################################# ANN #####################################

**#Objective: Prepare rules for the all the data sets**

**#1) Try different values of support and confidence. Observe the change in**

**#number of rules for different support,confidence values**

**#2) Change the minimum length in apriori algorithm**

**#3) Visulize the obtained rules using different plots**

**#Data : groceries.csv**

##################################################################################

install.packages("arules")

library("arules") # Used for building association rules i.e. apriori algorithm

**##Step1 : Data Exploration**

groceries<-read.transactions('D:\\Shilpa\\Datascience\\Assignments\\Association Rule\\groceries.csv',format="basket")

# showing only top 10 transactions

inspect(groceries[1:10])

#items

#[1] {bread,margarine,ready,citrus,fruit,semi-finished,soups}

#[2] {fruit,yogurt,coffee,tropical}

#[3] {milk,whole}

#[4] {,meat,cheese,fruit,yogurt,cream,pip,spreads}

#[5] {bakery,life,milk,condensed,milk,long,other,product,vegetables,whole}

#[6] {cleaner,milk,butter,yogurt,rice,abrasive,whole}

#[7] {rolls/buns}

#[8] {(appetizer),beer,liquor,other,vegetables,UHT-milk,rolls/buns,bottled}

#[9] {plants,pot}

#[10] {milk,cereals,whole}

#Examine the frequency of items

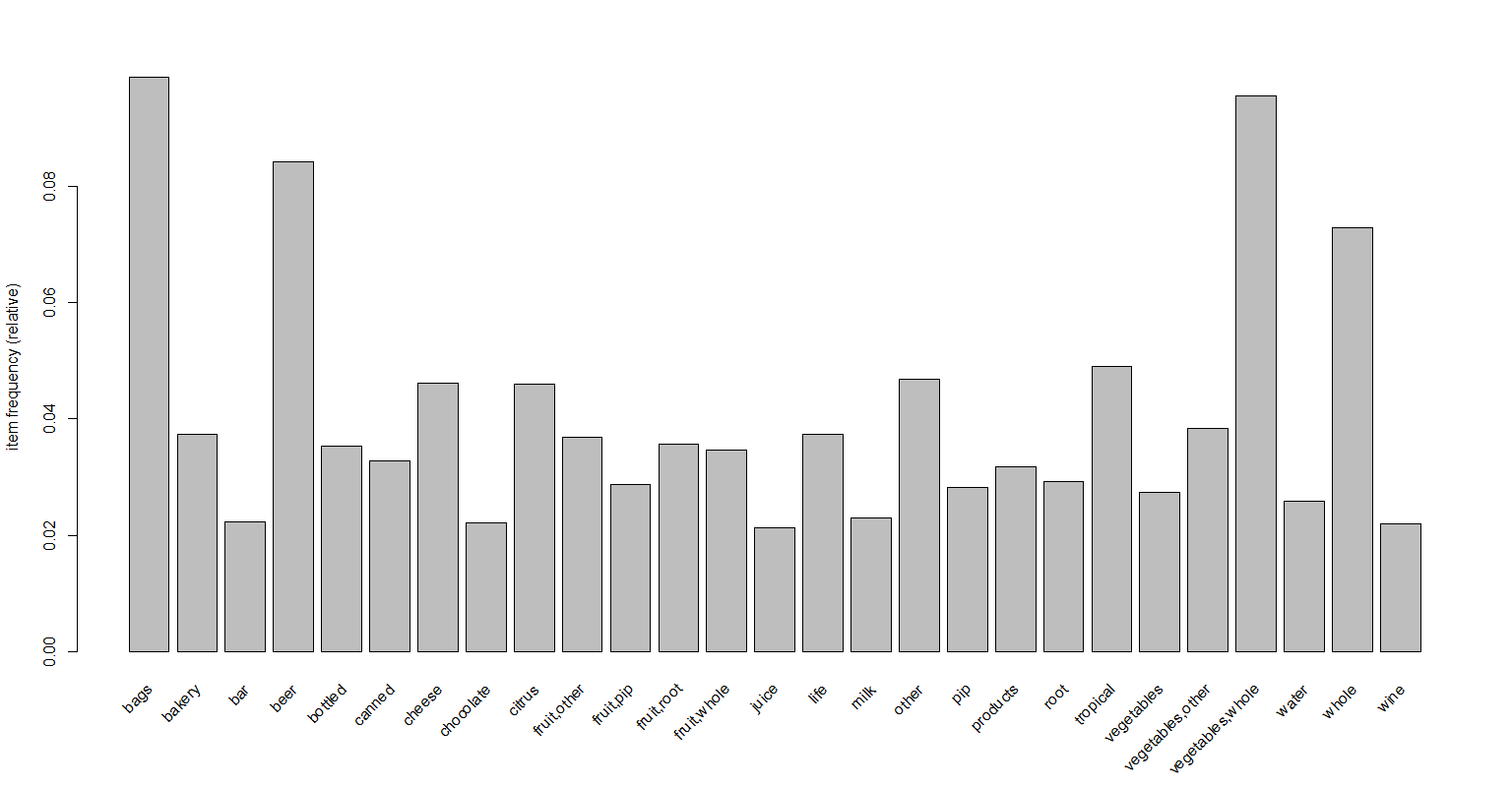
itemFrequency(groceries[, 1:3])

