MARKET BASKET ANALYSIS

Vivek Pruthi,Murali Korrapati, Jyothi Pulimamidi

August 17, 2017

## An attempt to understand the customer preferences:

This project is an attempt to understand the transactional datasetby exploring the dataset further and then try to find out the patterns in the transactions and identify the association Rules. Identifying the association among the parts of transaction can help us understand the effect one part can make on the other part. Once the if x then y scenario is understood, interested party can tweek the purchasing,sales offers, position of item , inventory and even discontinue few categories to help the bottomline of the company.

We will use the "Groceries" dataset . The Groceries data set contains 1 month (30 days) of real-world point-of-sale transaction data from a typical local grocery outlet. The data set contains 9835 transactions and the items are aggregated to 169 items.

Before starting the project we will load the requisite libraries :

library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

library(arulesViz)

## Loading required package: grid

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:arules':  
##   
## intersect, recode, setdiff, setequal, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(plotrix)

let's first check the data provided to get an overview:

data("Groceries")  
class(Groceries)

## [1] "transactions"  
## attr(,"package")  
## [1] "arules"

str(Groceries)

## Formal class 'transactions' [package "arules"] with 3 slots  
## ..@ data :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots  
## .. .. ..@ i : int [1:43367] 13 60 69 78 14 29 98 24 15 29 ...  
## .. .. ..@ p : int [1:9836] 0 4 7 8 12 16 21 22 27 28 ...  
## .. .. ..@ Dim : int [1:2] 169 9835  
## .. .. ..@ Dimnames:List of 2  
## .. .. .. ..$ : NULL  
## .. .. .. ..$ : NULL  
## .. .. ..@ factors : list()  
## ..@ itemInfo :'data.frame': 169 obs. of 3 variables:  
## .. ..$ labels: chr [1:169] "frankfurter" "sausage" "liver loaf" "ham" ...  
## .. ..$ level2: Factor w/ 55 levels "baby food","bags",..: 44 44 44 44 44 44 44 42 42 41 ...  
## .. ..$ level1: Factor w/ 10 levels "canned food",..: 6 6 6 6 6 6 6 6 6 6 ...  
## ..@ itemsetInfo:'data.frame': 0 obs. of 0 variables

As we see that this is transaction dataset . let's try to explore the data within. following is the sparce matrix in the data :

Groceries@data[1:10,1:10]

## 10 x 10 sparse Matrix of class "ngCMatrix"  
##   
## [1,] . . . . . . . . . .  
## [2,] . . . . . . . . . .  
## [3,] . . . . . . . . . .  
## [4,] . . . . . . . . . .  
## [5,] . . . . . . . . . .  
## [6,] . . . . . . . . . .  
## [7,] . . . . . . . . . .  
## [8,] . . . . . . . . . .  
## [9,] . . . . . . . . . .  
## [10,] . . . . . . . . . .

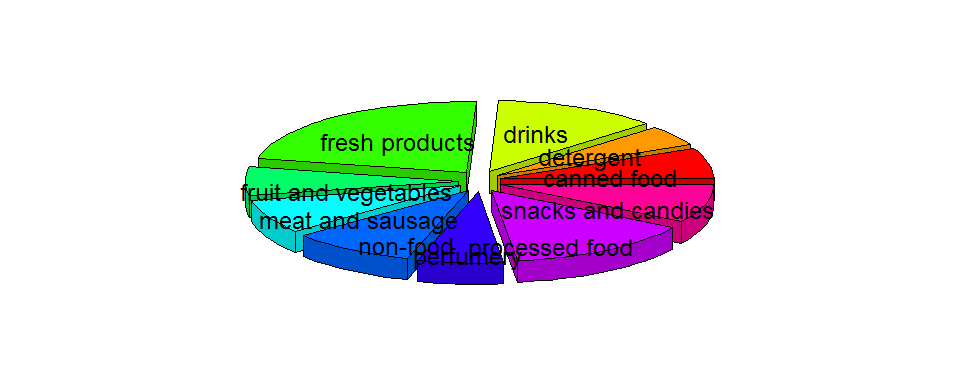
Let's check the iteminfo in the data :

iteminfo<-Groceries@itemInfo  
iteminfo<-as.data.frame(iteminfo)  
iteminfo[1:20,]

