**Context:**

The **Apriori** Algorithm **is** an influential **algorithm** for mining frequent itemsets for boolean association rules. **Apriori uses** a "bottom up" approach, where frequent subsets **are** extended one item at a time (a step known as candidate generation, and groups of candidates **are** tested against the data.)

After Exploratory Analysis of the Data given by Client. We’ve partitioned our data in Three Seasons each containing Four months:

Season 1 (Nov to Feb), Season 2 (Mar to June), Season 3 (July to Oct)

Here, We are applying our Apriori algorithm on these 3 seasons and on the main data as a whole containing all seasons(months) to find out the most selling associated items.

**Author:**

Arijit Pal (aryanarijit@gmail.com),

Abhishek Biswas (abby33459@gmail.com)

**Dataset: Totalcsv.csv, CaseStudy\_Season 1.csv,** **CaseStudy\_Season 2.csv,**  **CaseStudy\_Season 2.csv**

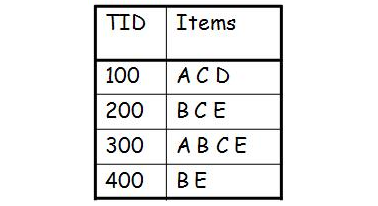
**Read the data**

Read the ‘Groceries\_dataset’ csv file. to the csv file.

**df\_shop <- read.csv(file = "CaseStudy\_Season 1.csv")**

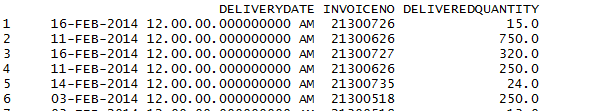
## Data cleaning and manipulations using R

The data required for Apriori must be in the following basket format:



However, the data we have is something like this:

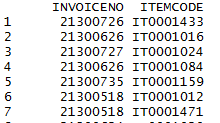
df\_shop



Since the structure of the data is not in the format necessary to find association rules, we have to perform some data manipulations before finding the relationships.

The basket format must have first column as a unique identifier of each transaction, something like a unique receipt number. The second column consists of the items bought in that transaction, separated by spaces or commas or some other separator.

However, the data we now as a subset of main dataframe is something like this:



**Str(df\_shop)**

C:\Users\abby\Desktop\airpassanger\1.PNG

Since the structure of the data is not in the format necessary to find association rules, we have to perform some data manipulations before finding the relationships.

Lets first make sure that the **INVOICENO** are of numeric data type and then sort the dataframe based on the **INVOICENO**.

Lets first make sure that the Member numbers are of numeric data type and then sort the dataframe based on the **INVOICENO**.

**df\_sorted <- df\_shop[order(df\_shop$INVOICENO),]**

**df\_sorted$INVOICENO <- as.numeric(df\_sorted$INVOICENO)**

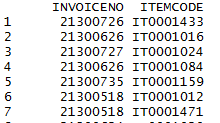
The next step is to actually convert the dataframe into basket format, based on the **INVOICENO** and Date of transaction.

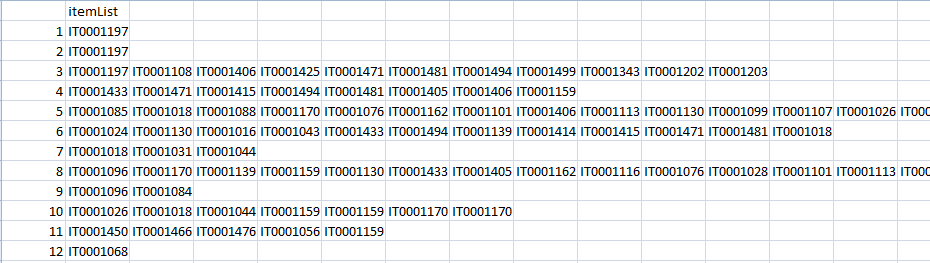
**df\_itemList <- ddply(df\_sorted, c("INVOICENO"), function(df1)paste(df1$ITEMCODE,collapse = ","))**

library(plyr)

The above function ddply() checks the date and **INVOICENO** and pivots the item descriptions with same date and same **INVOICENO** in one line, separated by commas.

Something like this:



becomes:

Once we have the transactions, we no longer need the date and member numbers in our analysis. Go ahead and delete those columns.

**df\_itemList$INVOICENO <- NULL**

**colnames(df\_itemList) <- c("itemList")**

Write the resulting table to a csv file. The reason we do this is, when we write a dataframe to a .csv file, it attaches a row number by default. (unless, of course you were to explicitly tell it not to, by using the argument “row.names=FALSE” in the write.csv function).  
We can simply use these row numbers as transaction IDs, as they would be unique to each transaction.