#(appetizer) (appetizer),bathroom (appetizer),cake

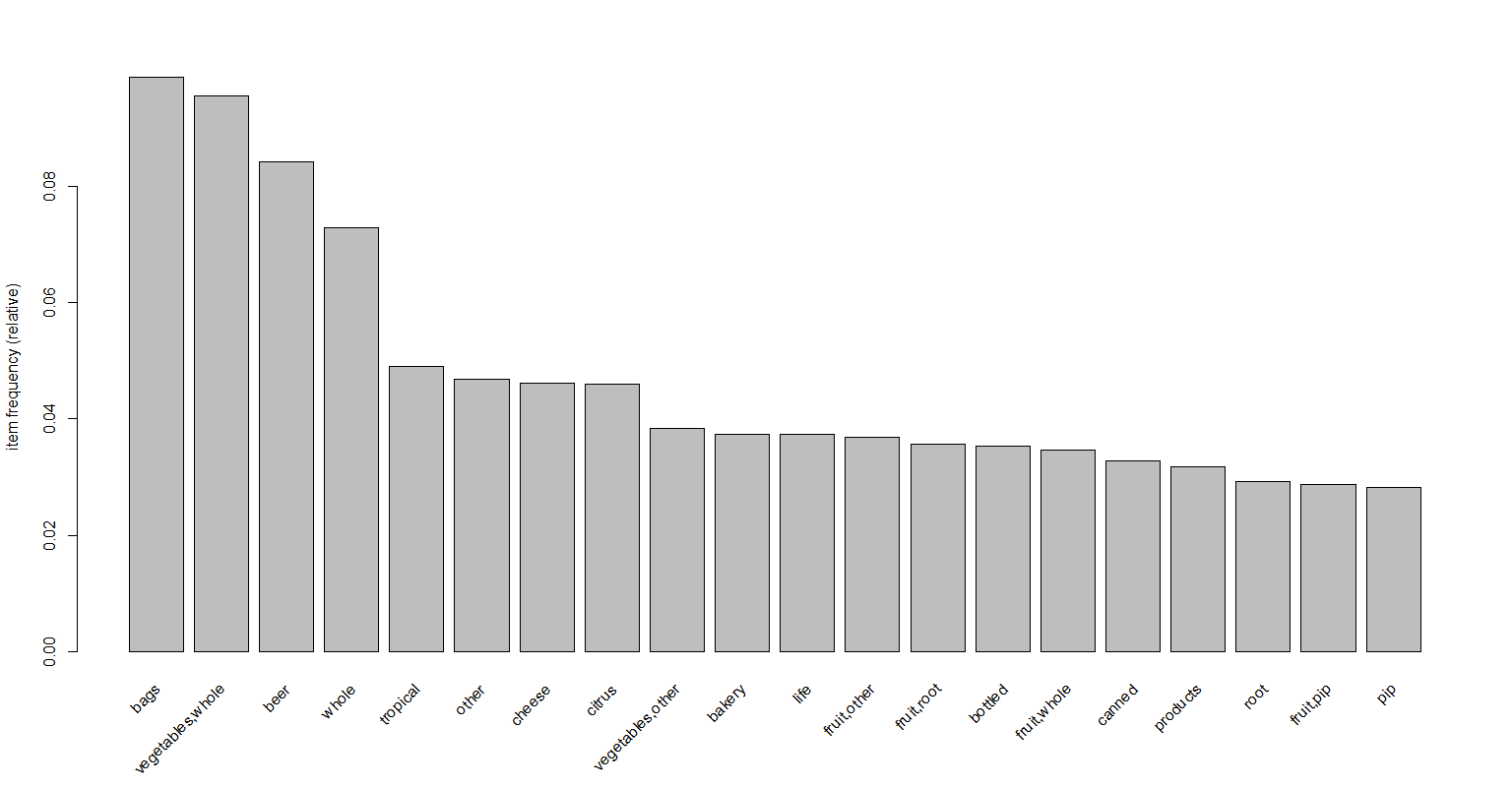
#0.0026436197 0.0001016777 0.0001016777

#plot the frequency of items

itemFrequencyPlot(groceries, support = 0.02)



itemFrequencyPlot(groceries, topN = 20)



**#Step 2: Training a model on the data**

#install.packages("arulesViz")

library("arulesViz") # for visualizing rules

# making rules using apriori algorithm

# Keep changing support and confidence values to obtain different rules

?apriori

**#default settings result in zero rules learned**

arules <- apriori(groceries)

#Apriori

#Parameter specification:

# confidence minval smax arem aval originalSupport maxtime support minlen

#0.8 0.1 1 none FALSE TRUE 5 0.1 1

#maxlen target ext

#10 rules TRUE

#Algorithmic control:

# filter tree heap memopt load sort verbose

#0.1 TRUE TRUE FALSE TRUE 2 TRUE

#Absolute minimum support count: 983

#set item appearances ...[0 item(s)] done [0.00s].

#set transactions ...[6928 item(s), 9835 transaction(s)] done [0.03s].

#sorting and recoding items ... [0 item(s)] done [0.00s].

#creating transaction tree ... done [0.00s].

#checking subsets of size 1 done [0.00s].

#writing ... [0 rule(s)] done [0.00s].

#creating S4 object ... done [0.00s].0

#set better support and confidence levels to learn more rules

# Building rules using apriori algorithm support=0.002,confidence=0.7

arules <- apriori(groceries, parameter = list(support=0.002,confidence=0.7))

arules

#Apriori

#Parameter specification:

# confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

#0.7 0.1 1 none FALSE TRUE 5 0.002 1 10 rules TRUE

#Algorithmic control:

# filter tree heap memopt load sort verbose

#0.1 TRUE TRUE FALSE TRUE 2 TRUE

#Absolute minimum support count: 30

#set item appearances ...[0 item(s)] done [0.00s].

#set transactions ...[655 item(s), 15295 transaction(s)] done [0.01s].

#sorting and recoding items ... [297 item(s)] done [0.00s].

#creating transaction tree ... done [0.01s].

#checking subsets of size 1 2 3 4 done [0.00s].

#writing ... [353 rule(s)] done [0.00s].

