***Market Basket Analysis***

Market basket analysis with Apriori algorithm

The retailer wants to target customers with suggestions on itemset that a customer is most likely to purchase .I was given dataset contains data of a retailer; the transaction data provides data around all the transactions that have happened over a period of time. Retailer will use result to grove in his industry and provide for customer suggestions on itemset, we be able increase customer engagement and improve customer experience and identify customer behavior. I will solve this problem with use Association Rules type of unsupervised learning technique that checks for the dependency of one data item on another data item.

### **Introduction**

Association Rule is most used when you are planning to build association in different objects in a set. It works when you are planning to find frequent patterns in a transaction database. It can tell you what items do customers frequently buy together and it allows retailer to identify relationships between the items.

### An Example of Association Rules

Assume there are 100 customers, 10 of them bought Computer Mouth, 9 bought Mat for Mouse and 8 bought both of them.

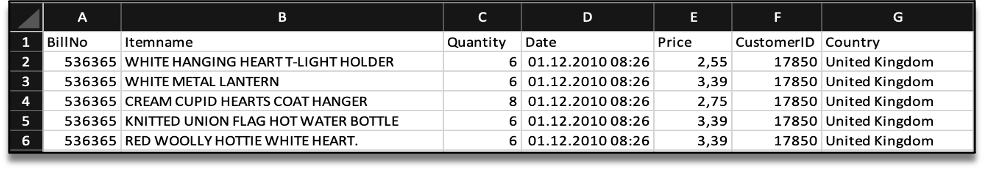
* bought Computer Mouth => bought Mat for Mouse
* support = P(Mouth & Mat) = 8/100 = 0.08
* confidence = support/P(Mat for Mouse) = 0.08/0.09 = 0.89
* lift = confidence/P(Computer Mouth) = 0.89/0.10 = 8.9  
  This just simple example. In practice, a rule needs the support of several hundred transactions, before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

### Strategy

* Data Import
* Data Understanding and Exploration
* Transformation of the data – so that is ready to be consumed by the association rules algorithm
* Running association rules
* Exploring the rules generated
* Filtering the generated rules
* Visualization of Rule

### **Dataset Description**

* File name: Assignment-1\_Data
* List name: retaildata
* File format: . xlsx
* Number of Row: 522065
* Number of Attributes: 7
  + BillNo: 6-digit number assigned to each transaction. Nominal.
  + Itemname: Product name. Nominal.
  + Quantity: The quantities of each product per transaction. Numeric.
  + Date: The day and time when each transaction was generated. Numeric.
  + Price: Product price. Numeric.
  + CustomerID: 5-digit number assigned to each customer. Nominal.
  + Country: Name of the country where each customer resides. Nominal.



### **Libraries in R:**

First, we need to load required libraries. Shortly I describe all libraries.

* arules - Provides the infrastructure for representing,  
  manipulating and analyzing transaction data and patterns (frequent itemsets and association rules).
* arulesViz - Extends package 'arules' with various visualization.  
  techniques for association rules and item-sets. The package also includes several interactive visualizations for rule exploration.
* tidyverse - The tidyverse is an opinionated collection of R packages designed for data science.
* readxl - Read Excel Files in R.
* plyr - Tools for Splitting, Applying and Combining Data.
* ggplot2 - A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