## labels level2 level1  
## 1 frankfurter sausage meat and sausage  
## 2 sausage sausage meat and sausage  
## 3 liver loaf sausage meat and sausage  
## 4 ham sausage meat and sausage  
## 5 meat sausage meat and sausage  
## 6 finished products sausage meat and sausage  
## 7 organic sausage sausage meat and sausage  
## 8 chicken poultry meat and sausage  
## 9 turkey poultry meat and sausage  
## 10 pork pork meat and sausage  
## 11 beef beef meat and sausage  
## 12 hamburger meat beef meat and sausage  
## 13 fish fish meat and sausage  
## 14 citrus fruit fruit fruit and vegetables  
## 15 tropical fruit fruit fruit and vegetables  
## 16 pip fruit fruit fruit and vegetables  
## 17 grapes fruit fruit and vegetables  
## 18 berries fruit fruit and vegetables  
## 19 nuts/prunes fruit fruit and vegetables  
## 20 root vegetables vegetables fruit and vegetables

Let's try to get an overview of level1 of the groceries i.e. a top level view , means what are the main categories sold in the store and how many items are in those categories.

category\_details<-iteminfo%>%group\_by(level1)%>%summarize(no\_of\_items=n())  
  
pie3D(x = category\_details$no\_of\_items,labels = category\_details$level1 ,radius = 2,explode=0.2)



we can clearly see that there are 10 categories of products sold in these transactions and most of them are edibles:

levels(category\_details$level1)

## [1] "canned food" "detergent" "drinks"   
## [4] "fresh products" "fruit and vegetables" "meat and sausage"   
## [7] "non-food" "perfumery" "processed food"   
## [10] "snacks and candies"

Let's go into details and check the item labels sold in these transactions:

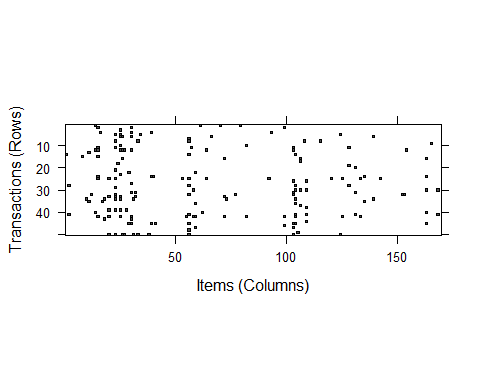
itemLabels(Groceries)

