1.069723    86
[10] {Beverages,
      Foodgrains, Oil & Masala}   => {Gourmet & World Food} 0.02336664  0.5448029 0.04289008 1.063928   152
```

```
> inspect(sort(rules, by = "confidence"))
     lhs                          rhs                     support confidence   coverage      lift count
[1]  {Cleaning & Household,
      Fruits & Vegetables,
      Gourmet & World Food}    => {Beauty & Hygiene}   0.01029977  0.7613636 0.01352806 1.0844472    67
[2]  {Baby Care,
      Gourmet & World Food,
      Snacks & Branded Foods}  => {Beauty & Hygiene}   0.01322060  0.7610619 0.01737125 1.0840175    86
[3]  {Bakery, Cakes & Dairy,
      Cleaning & Household,
      Kitchen, Garden & Pets}  => {Beauty & Hygiene}   0.01122214  0.7448980 0.01506533 1.0609944    73
[4]  {Cleaning & Household,
      Fruits & Vegetables}     => {Beauty & Hygiene}   0.02136818  0.7393617 0.02890085 1.0531088   139
[5]  {Baby Care,
      Snacks & Branded Foods}  => {Beauty & Hygiene}   0.02413528  0.7302326 0.03305150 1.0401057   157
[6]  {Beverages,
      Snacks & Branded Foods}  => {Beauty & Hygiene}   0.03228286  0.7266436 0.04442736 1.0349938   210
[7]  {Beverages,
      Gourmet & World Food,
      Snacks & Branded Foods}  => {Beauty & Hygiene}   0.01752498  0.7261146 0.02413528 1.0342404   114
[8]  {Bakery, Cakes & Dairy,
      Cleaning & Household,
      Foodgrains, Oil & Masala} => {Beauty & Hygiene}  0.01045350  0.7234043 0.01445042 1.0303798    68
[9]  {Beverages,
      Kitchen, Garden & Pets,
      Snacks & Branded Foods}  => {Beauty & Hygiene}   0.01245196  0.7232143 0.01721752 1.0301092    81
[10] {Fruits & Vegetables,
      Kitchen, Garden & Pets}  => {Beauty & Hygiene}   0.02521138  0.7161572 0.03520369 1.0200575   164
```

1. **If a person buys Eggs, Meat & Fish, and Kitchen, Garden & Pets, there's a 58.96% chance they will also buy Gourmet & World Food.**
2. **If a person buys Beverages and Kitchen, Garden & Pets, there's a 57.31% chance they will also buy Gourmet & World Food.**
3. **If a customer purchases Cleaning & Household, Fruits & Vegetables, and Gourmet & World Food, there's a 76.14% chance they will also buy Beauty & Hygiene.**
4. **If a customer purchases Baby Care, Gourmet & World Food, and Snacks & Branded Foods, there's a 76.11% chance they will also buy Beauty & Hygiene.**

Cluster Analysis:

It's a method for categorizing similar items together in a dataset, organizing data into clusters where items within each cluster are more akin to each other than to those in other clusters.

Example (Customer Segmentation in Retail):

By analysing customer purchase data, like frequency, amount spent, and types of items bought, cluster analysis can group customers into segments with similar buying behaviors.

This helps retailers identify distinct customer segments, such as high-spending frequent buyers or budget-conscious shoppers, allowing them to tailor marketing strategies for each segment and enhance overall effectiveness.

Summary of steps to be done

Data Preparation:
Reads the dataset and selects relevant columns (Sales_price, Market_price, Rating).
Removes rows with missing values.
Standardizes the data.

Hierarchical Clustering:
Computes the dissimilarity matrix using Euclidean distance.
Performs hierarchical clustering using the Ward's method.
Plots the dendrogram.

K-means Clustering:
Performs K-means clustering with 3 clusters and 25 random starts.
Adds cluster numbers to the original dataset.
Visualizes clusters using a scatter plot.

Cluster Analysis:
Prints cluster means (centroid values for each cluster).
Prints cluster sizes (number of data points in each cluster).

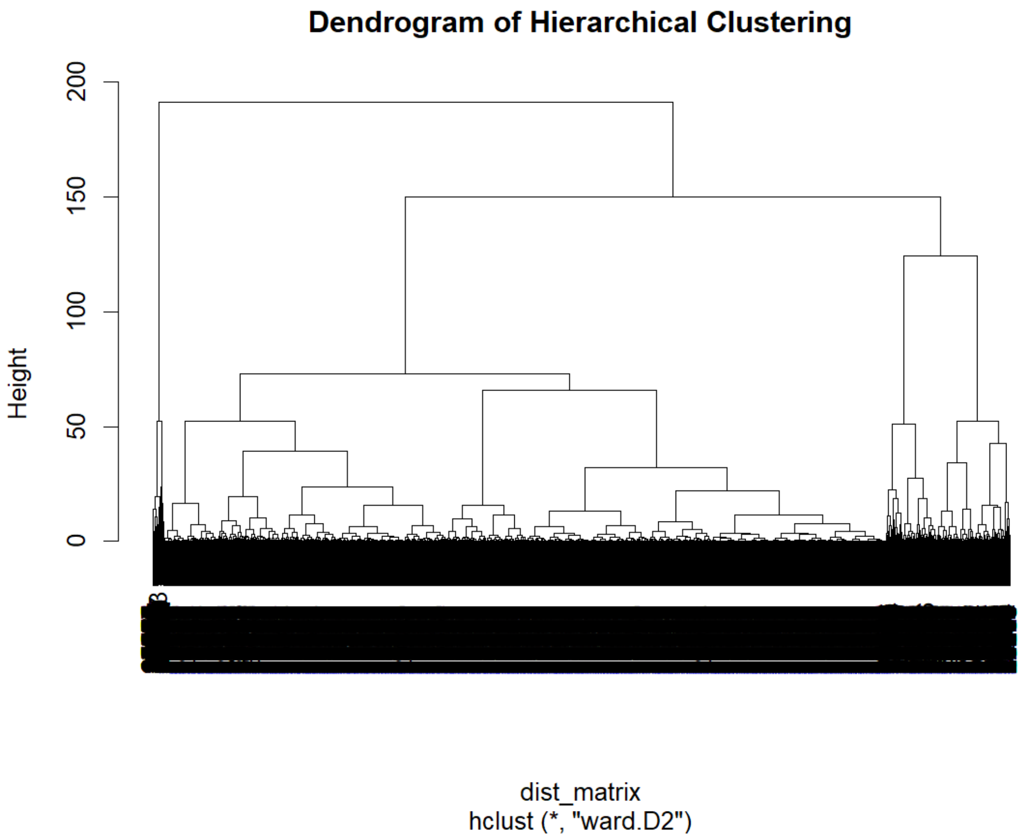
```
library(factoextra)
library(cluster)
library(NbClust)