**Importing the libraries:**

**In[1]:**

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

**In[2]:**

From mlxtend.frequent\_patterns import apriori

From mlxtend.frequent\_patterns import association\_rules

### **Data Pre-processing:**

Next, we need to upload Assignment-1\_Data. xlsx to R to read the dataset.Now we can see our data in R.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S no** | **Bill No** | **Item name** | **Quantity** | **Date** | **Price** | **CustomerID** | **Country** |
|  |  |  |  |  |  |  |  |
| 0 | 536365 | WHITE HANGING HEART T-LIGHT HOLDER | 6 | 2010-12-01 08:26:00 | 2.55 | 17850.0 | United Kingdom |
| 1 | 536365 | WHITE METAL LANTERN | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom |
| 2 | 536365 | CREAM CUPID HEARTS COAT HANGER | 8 | 2010-12-01 08:26:00 | 2.75 | 17850.0 | United Kingdom |
| 3 | 536365 | KNITTED UNION FLAG HOT WATER BOTTLE | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom |
| 4 | 536365 | RED WOOLLY HOTTIE WHITE HEART. | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom |
| … | … | … | … | … | … | … | … |
| 195 | 536389 | CHRISTMAS LIGHTS 10 REINDEER | 6 | 2010-12-01 10:03:00 | 8.50 | 12431.0 | Australia |
| 196 | 536389 | VINTAGE UNION JACK CUSHION COVER | 8 | 2010-12-01 10:03:00 | 4.95 | 12431.0 | Australia |
| 197 | 536389 | VINTAGE HEADS AND TAILS CARD GAME | 12 | 2010-12-01 10:03:00 | 1.25 | 12431.0 | Australia |
| 198 | 536389 | SET OF 3 COLOURED FLYING DUCKS | 6 | 2010-12-01 10:03:00 | 5.45 | 12431.0 | Australia |
| 199 | 536389 | SET OF 3 GOLD FLYING DUCKS | 4 | 2010-12-01 10:03:00 | 6.35 |  |  |

After we will clear our data frame, will remove missing values.

**#complete.cases(dota) removing rows with missing values in any column of data frame**

**itenslist itenslist [complete.coses(iterslist),]**

To apply Association Rule mining, we need to convert dataframe into transaction data to make all items that are bought together in one invoice will be in one row. Below lines of code will combine all products from one BillNo and Date and combine all products from that BillNo and Date as one row, with each item, separated by (,)

**#ddply(dataframe, variables\_to\_split\_dataframe, function)**

**transaxtionData ddply(itemslist,c(“BillNo”, “Date”),Function(df1)**

**paste(df1$Itemname,Collapse“,”))**

We don’t need BillNo and Date, we will make it as Null.  
Next, you have to store this transaction data into .csv

**transaxtionData BillNo <-NULL**

**transaxtionData$Date <-NULL**

**#will gave the name to column “item”**

**Colnames(transaxtionData) <- c(“items”)**

This how should look transaction data before we will go to next step.

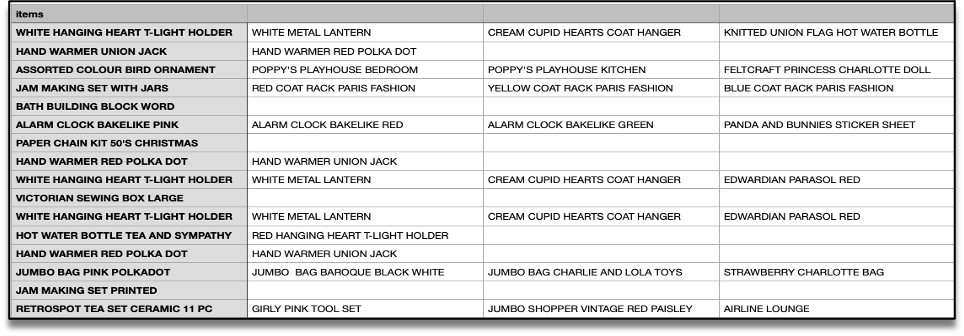
**#quote: If TRUE it will surround character or factor column with double quotes.**

**#1F FALSE nothing will be quoted**

**#row.names: either a logical value indicating whether the ros names of x are to be**

**#written along with x, or a character vector of row names to be written.**

**write.csv(transaxtionDato, “assignent\_itemslist.csv”, quote FALSE, row.names = FALSE)**



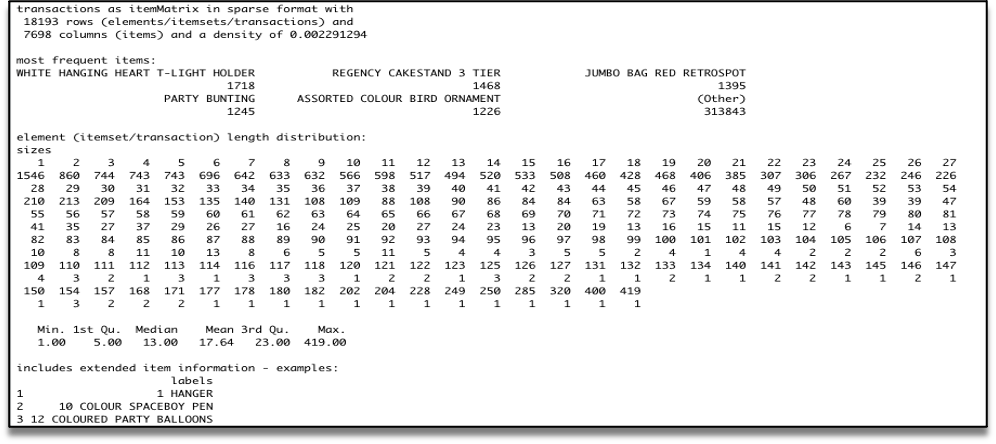
At this step we already have our transaction dataset, and it shows the matrix of items which bought together. We can’t see here any rules and how often it was purchase together. Now let’s check how many transactions we have and what they are. We will have to have to load this transaction data into an object of the transaction class. This is done by using the R function read.transactions of the arules package. Our format of Data frame is basket.