## [1] "frankfurter" "sausage"   
## [3] "liver loaf" "ham"   
## [5] "meat" "finished products"   
## [7] "organic sausage" "chicken"   
## [9] "turkey" "pork"   
## [11] "beef" "hamburger meat"   
## [13] "fish" "citrus fruit"   
## [15] "tropical fruit" "pip fruit"   
## [17] "grapes" "berries"   
## [19] "nuts/prunes" "root vegetables"   
## [21] "onions" "herbs"   
## [23] "other vegetables" "packaged fruit/vegetables"  
## [25] "whole milk" "butter"   
## [27] "curd" "dessert"   
## [29] "butter milk" "yogurt"   
## [31] "whipped/sour cream" "beverages"   
## [33] "UHT-milk" "condensed milk"   
## [35] "cream" "soft cheese"   
## [37] "sliced cheese" "hard cheese"   
## [39] "cream cheese " "processed cheese"   
## [41] "spread cheese" "curd cheese"   
## [43] "specialty cheese" "mayonnaise"   
## [45] "salad dressing" "tidbits"   
## [47] "frozen vegetables" "frozen fruits"   
## [49] "frozen meals" "frozen fish"   
## [51] "frozen chicken" "ice cream"   
## [53] "frozen dessert" "frozen potato products"   
## [55] "domestic eggs" "rolls/buns"   
## [57] "white bread" "brown bread"   
## [59] "pastry" "roll products "   
## [61] "semi-finished bread" "zwieback"   
## [63] "potato products" "flour"   
## [65] "salt" "rice"   
## [67] "pasta" "vinegar"   
## [69] "oil" "margarine"   
## [71] "specialty fat" "sugar"   
## [73] "artif. sweetener" "honey"   
## [75] "mustard" "ketchup"   
## [77] "spices" "soups"   
## [79] "ready soups" "Instant food products"   
## [81] "sauces" "cereals"   
## [83] "organic products" "baking powder"   
## [85] "preservation products" "pudding powder"   
## [87] "canned vegetables" "canned fruit"   
## [89] "pickled vegetables" "specialty vegetables"   
## [91] "jam" "sweet spreads"   
## [93] "meat spreads" "canned fish"   
## [95] "dog food" "cat food"   
## [97] "pet care" "baby food"   
## [99] "coffee" "instant coffee"   
## [101] "tea" "cocoa drinks"   
## [103] "bottled water" "soda"   
## [105] "misc. beverages" "fruit/vegetable juice"   
## [107] "syrup" "bottled beer"   
## [109] "canned beer" "brandy"   
## [111] "whisky" "liquor"   
## [113] "rum" "liqueur"   
## [115] "liquor (appetizer)" "white wine"   
## [117] "red/blush wine" "prosecco"   
## [119] "sparkling wine" "salty snack"   
## [121] "popcorn" "nut snack"   
## [123] "snack products" "long life bakery product"   
## [125] "waffles" "cake bar"   
## [127] "chewing gum" "chocolate"   
## [129] "cooking chocolate" "specialty chocolate"   
## [131] "specialty bar" "chocolate marshmallow"   
## [133] "candy" "seasonal products"   
## [135] "detergent" "softener"   
## [137] "decalcifier" "dish cleaner"   
## [139] "abrasive cleaner" "cleaner"   
## [141] "toilet cleaner" "bathroom cleaner"   
## [143] "hair spray" "dental care"   
## [145] "male cosmetics" "make up remover"   
## [147] "skin care" "female sanitary products"   
## [149] "baby cosmetics" "soap"   
## [151] "rubbing alcohol" "hygiene articles"   
## [153] "napkins" "dishes"   
## [155] "cookware" "kitchen utensil"   
## [157] "cling film/bags" "kitchen towels"   
## [159] "house keeping products" "candles"   
## [161] "light bulbs" "sound storage medium"   
## [163] "newspapers" "photo/film"   
## [165] "pot plants" "flower soil/fertilizer"   
## [167] "flower (seeds)" "shopping bags"   
## [169] "bags"

We can plot the transactions also. For clarity of graph and for proper understanding it is better to plot a subset of dataas:

plot(Groceries[1:50])

## Warning in plot.itemMatrix(Groceries[1:50]): Use image() instead of plot().



There are 169 products which are sold on these transactions. Let's try the first overview over the popularity of a product by checking the frequency of the product. we will find the 10 most popular products and 10 least popular products . Top 10 products by frequency are :

sort(itemFrequency(Groceries,type="absolute"),decreasing = T)[1:10]

## whole milk other vegetables rolls/buns soda   
## 2513 1903 1809 1715   
## yogurt bottled water root vegetables tropical fruit   
## 1372 1087 1072 1032   
## shopping bags sausage   
## 969 924