#creating S4 object ... done [0.00s].

inspect(head(sort(arules,by="lift")))# to view we use inspect

#lhs rhs support confidence coverage lift count

#[1] {citrus.fruit=citrus fruit,

# margarine=pip fruit} => {semi.finished.bread=tropical fruit} 0.002026806 1.0000000 0.002026806 42.60446 31

#[2] {semi.finished.bread=root vegetables,

# ready.soups=whole milk} => {margarine=other vegetables} 0.003988231 0.8243243 0.004838182 28.58966 61

#[3] {semi.finished.bread=other vegetables,

# ready.soups=butter} => {margarine=whole milk} 0.002157568 0.9705882 0.002222949 26.65197 33

#[4] {semi.finished.bread=sausage} => {citrus.fruit=frankfurter} 0.006472703 1.0000000 0.006472703 26.37069 99

#[5] {citrus.fruit=root vegetables,

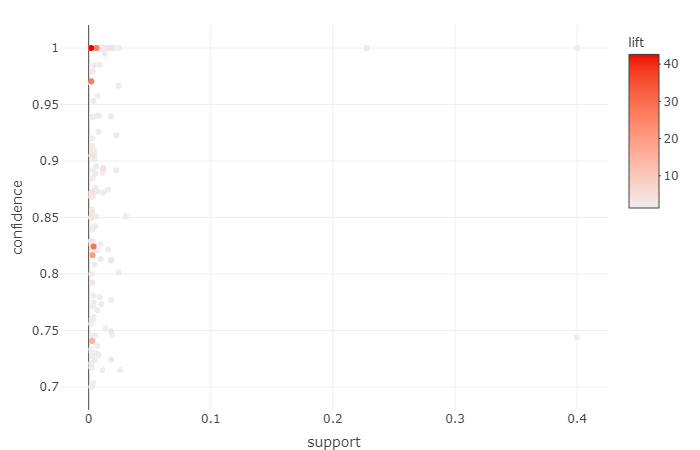
# margarine=whole milk} => {semi.finished.bread=other vegetables} 0.003203661 0.8166667 0.003922851 20.14664 49

#[6] {citrus.fruit=other vegetables,

# margarine=yogurt} => {semi.finished.bread=whole milk} 0.002615234 0.7407407 0.003530566 15.24849 40

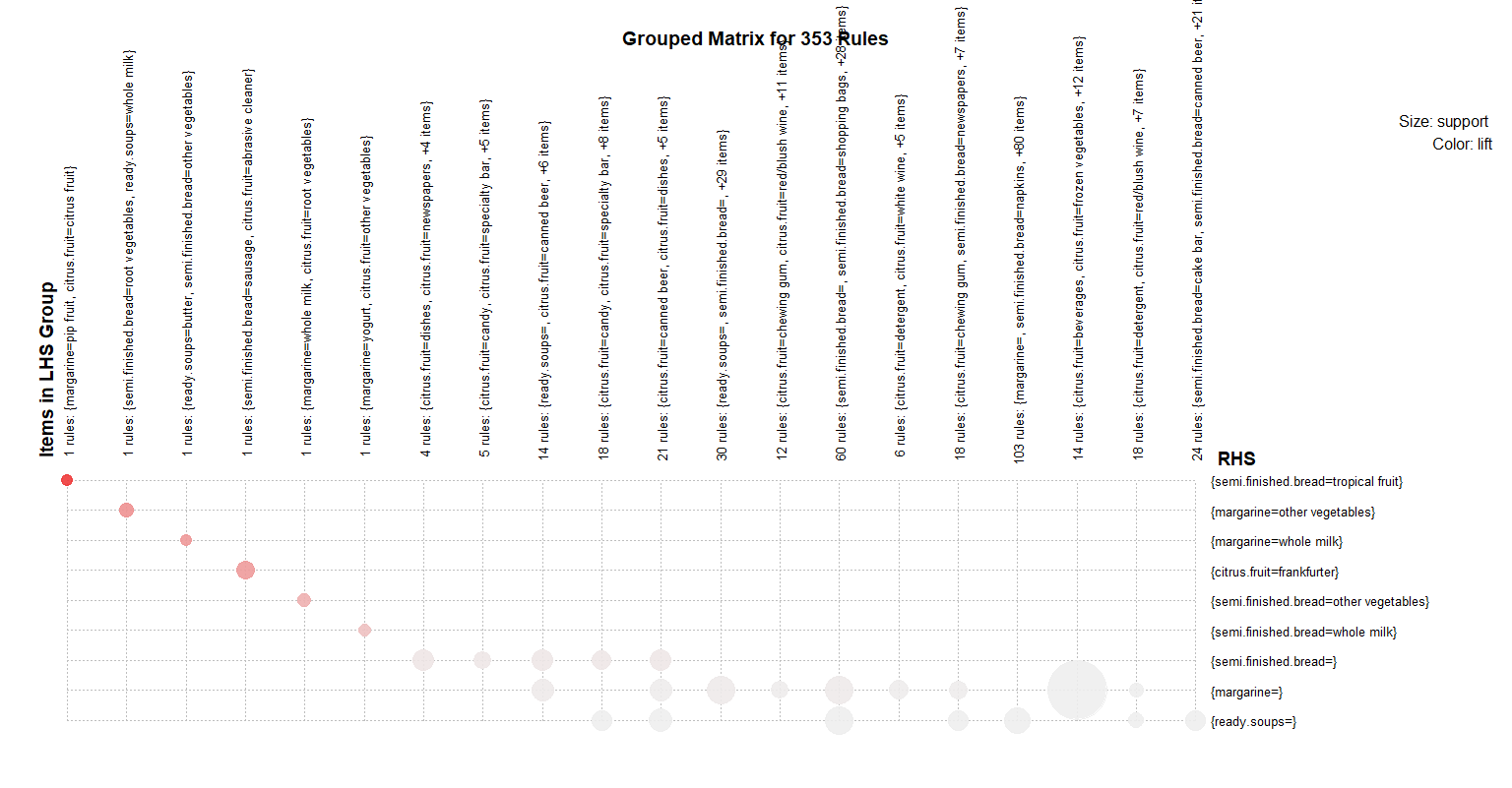
inspect(sort(arules,by="lift"))

plotly\_arules(arules)



**# Different Ways of Visualizing Rules**

plot(arules,method="grouped")



#############Association with support = 0.002,confidence = 0.05,minlen=3###################

groceries\_rules<-apriori(groceries,parameter = list(support = 0.002,confidence = 0.05,minlen=3))

#Apriori

#Parameter specification:

# confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

#0.05 0.1 1 none FALSE TRUE 5 0.002 3 10 rules TRUE

#Algorithmic control:

# filter tree heap memopt load sort verbose

#0.1 TRUE TRUE FALSE TRUE 2 TRUE

#Absolute minimum support count: 19

#set item appearances ...[0 item(s)] done [0.00s].