Write dataframe to a csv file using write.csv()

**write.csv(df\_itemList,"ItemList\_Season1.csv", quote = FALSE, row.names = TRUE)**

**Find the association rules**

Read the csv file u just saved and you will automatically get the transaction IDs in the dataframe  
Run algorithm on ItemList.csv to find relationships among the items. Apriori find these relations based on the frequency of items bought together.

For implementation in R, there is a package called ‘arules’ available that provides functions to read the transactions and find association rules.

So, install and load the package:

**install.packages(“arules”, dependencies=”TRUE”)**

**library(arules)**

convert csv file to basket format

**txn = read.transactions(file="ItemList.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1);**

Quotes are introduced in transactions, which are unnecessary and result in some incorrect results. So, we must get rid of them:

**txn@itemInfo$labels <- gsub("\"","",txn@itemInfo$labels)**

Finally, run the apriori algorithm on the transactions by specifying minimum values for support and confidence.

**basket\_rules <- apriori(txn,parameter = list(minlen=2,sup = 0.001, conf = 0.01, target="rules"))**

Print the association rules. To print the association rules, we use a function called inspect(). However, if you have package ‘tm’ attached in the session, it creates a conflict with the arules package. Thus, we need to check and detach the package.

**if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){**

**detach(package:sentiment, unload=TRUE)**

**detach(package:tm, unload=TRUE)**

**}**

**inspect(basket\_rules)**

Alternative to inspect() is to convert rules to a dataframe and then use Vi

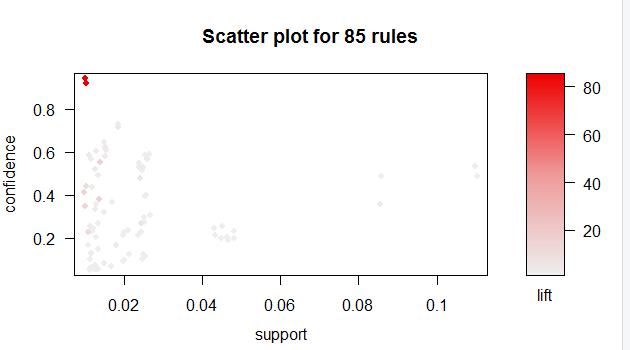
**df\_basket <- as(basket\_rules,"data.frame")**

**View(df\_basket)**

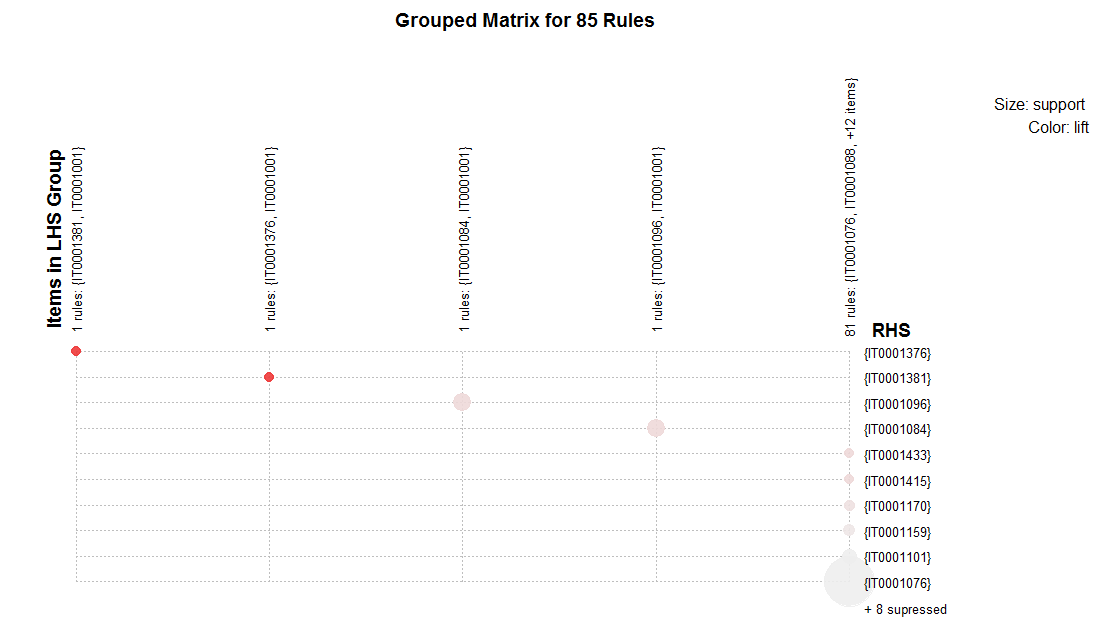
Plot a few graphs that can help you visualize the rules. Install and load the ‘arulesViz’ library for association rules specific visualizations:

**library(arulesViz)**

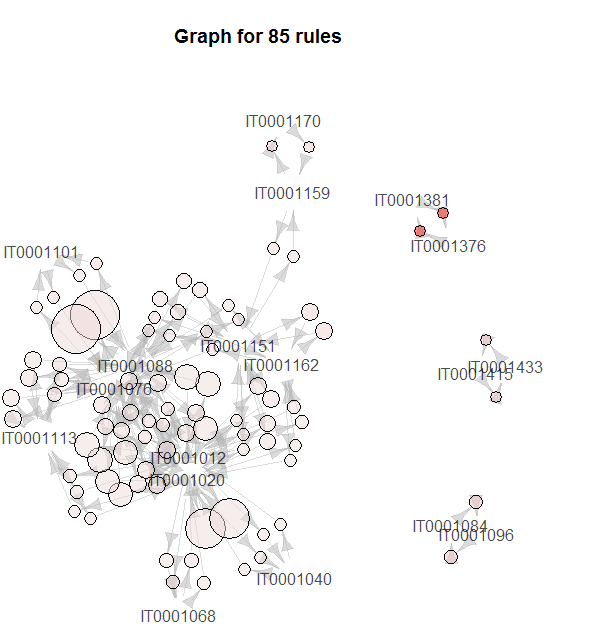
**plot(basket\_rules)**

****

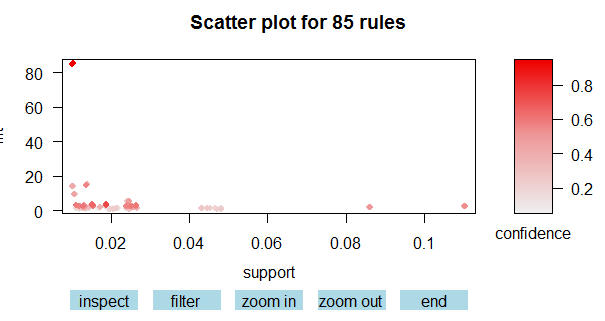
**plot(basket\_rules, method = "grouped", control = list(k = 5))**

****

**plot(basket\_rules, method="graph", control=list(type="items"))**

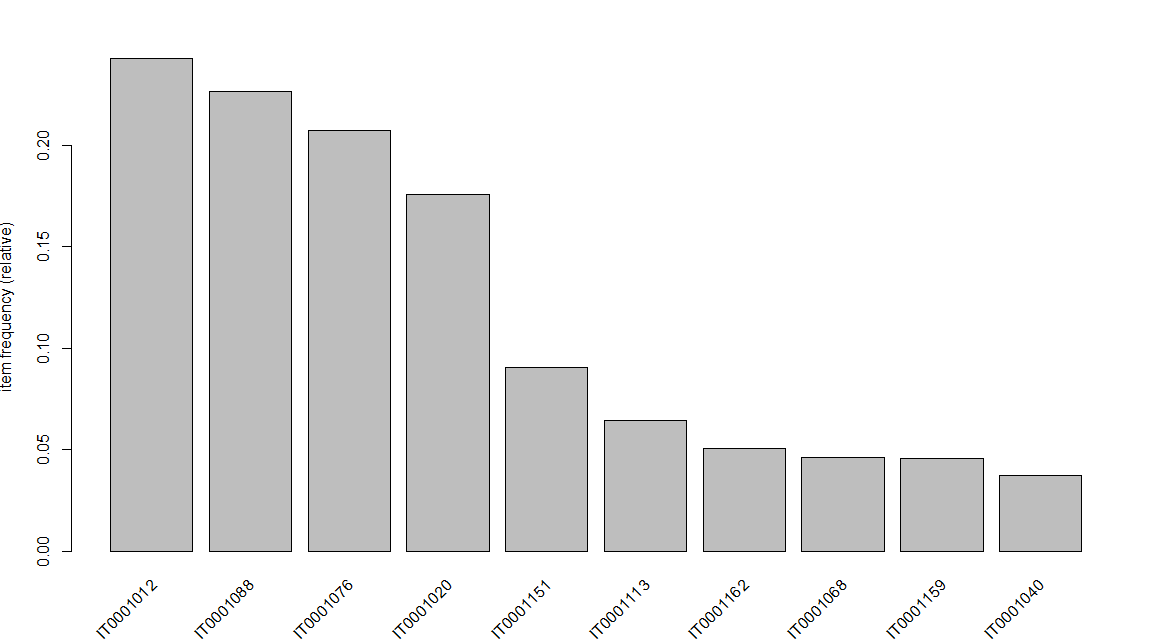
****

**plot(basket\_rules,measure=c("support","lift"),shading="confidence",interactive=T)**

****

Graph to display top 10 items

**itemFrequencyPlot(txn, topN = 10)**

****

**Optional Feature :**

**We have taken the extract of rules named df\_basket\_csv and please check the excel named ‘Recommendation.xls’ where we have tried to group or categories the items based on confident.**

**Finance team needs to give more priority to sell the lowest confidence grouped product to increment the sell in terms of promotion and customer recommendation .**