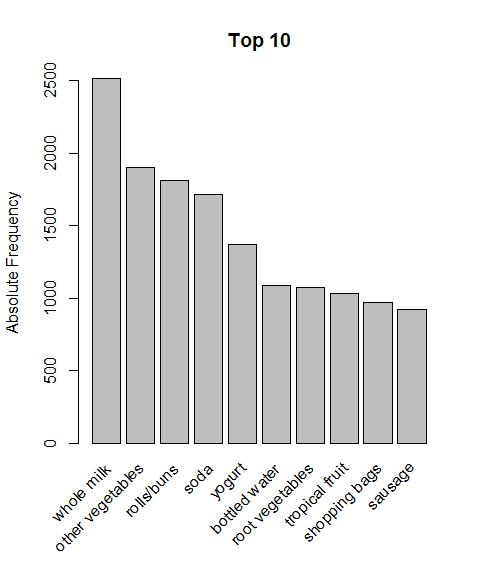
Bottom 10 products by frequency are :

sort(itemFrequency(Groceries,type="absolute"),decreasing = F)[1:10]

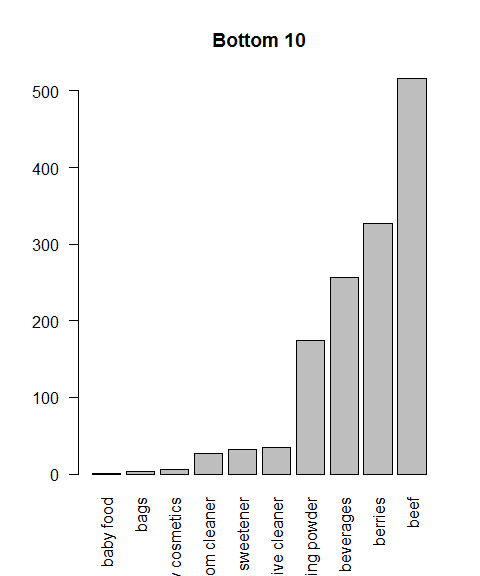
## baby food sound storage medium preservation products   
## 1 1 2   
## kitchen utensil bags frozen chicken   
## 4 4 6   
## baby cosmetics toilet cleaner salad dressing   
## 6 7 8   
## whisky   
## 8

we can plot the frequencies of most famous 10 products next to the least 10 sold products for visual interpratations as:

itemFrequencyPlot(Groceries,topN=10,main="Top 10" , type="absolute",ylab="Absolute Frequency")



barplot(sort(table(unlist(LIST(Groceries)))[1:10],decreasing = F),las=2 , main="Bottom 10")



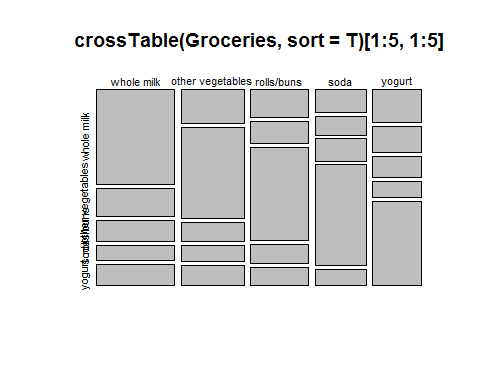
Next obvious step would be to see the association in sales among different pairs of products and try a graphical representation :

crossTable(Groceries,sort=T)[1:5,1:5]

## whole milk other vegetables rolls/buns soda yogurt  
## whole milk 2513 736 557 394 551  
## other vegetables 736 1903 419 322 427  
## rolls/buns 557 419 1809 377 338  
## soda 394 322 377 1715 269  
## yogurt 551 427 338 269 1372

We can graphically check these items with the mosaicplot , but the limitations come to the forefront as the number of items increase in the matrix.

mosaicplot(crossTable(Groceries,sort=T)[1:5,1:5])



we can find the number of transaction in which 2 items were bought together like:

crossTable(Groceries)["soda","yogurt"]

## [1] 269

Not only can we use the count but other measures like "support","confidence","lift","chisquared" can also be found using the crosstable as:

crossTable(Groceries,measure="support")["soda","yogurt"]