#set transactions ...[6928 item(s), 9835 transaction(s)] done [0.04s].

#sorting and recoding items ... [257 item(s)] done [0.00s].

#creating transaction tree ... done [0.00s].

#checking subsets of size 1 2 3 4 done [0.00s].

#writing ... [118 rule(s)] done [0.00s].

#creating S4 object ... done [0.00s].

**#Step 3: Evaluating model performance**

# summary of grocery association rules

summary(groceries\_rules)

#set of 118 rules

#rule length distribution (lhs + rhs):sizes

#3 4

#114 4

#Min. 1st Qu. Median Mean 3rd Qu. Max.

#3.000 3.000 3.000 3.034 3.000 4.000

#summary of quality measures:

# support confidence coverage lift

#Min. :0.002034 Min. :0.05435 Min. :0.002034 Min. : 0.8946

#1st Qu.:0.002237 1st Qu.:0.18391 1st Qu.:0.002771 1st Qu.: 5.6638

#Median :0.002440 Median :0.43396 Median :0.005897 Median : 12.1399

#Mean :0.003091 Mean :0.55580 Mean :0.011199 Mean : 19.4259

#3rd Qu.:0.002822 3rd Qu.:1.00000 3rd Qu.:0.014438 3rd Qu.: 26.7255

#Max. :0.013218 Max. :1.00000 Max. :0.037417 Max. :185.5660

#count

#Min. : 20.00

#1st Qu.: 22.00

#Median : 24.00

#Mean : 30.40

#3rd Qu.: 27.75

#Max. :130.00

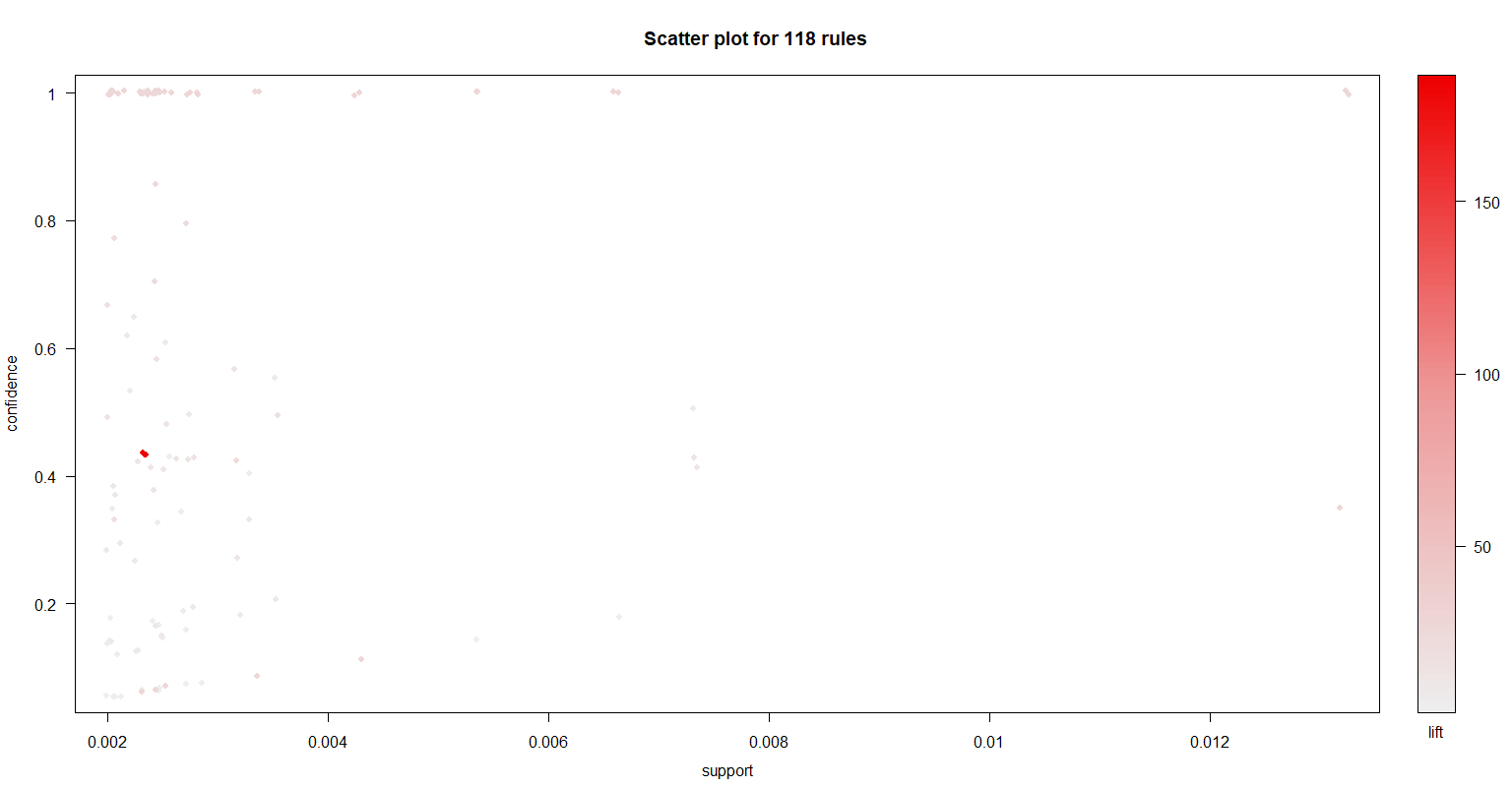
#mining info:

# data ntransactions support confidence

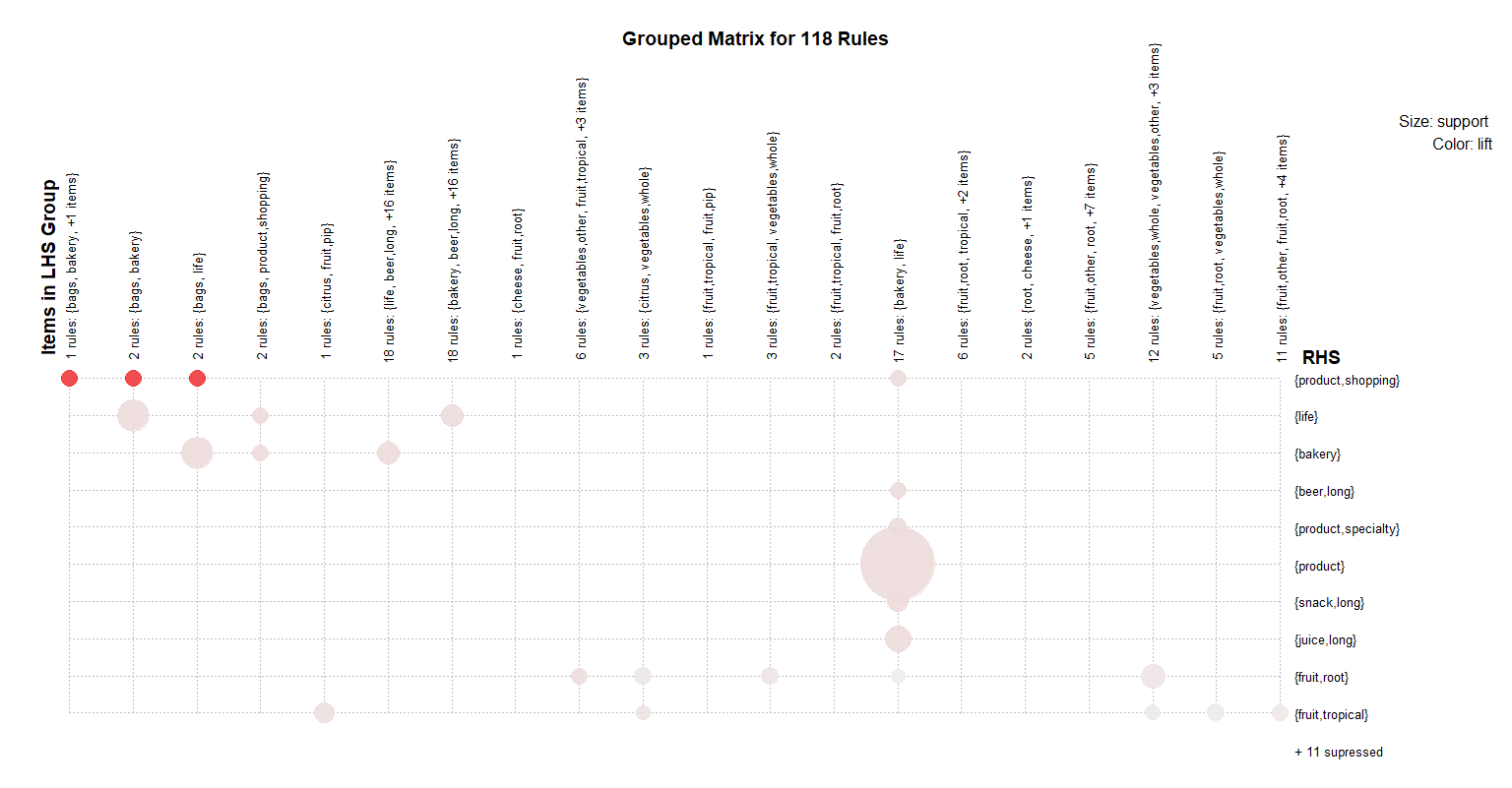
#groceries 9835 0.002 0.05

**#Step 5:Visualization**

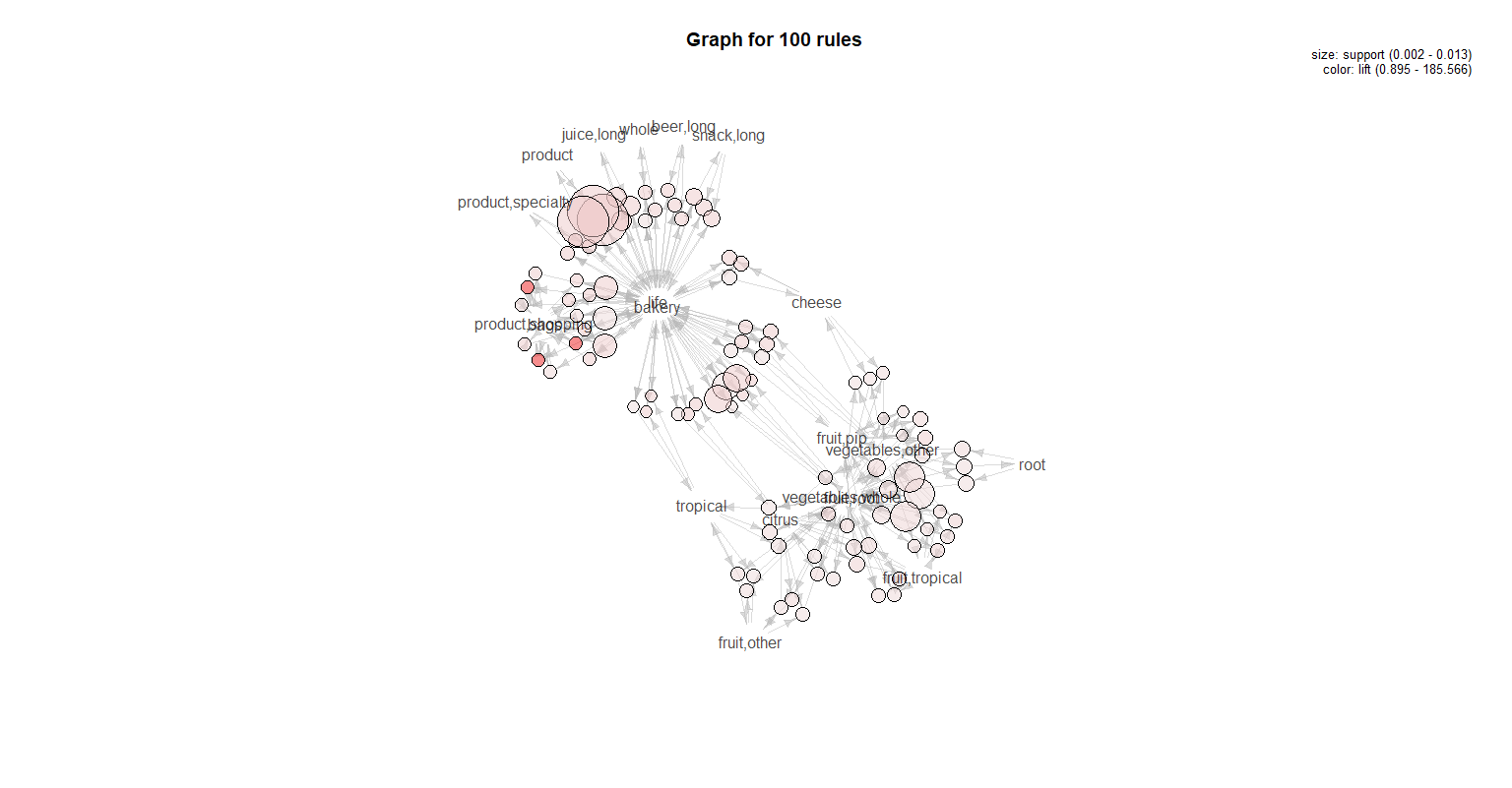
plot(groceries\_rules,method = "scatterplot")