**transactions <-read, transactions('/Users/asik/Desktop/ossigments\_itemslist.csv'. format basket', sep-1,73)**

Let’s have a view our transaction object by summary(transaction)

**Summary (transaction)**

We can see 18193 transactions (rows) and 7698 items (columns). 7698 is the product descriptions and 18193 transactions are collections of these items.



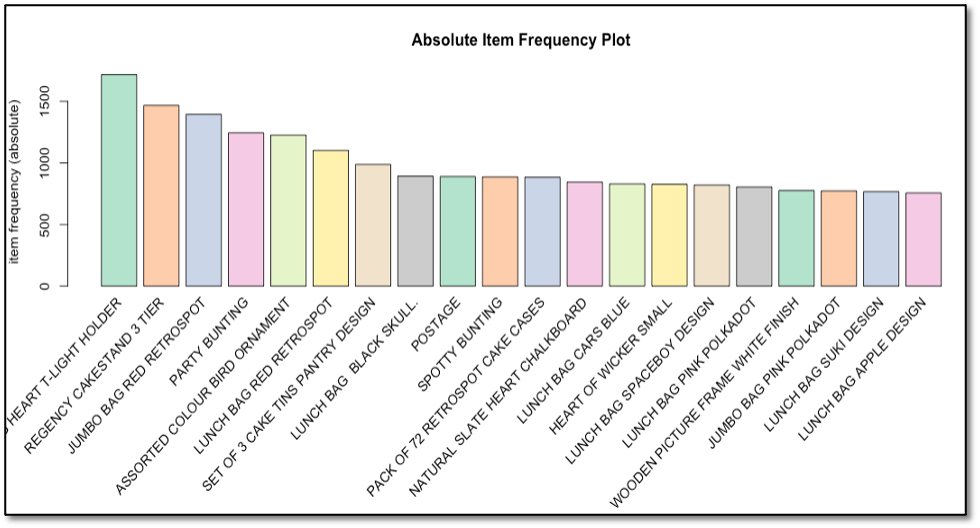
The summary gives us some useful information:

* Density tells the percentage of non-zero cells in a sparse matrix. In other words, total number of items that are purchased divided by a possible number of items in that matrix. You can calculate how many items were purchased by using density: 18193x7698x0.002291294=337445
* Summary will show us most frequent items.
* Element (itemset/transaction) length distribution: It will gave us how many transactions are there for 1-itemset, 2-itemset and so on. The first row is telling you a number of items and the second row is telling you the number of transactions.  
  For example, there is only 1546 transaction for one item, 860 transactions for 2 items, and there are 419 items in one transaction which is the longest.

Let’s check item frequency plot, we will generate an itemFrequencyPlot to create an item Frequency Bar Plot to view the distribution of objects based on itemMatrix (e.g., >transactions or items in >itemsets and >rules) which is our case.

**itemfrequencyFlat transactions, top-type-“”,Pol-brewer pal/8, “Paste”), nain-“bsolute Iten Frequency Pict”)**

**if (Irequire(“RColorBrewer”)) {install.packages(“RColorBrewer”) library(RColorBrewer)**



In itemFrequencyPlot(transaction,topN=20,type="absolute") first argument - our transaction object to be plotted that is tr. topN is allows us to plot top N highest frequency items. type can be as type="absolute" or type="relative". If we will chouse absolute it will plot numeric frequencies of each item independently. If relative it will plot how many times these items have appeared as compared to others. As well I made it in colure for better visualization.