## [1] 0.0273513

next obvious step will be to find out a way to create more itemsets that are bought together on the transactions and to find the frequency of those itemsets as well as the other measures like support,confidence,lift and chi. we will use the arules package for this as:

itemset<-apriori(Groceries,parameter = list(support=0.001,minlen=2,target="frequent"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## NA 0.1 1 none FALSE TRUE 5 0.001 2  
## maxlen target ext  
## 10 frequent itemsets FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 9   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].  
## sorting and recoding items ... [157 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 done [0.04s].  
## writing ... [13335 set(s)] done [0.00s].  
## creating S4 object ... done [0.01s].

summary(itemset)

## set of 13335 itemsets  
##   
## most frequent items:  
## whole milk other vegetables yogurt root vegetables   
## 3764 3341 2401 1958   
## tropical fruit (Other)   
## 1796 27683   
##   
## element (itemset/transaction) length distribution:sizes  
## 2 3 4 5 6   
## 2981 6831 3137 376 10   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 3.00 3.00 3.07 4.00 6.00   
##   
## summary of quality measures:  
## support   
## Min. :0.001017   
## 1st Qu.:0.001118   
## Median :0.001423   
## Mean :0.002259   
## 3rd Qu.:0.002237   
## Max. :0.074835   
##   
## includes transaction ID lists: FALSE   
##   
## mining info:  
## data ntransactions support confidence  
## Groceries 9835 0.001 1

to find the top 10 itemsets using support as the main measure :

inspect(sort(itemset,decreasing = T,by="support")[1:10])

## items support   
## [1] {other vegetables,whole milk} 0.07483477  
## [2] {whole milk,rolls/buns} 0.05663447  
## [3] {whole milk,yogurt} 0.05602440  
## [4] {root vegetables,whole milk} 0.04890696  
## [5] {root vegetables,other vegetables} 0.04738180  
## [6] {other vegetables,yogurt} 0.04341637  
## [7] {other vegetables,rolls/buns} 0.04260295  
## [8] {tropical fruit,whole milk} 0.04229792  
## [9] {whole milk,soda} 0.04006101  
## [10] {rolls/buns,soda} 0.03833249

to add another measure to the result e.g check lift :

itemset@quality$lift<-interestMeasure(itemset,measure = "lift",transactions = Groceries)  
inspect(sort(itemset,decreasing = T,by="support")[1:10])

## items support lift   
## [1] {other vegetables,whole milk} 0.07483477 1.5136341  
## [2] {whole milk,rolls/buns} 0.05663447 1.2050318  
## [3] {whole milk,yogurt} 0.05602440 1.5717351  
## [4] {root vegetables,whole milk} 0.04890696 1.7560310  
## [5] {root vegetables,other vegetables} 0.04738180 2.2466049  
## [6] {other vegetables,yogurt} 0.04341637 1.6084566  
## [7] {other vegetables,rolls/buns} 0.04260295 1.1970465  
## [8] {tropical fruit,whole milk} 0.04229792 1.5775950  
## [9] {whole milk,soda} 0.04006101 0.8991124  
## [10] {rolls/buns,soda} 0.03833249 1.1951242

when we need to search for rules insted of the itemsets we change the target to the "rules" in the apriori command as :

rules<-apriori(Groceries,parameter = list(support=.001,confidence=0.9,target="rules"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.9 0.1 1 none FALSE TRUE 5 0.001 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 9   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].  
## sorting and recoding items ... [157 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 done [0.02s].  
## writing ... [129 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(rules)

## set of 129 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 3 4 5 6   
## 10 57 56 6   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.00 4.00 4.00 4.45 5.00 6.00   
##   
## summary of quality measures:  
## support confidence lift   
## Min. :0.001017 Min. :0.9000 Min. : 3.522   
## 1st Qu.:0.001017 1st Qu.:0.9091 1st Qu.: 3.588   
## Median :0.001017 Median :0.9167 Median : 3.669   
## Mean :0.001139 Mean :0.9337 Mean : 4.216   
## 3rd Qu.:0.001220 3rd Qu.:0.9333 3rd Qu.: 4.698   
## Max. :0.001932 Max. :1.0000 Max. :11.235   
##   
## mining info:  
## data ntransactions support confidence  
## Groceries 9835 0.001 0.9

