Market Basket Analysis

setwd("D:/Great Lakes/Projects/Domain/Marketing")  
getwd()

## [1] "D:/Great Lakes/Projects/Domain/Marketing"

MBA <- read.table("Cafe Coffee Night-1.csv",sep=",",header = TRUE)  
MBA <- MBA[,-11]

MBA.Agg <- split(MBA$Item.Desc,MBA$Bill.Number)  
head(MBA.Agg)

## $G0470109  
## [1] COSTARICA TARRAZU (REG)   
## 580 Levels: 1 AXE TWIST ...  
##   
## $G0470110  
## [1] SILVER APPLE SINGLE   
## 580 Levels: 1 AXE TWIST ...  
##   
## $G0470111  
## [1] CHAI LATTE PINK LEMONADE   
## [3] CRUMBED TOFU CRUSTINI ROMA TOMATO & JALAPENO CROQUE   
## 580 Levels: 1 AXE TWIST ...  
##   
## $G0470112  
## [1] QUA MINERAL WATER(1000ML) NIRVANA HOOKAH SINGLE   
## 580 Levels: 1 AXE TWIST ...  
##   
## $G0470113  
## [1] NIRVANA HOOKAH SINGLE   
## 580 Levels: 1 AXE TWIST ...  
##   
## $G0470114  
## [1] QUA MINERAL WATER(1000ML) CAPPUCCINO   
## [3] ITALIAN OMELETTE BREAKFAST   
## 580 Levels: 1 AXE TWIST ...

MBA.Agg1 <- list()  
for(i in 1: length(MBA.Agg)){  
 MBA.Agg1[[i]] = unique(MBA.Agg[[i]])  
}  
head(MBA.Agg1)

## [[1]]  
## [1] COSTARICA TARRAZU (REG)   
## 580 Levels: 1 AXE TWIST ...  
##   
## [[2]]  
## [1] SILVER APPLE SINGLE   
## 580 Levels: 1 AXE TWIST ...  
##   
## [[3]]  
## [1] CHAI LATTE PINK LEMONADE   
## [3] CRUMBED TOFU CRUSTINI ROMA TOMATO & JALAPENO CROQUE   
## 580 Levels: 1 AXE TWIST ...  
##   
## [[4]]  
## [1] QUA MINERAL WATER(1000ML) NIRVANA HOOKAH SINGLE   
## 580 Levels: 1 AXE TWIST ...  
##   
## [[5]]  
## [1] NIRVANA HOOKAH SINGLE   
## 580 Levels: 1 AXE TWIST ...  
##   
## [[6]]  
## [1] QUA MINERAL WATER(1000ML) CAPPUCCINO   
## [3] ITALIAN OMELETTE BREAKFAST   
## 580 Levels: 1 AXE TWIST ...

library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'arules'

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

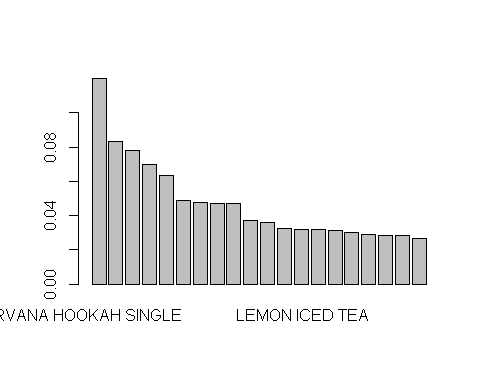
Txns = as(MBA.Agg1,"transactions")  
summary(Txns)

## transactions as itemMatrix in sparse format with  
## 69982 rows (elements/itemsets/transactions) and  
## 580 columns (items) and a density of 0.003573628   
##   
## most frequent items:  
## NIRVANA HOOKAH SINGLE MINT FLAVOUR SINGLE   
## 8389 5812   
## CAPPUCCINO GREAT LAKES SHAKE   
## 5468 4871   
## SAMBUCA (Other)   
## 4423 116089   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 1 2 3 4 5 6 7 8 9 10 11 12   
## 31890 19118 9720 4839 2284 1101 511 255 115 66 27 22   
## 13 14 15 16 17 19 21 23   
## 12 9 4 2 3 1 1 2   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 2.073 3.000 23.000   
##   
## includes extended item information - examples:  
## labels  
## 1 1 AXE TWIST   
## 2 1+1 BTL4 SEASON WHITE   
## 3 1+1 GLS 4SEASON RED