### **Generating Rules:**

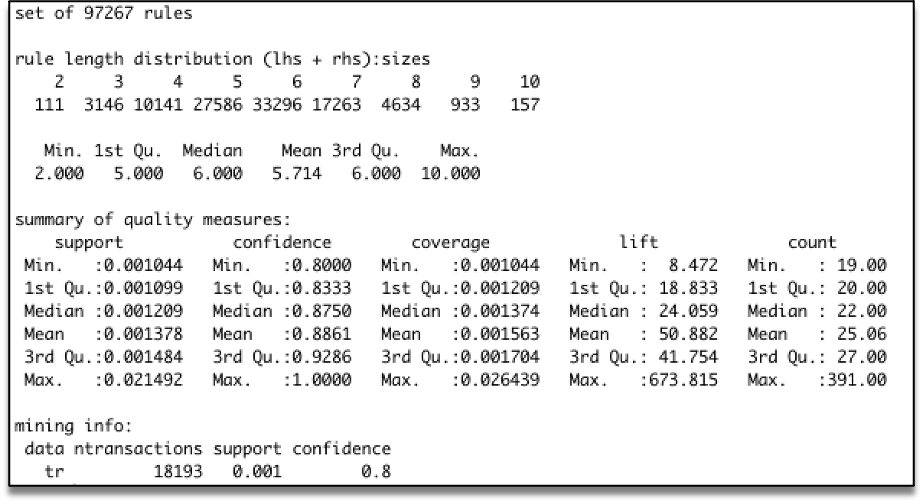
Next, we will generate rules using the Apriori algorithm. The function apriori() is from package arules. The algorithm employs level-wise search for frequent itemsets. Algorithm will generate frequent itemsets and association rules. We pass supp=0.001 and conf=0.8 to return all the rules that have a support of at least 0.1% and confidence of at least 80%. We sort the rules by decreasing confidence and will check summary of the rules.

**generated. Rules apriori(transactions, parameter list(supp-0.001, conf-8.8,maxlen-10))**

**generated. Rules sort generated.rules, by-confidence, decreasing = TRUE)**

**Summary( generated rules)**

The apriori will take (transaction) as the transaction object on which mining is to be applied. parameter will allow you to set min\_sup and min\_confidence. The default values for parameter are minimum support of 0.1, the minimum confidence of 0.8, maximum of 10 items (maxlen).

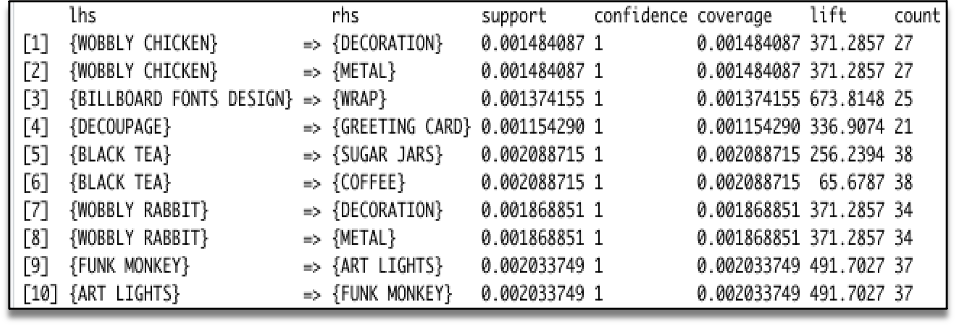


Summary of rules give us clear information as:

* Number of rules: 97267
* The distribution of rules by length: a length of 6 items has the most 33296 and length of 2 items has lowest number of rules 111
* The summary of quality measures: ranges of support, confidence, and lift.
* The information on data mining: total data mined, and the minimum parameters we set earlier

Now, 97267 it a lot of rules. We will identify only top 10.

**Inspect(generated.rules[1:10])**



### **Visualizing Association Rules:**

We have thousands of rules generated based on data, we will need a couple of ways to present our findings. We will use ItemFrequencyPlot to visualize association rules.

