1. **Write a program to Generate Association rules by using Apriori Algorithm**

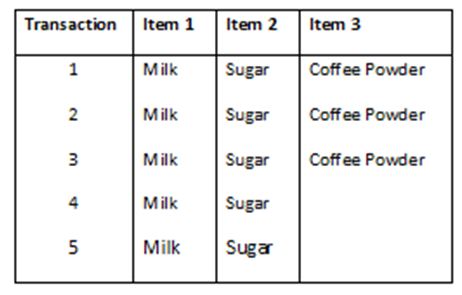
## The Approach(Apriori Algorithm)

When you go to a store, would you not want the aisles to be ordered in such a manner that reduces your efforts to buy things?

For example, I would want the toothbrush, the paste, the mouthwash  & other dental products on a single aisle – because when I buy, I tend to buy them together. This is done by a way in which we find associations between items.

In order to understand the concept better, let’s take a simple dataset (let’s name it as Coffee dataset) consisting of a few hypothetical transactions. We will try to understand this in simple English.

The Coffee dataset consisting of items purchased from a retail store.

Coffee dataset:

**The Association Rules:**

For this dataset, we can write the following association rules: (Rules are just for illustrations and understanding of the concept. They might not represent the actuals).

**Rule 1:** If Milk is purchased, then Sugar is also purchased.

**Rule 2:**  If Sugar is purchased, then Milk is also purchased.

**Rule 3:** If Milk and Sugar are purchased, Then Coffee powder is also purchased in 60% of the transactions.

Generally, association rules are written in “IF-THEN” format. We can also use the term “Antecedent” for IF (LHS) and “Consequent” for THEN (RHS).

From the above rules, we understand the following explicitly:

1. Whenever Milk is purchased, Sugar is also purchased or vice versa.
2. If Milk and Sugar are purchased then the coffee powder is also purchased. This is true in 3 out of the 5 transactions.

### Handling and Readying The Dataset

The first part of any analysis is to bring in the dataset. We will be using an inbuilt dataset “Groceries” from the ‘arules’ package to simplify our analysis.

All stores and retailers store their information of transactions in a specific type of dataset called the “Transaction” type dataset.

*The ‘pacman’ package is an assistor to help load and install the packages. we will be using pacman to load the arules package.*

The p\_load() function from “pacman” takes names of packages as arguments.

If your system has those packages, it will load them and if not, it will install and load them.

### Structural Overview and Prerequisites

Before we begin applying the “Apriori” algorithm on our dataset, we need to make sure that it is of the type “Transactions”.

The structure of our transaction type dataset shows us that it is internally divided into three slots: Data, itemInfo and itemsetInfo.

The slot “Data” contains the dimensions, dimension names and other numerical values of number of products sold by every transaction made.

## Implementing Apriori Algorithm and Key Terms and Usage

rules <- apriori(Groceries,

parameter = list(supp = 0.001, conf = 0.80))

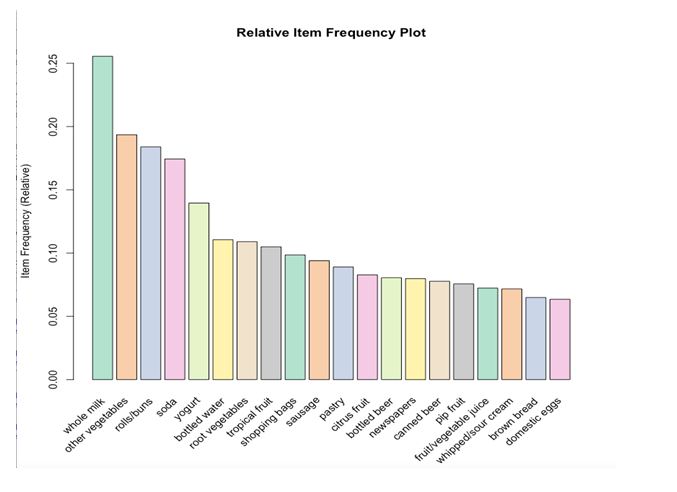
## Interpretations and Analysis

Let us first identify which products were sold how frequently in our dataset.

 The Item Frequency Histogram

These histograms depict how many times an item has occurred in our dataset as compared to the others.

The relative frequency plot accounts for the fact that “Whole Milk” and “Other Vegetables” constitute around half of the transaction dataset; half the sales of the store are of these items.



### Graphical Representation

Moving forward in the visualisation, we can use a graph to highlight the support and lifts of various items in our repository but mostly to see which product is associated with which one in the sales environment.

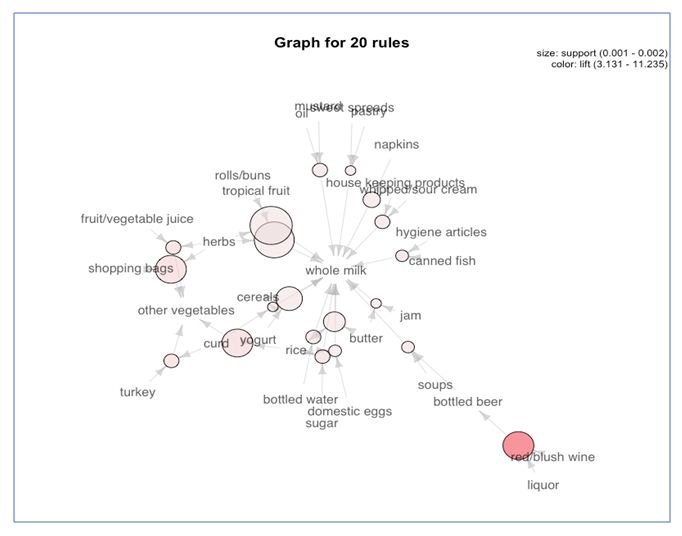
plot(rules[1:20],

method = "graph",

control = list(type = "items"))

This representation gives us a graph model of items in our dataset.

The size of graph nodes is based on support levels and the colour on lift ratios. The incoming lines show the Antecedants or the LHS and the RHS is represented by names of items.



### Individual Rule Representation

The next plot offers us a parallel coordinate system of visualisation. It would help us clearly see that which products along with which ones, result in what kinds of sales.

### Interactive Scatterplot

These plots show us each and every rule visualised into a form of a scatterplot. The confidence levels are plotted on the Y axis and Support levels on the X axis for each rule. We can hover over them in our interactive plot to see the rule