# Read the data
file_path <- "C:\\Users\\shriv\\Downloads\\r project\\BigBasket Data.csv"
data <- read.csv(file_path)
```

#loading library and dataset

#pre processing the data removing missing values

#making HAC

```
# Compute the distance matrix
dist_matrix <- dist(scaled_data, method = "euclidean")
print(dist_matrix) ##do not print its big data will take lot time

# Hierarchical clustering
hierarchical_clustering <- hclust(dist_matrix, method = "ward.D2")

# Plot the dendrogram
plot(hierarchical_clustering, main = "Dendrogram of Hierarchical Clustering")
```



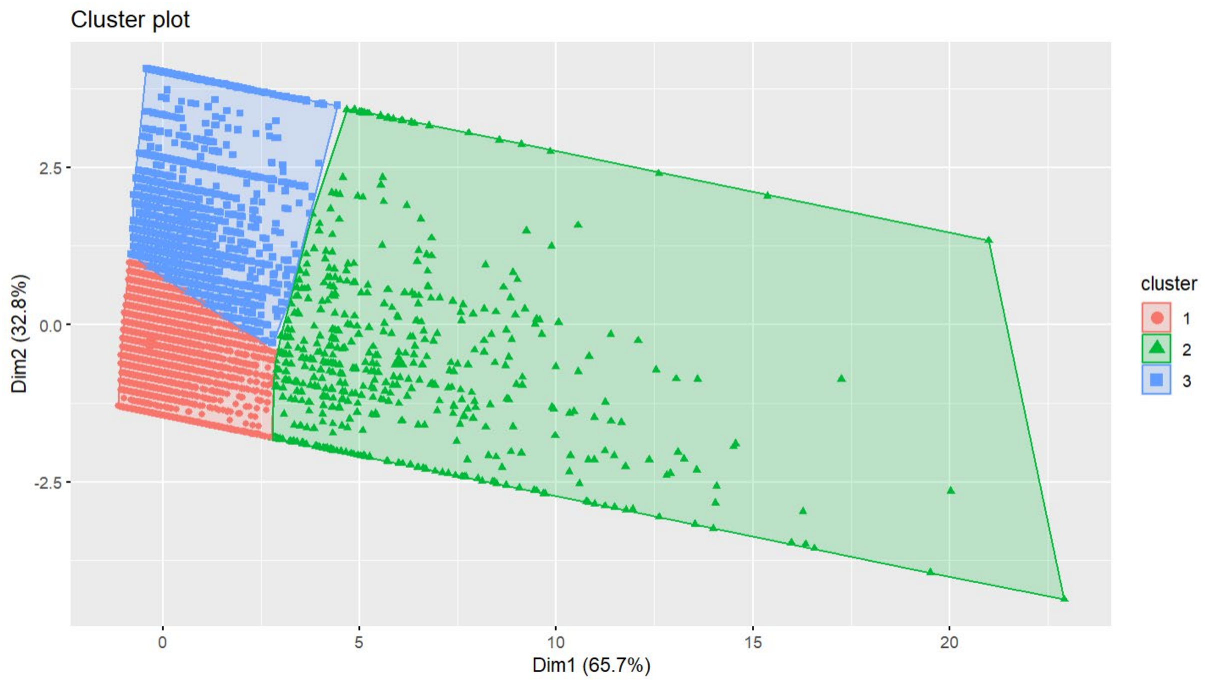
We will choose 3 cluster