#### Scatter-Plot:

**#Filter rules with confidence greater than 0.6 or 60%**

**Rules<-generated.rules[quality(generated. rules) Sconfidence>0.6]**

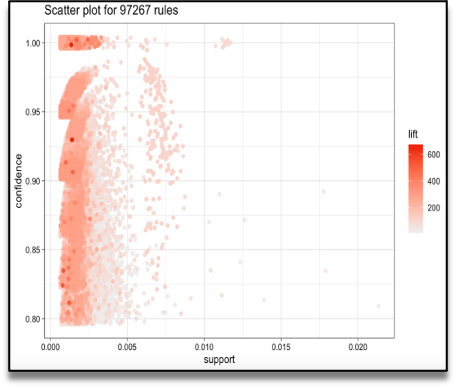
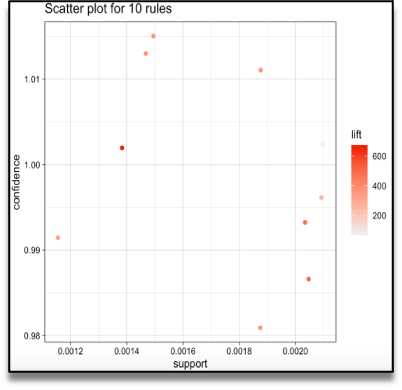
**#Plot Rules**

**plot(Rules)**

**top10Rules - head generated. rules, n = 10, by = "confidence")**

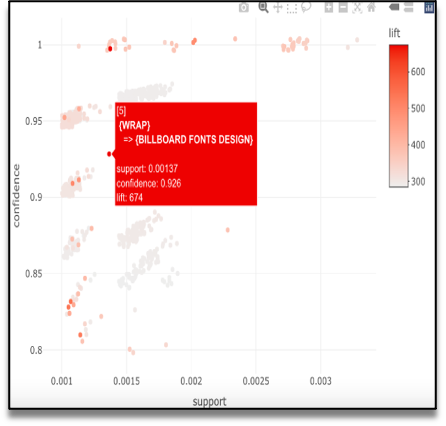
**plot(top10Rules)**

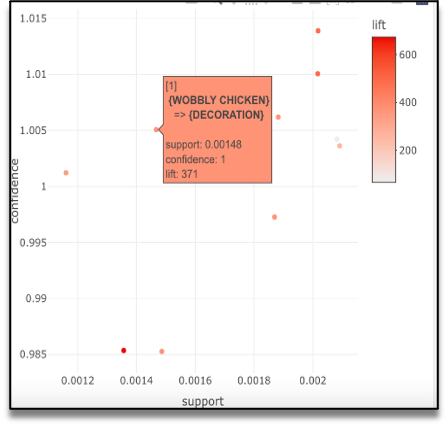
A straight-forward visualization of association rules is to use a scatter plot using plot() of the arulesViz package. It uses Support and Confidence on the axes. In addition, third measure Liftis used by default to color (grey levels) of the points.

#### **Interactive Scatter-Plot:**

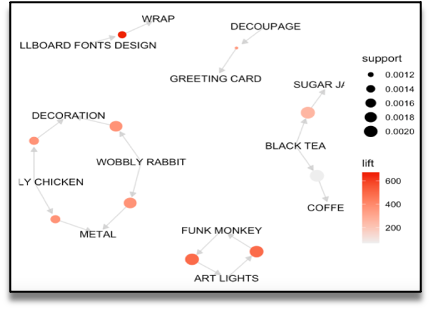
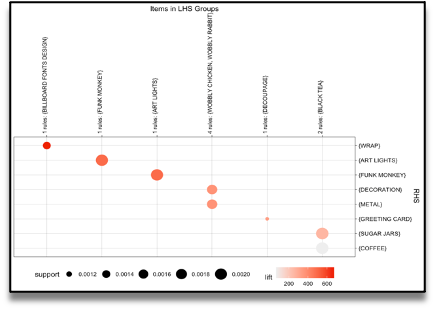
We can have a look for each rule (interactively) and view all quality measures (support, confidence and lift).





#### **Graph - Based Visualization and Group Method:**

Graph plots are a great way to visualize rules but tend to become congested as the number of rules increases. So, it is better to visualize a smaller number of rules with graph-based visualizations. We can see as well group method for top 10 items.

**Conclusion:**

Based on the results of these calculations can be used as a recommendation for retail owners to arrange the arrangement of product catalogs and take strategic steps to improve product marketing.. the retailer can apply effective marketing and sales promotion strategies, he will be able increase customer engagement and improve customer experience and identify customer behavior.