As with the 0.9 confidence we got 129 rules, we can inspect the top 10 after sorting with support as well as confidence:

inspect(sort(rules,decreasing = T,by=c("support","confidence"))[1:10])

## lhs rhs support confidence lift  
## [1] {liquor,   
## red/blush wine} => {bottled beer} 0.001931876 0.9047619 11.235269  
## [2] {tropical fruit,   
## whipped/sour cream,   
## fruit/vegetable juice} => {other vegetables} 0.001931876 0.9047619 4.675950  
## [3] {pip fruit,   
## butter,   
## whipped/sour cream} => {whole milk} 0.001830198 0.9000000 3.522284  
## [4] {tropical fruit,   
## whipped/sour cream,   
## domestic eggs} => {whole milk} 0.001830198 0.9000000 3.522284  
## [5] {root vegetables,   
## whipped/sour cream,   
## flour} => {whole milk} 0.001728521 1.0000000 3.913649  
## [6] {other vegetables,   
## cream cheese ,   
## sugar} => {whole milk} 0.001525165 0.9375000 3.669046  
## [7] {sausage,   
## tropical fruit,   
## root vegetables,   
## yogurt} => {whole milk} 0.001525165 0.9375000 3.669046  
## [8] {root vegetables,   
## other vegetables,   
## yogurt,   
## oil} => {whole milk} 0.001423488 1.0000000 3.913649  
## [9] {citrus fruit,   
## domestic eggs,   
## sugar} => {whole milk} 0.001423488 0.9333333 3.652739  
## [10] {yogurt,   
## domestic eggs,   
## sugar} => {whole milk} 0.001423488 0.9333333 3.652739

as we see that we have rules in the summary insted of the itemsets now with lhs and rhs shown, it is of utmost importance for business purposes to focus on the relevant rules by filtering , so that proper business decisions can be taken.

checking the items on lhs only :

focus\_rules1<-subset(rules,subset=(lhs%in% "yogurt"))  
inspect(sort(focus\_rules1,by="support",decreasing = T)[1:10])

## lhs rhs support confidence lift  
## [1] {sausage,   
## tropical fruit,   
## root vegetables,   
## yogurt} => {whole milk} 0.001525165 0.9375000 3.669046  
## [2] {yogurt,   
## domestic eggs,   
## sugar} => {whole milk} 0.001423488 0.9333333 3.652739  
## [3] {root vegetables,   
## other vegetables,   
## yogurt,   
## oil} => {whole milk} 0.001423488 1.0000000 3.913649  
## [4] {root vegetables,   
## whole milk,   
## yogurt,   
## oil} => {other vegetables} 0.001423488 0.9333333 4.823612  
## [5] {citrus fruit,   
## tropical fruit,   
## root vegetables,   
## whole milk,   
## yogurt} => {other vegetables} 0.001423488 0.9333333 4.823612  
## [6] {root vegetables,   
## other vegetables,   
## yogurt,   
## rice} => {whole milk} 0.001321810 0.9285714 3.634103  
## [7] {root vegetables,   
## whole milk,   
## yogurt,   
## rice} => {other vegetables} 0.001321810 0.9285714 4.799002  
## [8] {beef,   
## tropical fruit,   
## yogurt,   
## rolls/buns} => {whole milk} 0.001321810 0.9285714 3.634103  
## [9] {root vegetables,   
## other vegetables,   
## yogurt,   
## hard cheese} => {whole milk} 0.001220132 0.9230769 3.612599  
## [10] {frankfurter,   
## tropical fruit,   
## root vegetables,   
## yogurt} => {whole milk} 0.001220132 0.9230769 3.612599

above inspection show us the top 10 rules by support showing the itemsets with "yogurt" in it that will result in the purchase of other items i.e.a basket full of other products with yougurt being one of them will result in purchase of what new item.This may not be that important but it will become interesting furter. Let us find the rules with whole milk and yogurt on lhs result in purchase of what on rhs.