```
# K-means clustering
kmeans_results <- kmeans(scaled_data, centers = 3, nstart = 25)

# Add cluster numbers to the original dataset
data$Cluster <- kmeans_results$cluster

# Visualize clusters
fviz_cluster(kmeans_results, data = scaled_data, geom = "point")
```

Print cluster means

aggregate(scaled_data, by = list(Cluster = kmeans_results$cluster), mean)

Print cluster sizes




```
# Print cluster means
aggregate(scaled_data, by = list(Cluster = kmeans_results$cluster), mean)

# Print cluster sizes
kmeans_results$size
```

```
> aggregate(scaled_data, by = list(Cluster = kmeans_results$cluster), mean)
  Cluster Sales_price Market_price     Rating
1       1  -0.1961169   -0.2007379  0.3297602
2       2   4.2240800    4.2973962 -0.1521067
3       3   0.2047700    0.2150214 -1.7831859
> # Print cluster sizes
> kmeans_results$size
[1] 15524   584  2821
```

Cluster 1:

Sales_price: Slightly below average.
Market_price: Slightly below average.
Rating: Positive.
Size: 15,524 records.

Cluster 2:

Sales_price: Relatively high.
Market_price: Relatively high.
Rating: Close to neutral.
Size: 584 records.

Cluster 3:

Sales_price: Slightly above average.
Market_price: Slightly above average.
Rating: Significantly negative.
Size: 2,821 records.

Business Implications:

Cluster 1:

Characteristics: Products in this cluster have slightly below-average prices but receive positive ratings. It contains a large number of records, indicating a significant portion of the product portfolio.
Implications: Businesses can capitalize on the positive customer sentiment by focusing on marketing strategies to increase sales volume within this cluster. Additionally, there may be opportunities to optimize pricing strategies to further enhance competitiveness and profitability.

Cluster 2:

Characteristics: This cluster consists of products with relatively high prices and ratings close to neutral. While the number of records is smaller compared to other clusters, these products may represent premium offerings.
Implications: Businesses should prioritize maintaining the high quality and brand reputation of products in this cluster to justify the premium prices. Marketing efforts can target niche segments willing to pay premium prices for perceived value, ensuring sustainable profitability.

Cluster 3:

Characteristics: Products in this cluster have slightly above-average prices but receive significantly negative ratings. Despite the lower number of records, this cluster represents a notable portion of the product portfolio.
Implications: Immediate attention should be given to addressing the underlying issues affecting customer satisfaction within this cluster. Strategies may include product quality improvements, customer service enhancements, and targeted marketing campaigns to regain customer trust and loyalty. Additionally, optimizing pricing strategies to align with customer expectations can help improve competitiveness and profitability in the long term.

REFERENCES

https://www.rdocumentation.org/packages/arules/versions/1.6-4

https://www.analyticsvidhya.com/blog/2022/03/a-comprehensive-guide-on-ggplot2-in-r/

https://www.geeksforgeeks.org/hierarchical-clustering-in-r-programming/

https://www.geeksforgeeks.org/clustering-in-r-programming/

https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-market-basket-analysis/