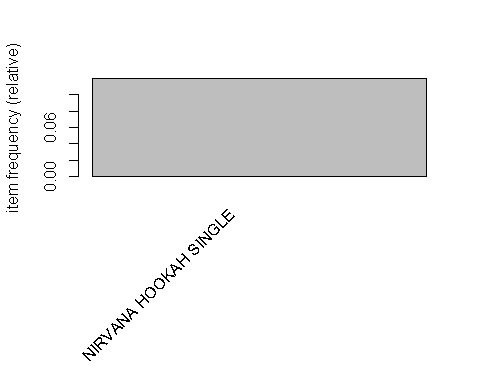
inspect(Txns[10])

## items   
## [1] {MASALA CHAI CUTTING ,  
## MOROCCAN MINT TEA }

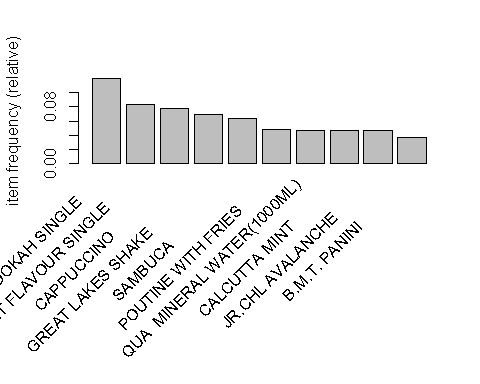
freq <- itemFrequency(Txns)  
freq <- freq[order(-freq)]  
barplot(freq[1:20])



itemFrequencyPlot(Txns,support=0.1)



itemFrequencyPlot(Txns,topN = 10)



#freq <- itemFrequency(Txns)  
#freq <- freq[order(-freq)]  
#barplot(freq[1:20])  
#itemFrequencyPlot(Txns,support=0.1)  
#itemFrequencyPlot(Txns,topN = 10)

#arules <- apriori(data = Txns)  
arules <- apriori(data = Txns,parameter = list(support=0.0005,confidence=0.2,maxlen=2))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.2 0.1 1 none FALSE TRUE 5 5e-04 1  
## maxlen target ext  
## 2 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 34   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[580 item(s), 69982 transaction(s)] done [0.02s].  
## sorting and recoding items ... [295 item(s)] done [0.00s].  
## creating transaction tree ... done [0.03s].  
## checking subsets of size 1 2

## Warning in apriori(data = Txns, parameter = list(support = 5e-04,  
## confidence = 0.2, : Mining stopped (maxlen reached). Only patterns up to a  
## length of 2 returned!

## done [0.01s].  
## writing ... [14 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

inspect(sort(arules,by="lift"))

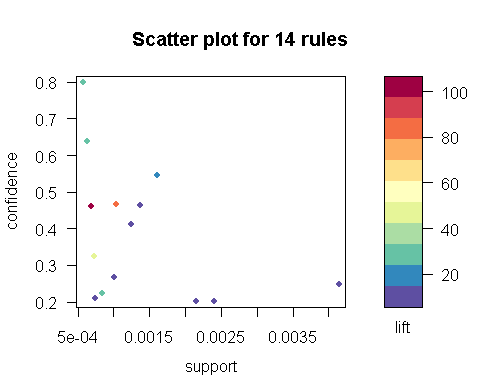
## lhs rhs support confidence lift count  
## [1] {BUTTERED TOASTS } => {KHEEMA GHOTALA } 0.0006858907 0.4615385 101.890803 48  
## [2] {ADD HERB ROAST CHICKEN } => {LEMON INFUSED CHAR GRILLED VEG} 0.0010431254 0.4649682 79.364393 73  
## [3] {ADD HERB ROAST CHICKEN } => {ORANGE ARRABIATA } 0.0007287588 0.3248408 45.925265 51  
## [4] {ADD VANILLA FLAVOUR } => {CAFFE LATTE } 0.0005715755 0.8000000 26.723437 40  
## [5] {ADD CARAMEL FLAVOUR } => {CAFFE LATTE } 0.0006287331 0.6376812 21.301290 44  
## [6] {TRADITIONAL ITALIAN CRUSTINI } => {ADD FRIES } 0.0008430739 0.2234848 20.797762 59  
## [7] {ADD HAZELNUT FLAVOUR } => {CAFFE LATTE } 0.0016147009 0.5458937 18.235195 113  
## [8] {FRENCH FRIES } => {B.M.T. PANINI } 0.0010145466 0.2659176 7.143741 71  
## [9] {ADD HAZELNUT FLAVOUR } => {CAPPUCCINO } 0.0013717813 0.4637681 5.935519 96  
## [10] {VANILLA ICECREAM } => {GREAT LAKES SHAKE } 0.0012431768 0.4123223 5.923863 87  
## [11] {ADD FRIES } => {B.M.T. PANINI } 0.0021576977 0.2007979 5.394333 151  
## [12] {RED BULL 2+1 } => {SAMBUCA } 0.0041439227 0.2485004 3.931846 290  
## [13] {3 RED BULL } => {NIRVANA HOOKAH SINGLE } 0.0007430482 0.2096774 1.749153 52  
## [14] {QUA MINERAL WATER(500ML) } => {NIRVANA HOOKAH SINGLE } 0.0024006173 0.2016807 1.682443 168