focus\_rules2<-subset(rules,subset=(lhs%ain%c("yogurt","whole milk")))  
  
inspect(sort(focus\_rules2,decreasing = T,by="support")[1:5])

## lhs rhs support confidence lift  
## [1] {root vegetables,   
## whole milk,   
## yogurt,   
## oil} => {other vegetables} 0.001423488 0.9333333 4.823612  
## [2] {citrus fruit,   
## tropical fruit,   
## root vegetables,   
## whole milk,   
## yogurt} => {other vegetables} 0.001423488 0.9333333 4.823612  
## [3] {root vegetables,   
## whole milk,   
## yogurt,   
## rice} => {other vegetables} 0.001321810 0.9285714 4.799002  
## [4] {tropical fruit,   
## grapes,   
## whole milk,   
## yogurt} => {other vegetables} 0.001016777 1.0000000 5.168156  
## [5] {tropical fruit,   
## root vegetables,   
## whole milk,   
## yogurt,   
## oil} => {other vegetables} 0.001016777 0.9090909 4.698323

In fact this comes with " yogurt" and " whole milk" with other products in the basket but if we want to focys only on the person buying only these two on lhs and are interested in checking the rhs for him/her , we will have to tweak the rules as :

rules\_tweek<-apriori(Groceries,parameter = list(support=0.01,confidence=0.9,target="rules"),appearance = list(lhs = c("whole milk","yogurt"), default="rhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.9 0.1 1 none FALSE TRUE 5 0.01 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 98   
##   
## set item appearances ...[2 item(s)] done [0.00s].  
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].  
## sorting and recoding items ... [88 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 done [0.00s].  
## writing ... [0 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(rules\_tweek)

## set of 0 rules

It comes with 0 rules, as there maynot be datapoints supporting this but we can check with basket containing {citrus fruit,butter,curd} on lhs for an example and check rhs:

rules\_tweek2<-apriori(Groceries,parameter = list(support=0.001,confidence=0.5,target="rules",minlen=4,maxlen=4),appearance = list(lhs = c("citrus fruit","butter","curd"), default="rhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.5 0.1 1 none FALSE TRUE 5 0.001 4  
## maxlen target ext  
## 4 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 9   
##   
## set item appearances ...[3 item(s)] done [0.00s].  
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].  
## sorting and recoding items ... [157 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4

## Warning in apriori(Groceries, parameter = list(support = 0.001, confidence  
## = 0.5, : Mining stopped (maxlen reached). Only patterns up to a length of 4  
## returned!

## done [0.02s].  
## writing ... [1 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(rules\_tweek2)

## set of 1 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 4   
## 1   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4 4 4 4 4 4   
##   
## summary of quality measures:  
## support confidence lift   
## Min. :0.001118 Min. :0.9167 Min. :3.588   
## 1st Qu.:0.001118 1st Qu.:0.9167 1st Qu.:3.588   
## Median :0.001118 Median :0.9167 Median :3.588   
## Mean :0.001118 Mean :0.9167 Mean :3.588   
## 3rd Qu.:0.001118 3rd Qu.:0.9167 3rd Qu.:3.588   
## Max. :0.001118 Max. :0.9167 Max. :3.588   
##   
## mining info:  
## data ntransactions support confidence  
## Groceries 9835 0.001 0.5

now we have 1 rule and we can inspect those as :