plot(groceries\_rules,method = "grouped")



plot(groceries\_rules,method = "graph")



#look at the first five rules

inspect(groceries\_rules[1:5])

#lhs rhs support confidence coverage lift count

#[1] {bakery,product,shopping} => {life} 0.002338587 1.0000 0.002338587 26.72554 23

#[2] {life,product,shopping} => {bakery} 0.002338587 1.0000 0.002338587 26.72554 23

#[3] {bakery,life} => {product,shopping} 0.002338587 0.0625 0.037417387 26.72554 23

#[4] {bakery,product,shopping} => {bags} 0.002338587 1.0000 0.002338587 10.12873 23

#[5] {bags,product,shopping} => {bakery} 0.002338587 1.0000 0.002338587 26.72554 23

**#Step 6: Improving model performance**

#sorting grocery rules by lift

rules <- sort(groceries\_rules,by="lift")

inspect(rules[1:4])

#lhs rhs support confidence coverage lift count

#[1] {bags,bakery} => {product,shopping} 0.002338587 0.4339623 0.005388917 185.56604 23

#[2] {bags,life} => {product,shopping} 0.002338587 0.4339623 0.005388917 185.56604 23

#[3] {bags,bakery,life} => {product,shopping} 0.002338587 0.4339623 0.005388917 185.56604 23

#[4] {bakery,product,shopping} => {life} 0.002338587 1.0000000 0.002338587 26.72554 23

##################################################################################

**#Association with support =0.001,confidence = 0.06,minlen=4**

groceries\_rules1<-apriori(groceries,parameter = list(support = 0.001,confidence = 0.06,minlen=4))

#Apriori

#Parameter specification:

# confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

#0.06 0.1 1 none FALSE TRUE 5 0.001 4 10 rules TRUE

#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 ...[6928 item(s), 9835 transaction(s)] done [0.03s].

#sorting and recoding items ... [483 item(s)] done [0.00s].

#creating transaction tree ... done [0.00s].

#checking subsets of size 1 2 3 4 done [0.00s].

#writing ... [64 rule(s)] done [0.00s].

#creating S4 object ... done [0.00s].

**#Step : Evaluating model performance**

# summary of grocery association rules

summary(groceries\_rules1)

#set of 64 rules

#rule length distribution (lhs + rhs):sizes

#4

#64

#Min. 1st Qu. Median Mean 3rd Qu. Max.

#4 4 4 4 4 4

#summary of quality measures:

# support confidence coverage lift count

#Min. :0.001017 Min. :0.08462 Min. :0.001017 Min. : 1.127 Min. :10.00

#1st Qu.:0.001118 1st Qu.:0.32143 1st Qu.:0.001220 1st Qu.: 7.505 1st Qu.:11.00

#Median :0.001220 Median :0.60556 Median :0.002237 Median : 21.159 Median :12.00

#Mean :0.001296 Mean :0.62651 Mean :0.003462 Mean : 22.118 Mean :12.75

#3rd Qu.:0.001271 3rd Qu.:1.00000 3rd Qu.:0.004550 3rd Qu.: 26.726 3rd Qu.:12.50

#Max. :0.002339 Max. :1.00000 Max. :0.013218 Max. :185.566 Max. :23.00

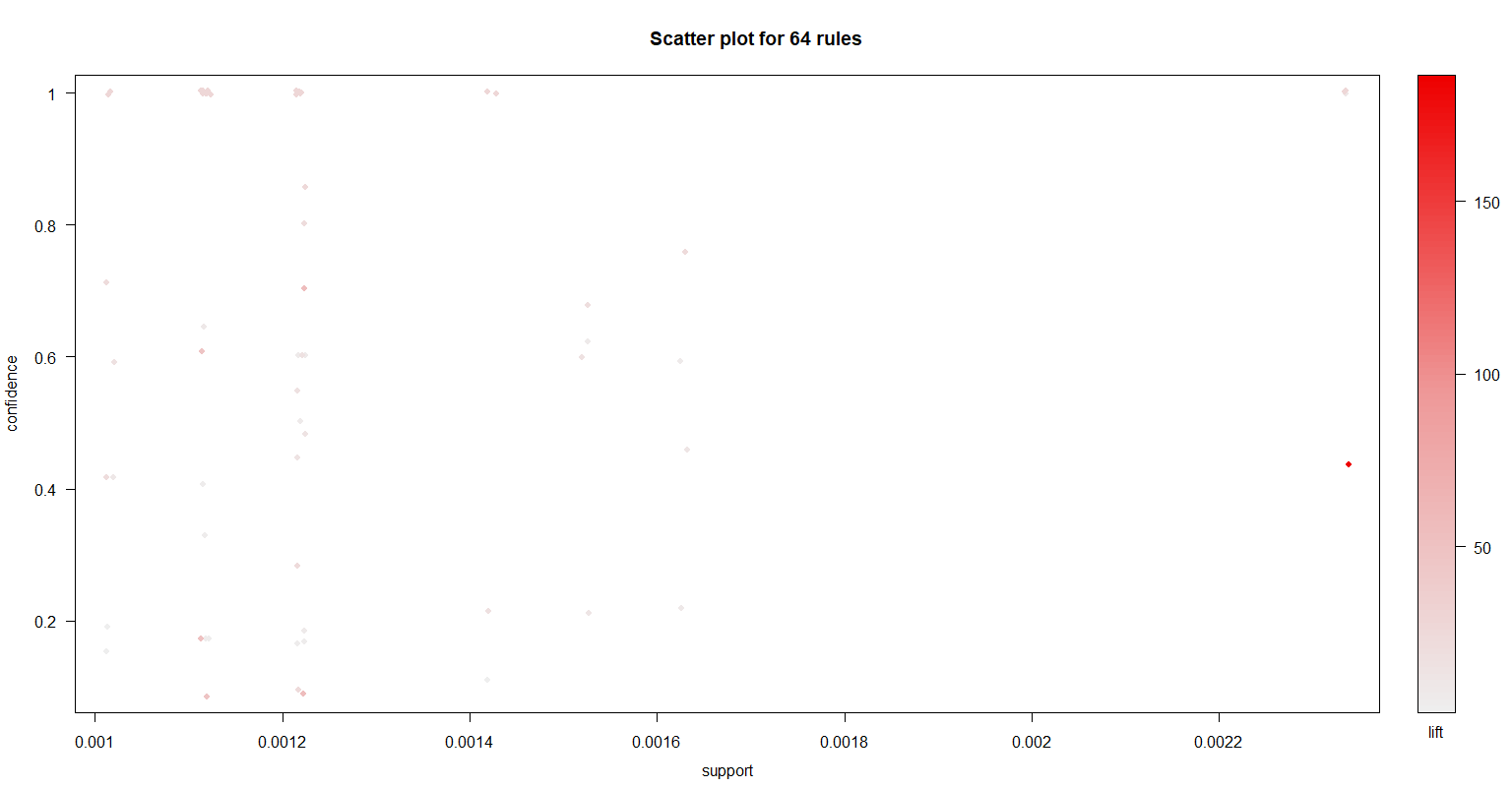
#mining info:

# data ntransactions support confidence

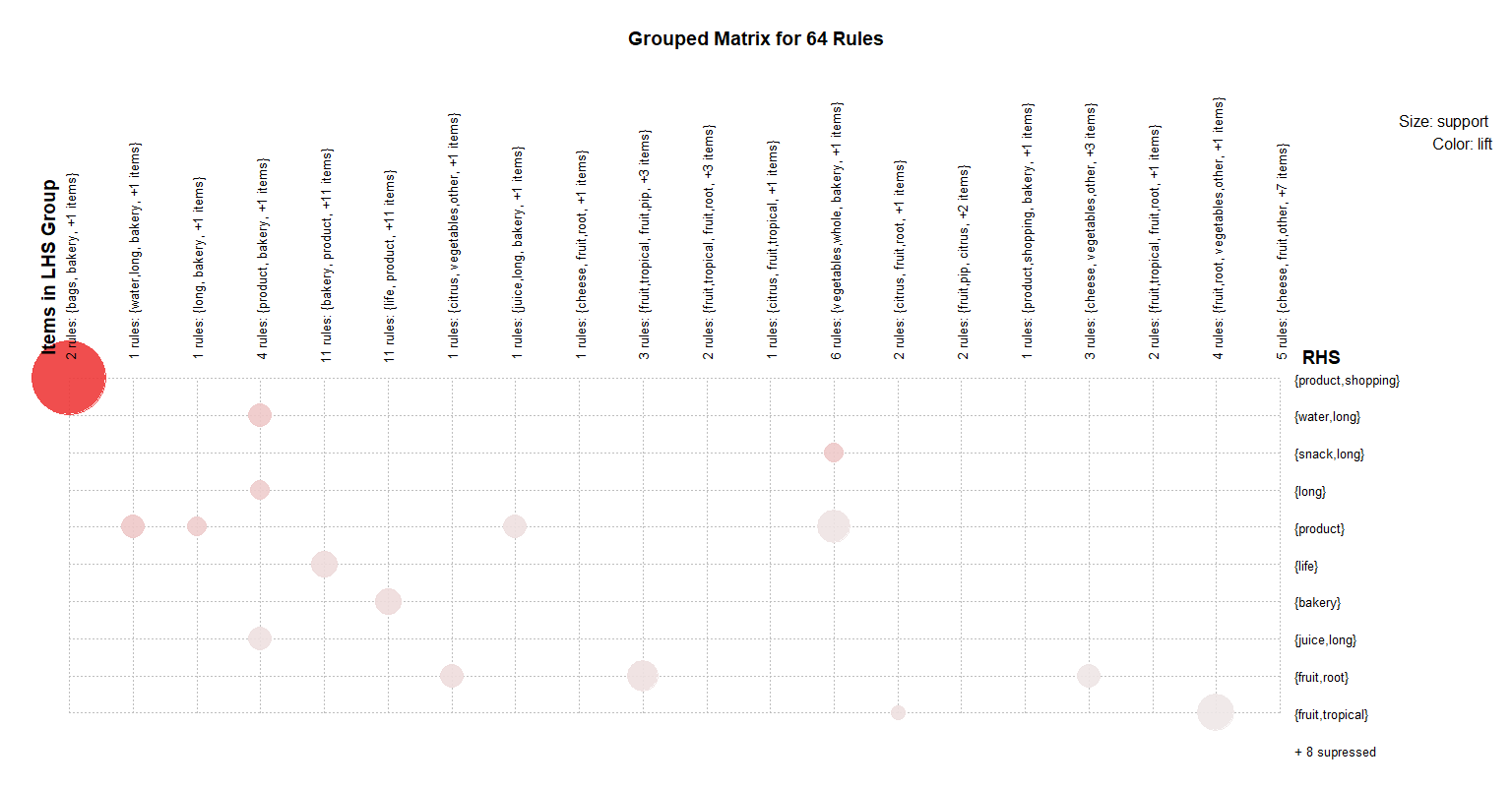
#groceries 9835 0.001 0.06

**#Step:Visualization**

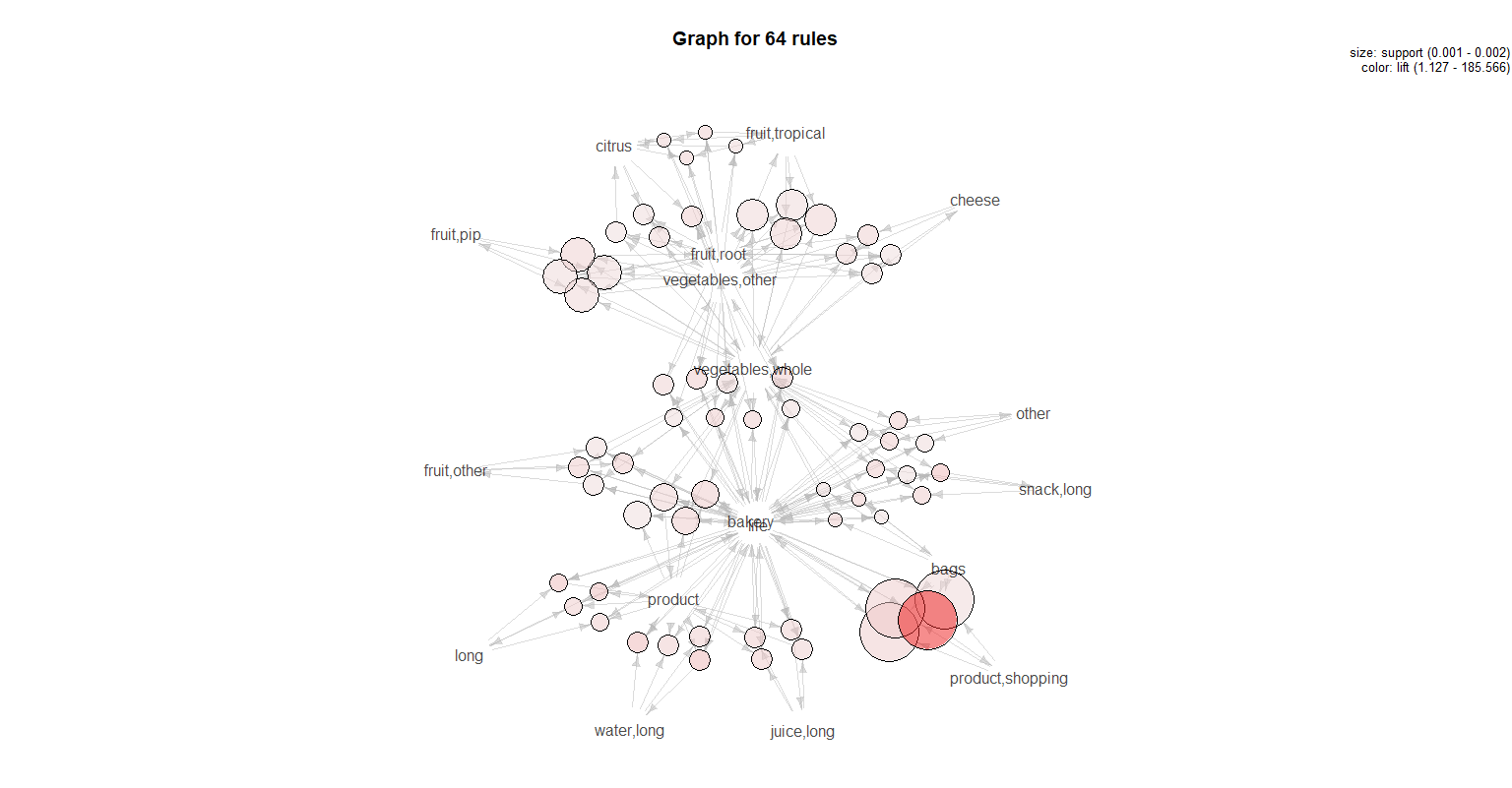
plot(groceries\_rules1,method = "scatterplot")



plot(groceries\_rules1,method = "grouped")



plot(groceries\_rules1,method = "graph")



#look at the first five rules

inspect(groceries\_rules1[1:5])

#lhs rhs support confidence coverage lift count

#[1] {bakery,long,product} => {life} 0.001118454 1.00000000 0.001118454 26.72554 11

#[2] {life,long,product} => {bakery} 0.001118454 1.00000000 0.001118454 26.72554 11

#[3] {bakery,life,long} => {product} 0.001118454 0.61111111 0.001830198 46.23291 11

#[4] {bakery,life,product} => {long} 0.001118454 0.08461538 0.013218099 46.23291 11

#[5] {bakery,product,water,long} => {life} 0.001220132 1.00000000 0.001220132 26.72554 12

**#Step: Improving model performance**

#sorting grocery rules by lift

rules1 <- sort(groceries\_rules1,by="lift")

inspect(rules1[1:4]

#lhs rhs support confidence coverage lift count

#[1] {bags,bakery,life} => {product,shopping} 0.002338587 0.43396226 0.005388917 185.56604 23

#[2] {bakery,life,water,long} => {product} 0.001220132 0.70588235 0.001728521 53.40271 12

#[3] {bakery,life,product} => {water,long} 0.001220132 0.09230769 0.013218099 53.40271 12

#[4] {bakery,life,vegetables,whole} => {snack,long} 0.001118454 0.16923077 0.006609049 50.43590 11

#The first rule, with a lift of about 185.56604,

#implies that people who buy bags,bakery,life are nearly four times more likely to product,shopping **.**

**#writing the rules to a CSV file**

write(rules1, file="a\_rules.csv",sep=",")

getwd()