library(arulesViz)

## Loading required package: grid

## Registered S3 method overwritten by 'seriation':  
## method from   
## reorder.hclust gclus

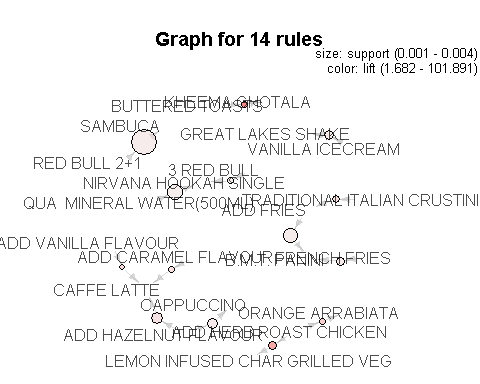
library(RColorBrewer)  
plot(arules,control = list(col = brewer.pal(11,"Spectral")))



Subrules <- head(sort(arules,by="support"),20)  
inspect(Subrules)

## lhs rhs support confidence lift count  
## [1] {RED BULL 2+1 } => {SAMBUCA } 0.0041439227 0.2485004 3.931846 290  
## [2] {QUA MINERAL WATER(500ML) } => {NIRVANA HOOKAH SINGLE } 0.0024006173 0.2016807 1.682443 168  
## [3] {ADD FRIES } => {B.M.T. PANINI } 0.0021576977 0.2007979 5.394333 151  
## [4] {ADD HAZELNUT FLAVOUR } => {CAFFE LATTE } 0.0016147009 0.5458937 18.235195 113  
## [5] {ADD HAZELNUT FLAVOUR } => {CAPPUCCINO } 0.0013717813 0.4637681 5.935519 96  
## [6] {VANILLA ICECREAM } => {GREAT LAKES SHAKE } 0.0012431768 0.4123223 5.923863 87  
## [7] {ADD HERB ROAST CHICKEN } => {LEMON INFUSED CHAR GRILLED VEG} 0.0010431254 0.4649682 79.364393 73  
## [8] {FRENCH FRIES } => {B.M.T. PANINI } 0.0010145466 0.2659176 7.143741 71  
## [9] {TRADITIONAL ITALIAN CRUSTINI } => {ADD FRIES } 0.0008430739 0.2234848 20.797762 59  
## [10] {3 RED BULL } => {NIRVANA HOOKAH SINGLE } 0.0007430482 0.2096774 1.749153 52  
## [11] {ADD HERB ROAST CHICKEN } => {ORANGE ARRABIATA } 0.0007287588 0.3248408 45.925265 51  
## [12] {BUTTERED TOASTS } => {KHEEMA GHOTALA } 0.0006858907 0.4615385 101.890803 48  
## [13] {ADD CARAMEL FLAVOUR } => {CAFFE LATTE } 0.0006287331 0.6376812 21.301290 44  
## [14] {ADD VANILLA FLAVOUR } => {CAFFE LATTE } 0.0005715755 0.8000000 26.723437 40

plot(Subrules,method = "graph")



rules\_df <- as(arules,"data.frame")  
#Rule {a} => {b}  
#Prob(A), LHS Support  
rules\_df$LHS\_Support <- rules\_df$support/rules\_df$confidence  
#Prob(B), RHS Support  
rules\_df$RHS\_Support <- rules\_df$confidence /rules\_df$lift  
#inspect(rules\_df)  
#write.table(rules\_df,file="mba\_output.csv",sep=",",append = FALSE,row.names = FALSE)