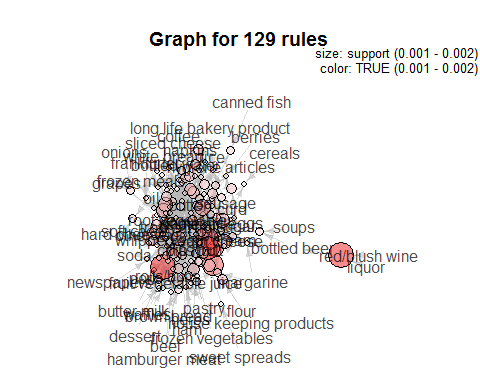
inspect(rules\_tweek2)

## lhs rhs support confidence  
## [1] {citrus fruit,butter,curd} => {whole milk} 0.001118454 0.9166667   
## lift   
## [1] 3.587512

Clearly it tells us that these 3 products together can lead to buying the 4th product " whole milk" .we have that happening on 9 out of 10 transactions like that . although the %age of such transactions in the total transactions is low. as the lift is high , putting these 3 products together may induce the customer to buy the 4th .

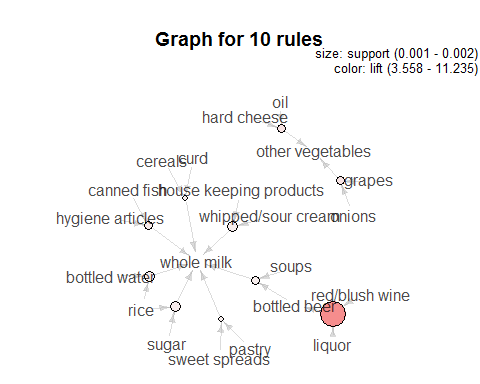
Now we come to the stage of graphically showing the rules in the form of a network diagram where inputs(lhs) are incoming arrows to the circle and the outputs(rhs) are outgoing arrows. It may not be clear from whole rules graph ,but we will filter down to understand:

plot(rules,method = "graph",shading = T)



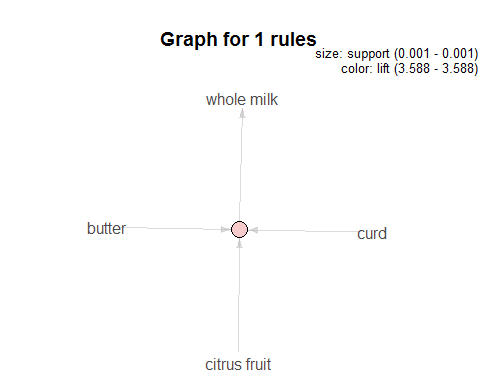
using the same on a subset of rules:

plot(rules[1:10],method = "graph",shading = "lift")



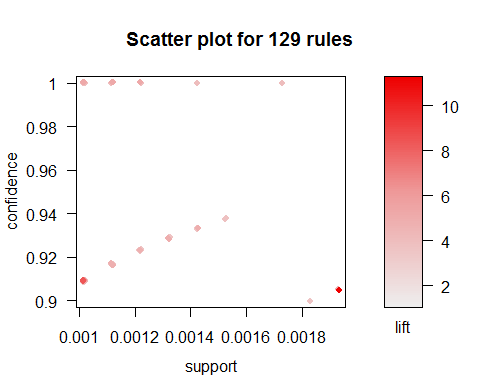
As we can see in the subset of rules that {liquor,red/blush wine} => {bottled beer} rule in the graph shows big red circle in the centre . size of the circle shows the support and shading shows higher lift. This means this rule needs special attention from the business point of view. To show the directionality of arrows we will graph the single rule as:

plot(rules\_tweek2,method = "graph",shading = "lift")



Another way to plot the rules is the following . This is useful , if you want to drill down the rules based on the graph and then analyse further. For this you need to use the interactive part of the plot.

plot(rules)



Conclusion:

Although from the analysis of Market basket analysis , it seems like a tool to help retailers ( online or brick) to help them in cross selling, Product placement,Affinity Promotion, fraud detection or customer behavior, but in fact it is an indespensible tool for a data scientist as it is basically and association analysis. By identifying itemsets or set of inputs, we can after analysis safely forecast the outcome with probability of happening mentioned alongside.