Market Basket Analysis

Godfred Somua-Gyimah

March 15, 2017

Contents

INTRODUCTION

One of the key techniques used by large retailers is called Market Basket Analysis, which uncovers associations between products by looking for combinations of products that frequently co-occur in transactions. In other words, it allows retailers to identify relationships between the products that people buy together. Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items.For example, customers that buy a pencil and paper are likely to buy a rubber or ruler. In this project, we will be using the Apriori Algorithm to generate a set of rules that link two or more products together.

Other Application Areas of this analysis include:

1. Analysis of credit card purchases.
2. Analysis of telephone calling patterns.
3. Identification of fraudulent medical insurance claims.
4. Analysis of telecom service purchases.

Note: In order to run this demo, the following R packages must be installed in your R environment:

- arules: mining association rules
- magrittr: forward pipe operator
- arulesViz: data visualization of association rules
- RColorBrewer: color palettes for plots

```
# Clean the environment
rm(list = ls())
# Load the arules package for mining association rules
library(arules) # mining association rules
```

```
## Warning: package 'arules' was built under R version 3.3.3
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

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

```
library(magrittr) # forward pipe operator
```

```
## Warning: package 'magrittr' was built under R version 3.3.3
```

1. Understand Dataset

Suppose we extract transaction-item relationship from a transactional database. The dataset is stored in a csv file. Now we use read.csv() method to read in the raw dataset.

```
# Read in transaction dataset
df<- read.csv("groceries_raw.csv")
```

```
# Show the head of the raw dataset
head(df)
```

```
##   TransactionID                Item        ItemCategory1
## 1             1        citrus fruit fruit and vegetables
## 2             1 semi-finished bread       fresh products
## 3             1           margarine       processed food
## 4             1         ready soups       processed food
## 5             2      tropical fruit fruit and vegetables
## 6             2              yogurt       fresh products
##            ItemCategory2
## 1                  fruit
## 2 bread and backed goods
## 3           vinegar/oils
## 4           soups/sauces
## 5                  fruit
## 6          dairy produce
```

The first column indicates the transaction ID. The second column is the item included in the transaction. The third and fourth columns are item categories ate different levels. As the above table shows, there are four items (citrus fruit, semi-finished bread, margarine, and ready soup) in the first transaction.

Let's check the structure of the dataset.

```
# Show the structure of the dataset
str(df)
```

```
## 'data.frame':    43367 obs. of  4 variables:
##  $ TransactionID: int  1 1 1 1 2 2 2 3 4 4 ...
##  $ Item         : Factor w/ 169 levels "abrasive cleaner",..: 30 133 89 119 158 168 34 167 110 168 .
```

```
##  $ ItemCategory1: Factor w/ 10 levels "canned food",..: 5 4 9 9 5 4 3 4 5 4 ...
##  $ ItemCategory2: Factor w/ 55 levels "baby food","bags",..: 25 7 54 49 25 18 15 18 25 18 ...
```

Because the transaction ID should be a categorical variable, we change it as a factor.

```
# Show the structure of the dataset
df$TransactionID <- factor(df$TransactionID)
```

Let's check the structure of the dataset again.

```
# Show the structure of the dataset
str(df)
```

```
## 'data.frame':    43367 obs. of  4 variables:
##  $ TransactionID: Factor w/ 9835 levels "1","2","3","4",..: 1 1 1 1 2 2 2 3 4 4 ...
##  $ Item         : Factor w/ 169 levels "abrasive cleaner",..: 30 133 89 119 158 168 34 167 110 168 .
##  $ ItemCategory1: Factor w/ 10 levels "canned food",..: 5 4 9 9 5 4 3 4 5 4 ...
##  $ ItemCategory2: Factor w/ 55 levels "baby food","bags",..: 25 7 54 49 25 18 15 18 25 18 ...
```

As the data structure shows, the raw dataset contains 9835 transactions containing combination of 169 items. Those items belong to 10 categories at level 1, and 55 categoris at level 2.

```
# Show all level 1 categories
levels(df$ItemCategory1)
```

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

2. Assocition Rules at Item Category Level

Now, we choose the item category 1 as the level of items. We need to construct a transactions object. So, we use the read.transactions() method in arules package to read the transaction data file from disk and creates a transactions object at the item category 1 level. Notice that the item category 1 is stored in the 3rd column. We need to use "rm.duplicates = TRUE" to remove duplicates since the transactin raw dataset contains two or more items in a transaction that belong to the same category 1.

```
trans_cat2<- read.transactions(file = "groceries_raw.csv",
                               format = "single", sep = ",",rm.duplicates = TRUE,
                               cols = c(1,3), skip = 1)
```

```
## distribution of transactions with duplicates:
## items
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2100 1135  778  512  351  196  119  110   66   43   29   18    9   13    1
##   16   17   18   19   20   22   26
##    3    3    1    2    2    1    1
```

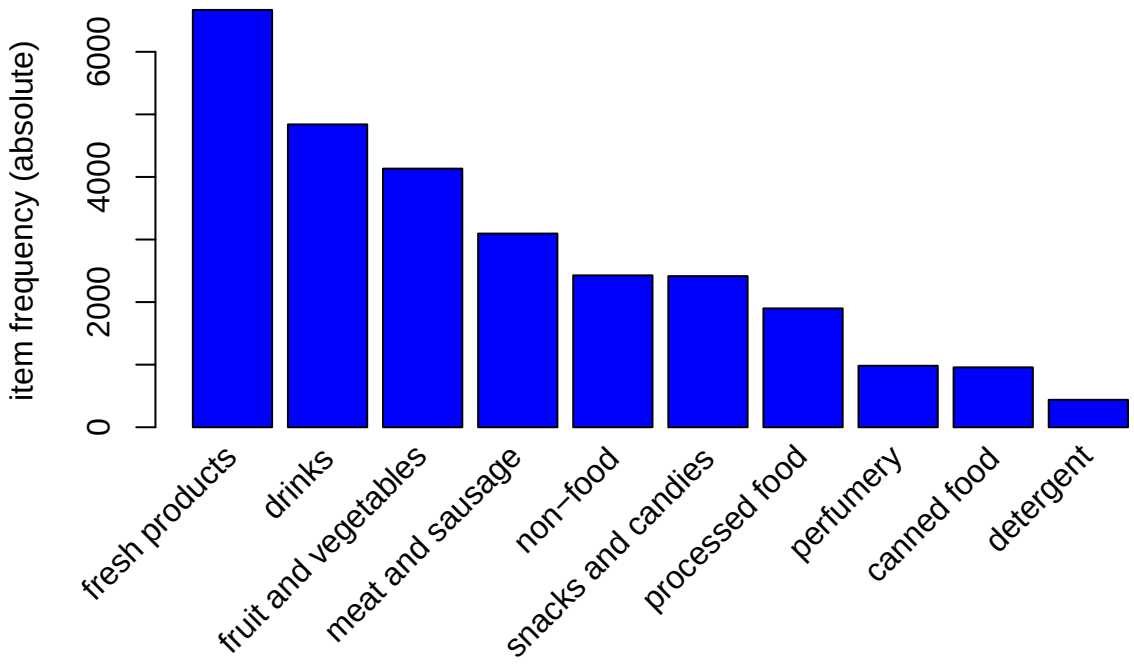
```
# Show a summary of the transactions dataset
trans_cat2
```

```
## transactions in sparse format with
##  9835 transactions (rows) and
##  10 items (columns)
```

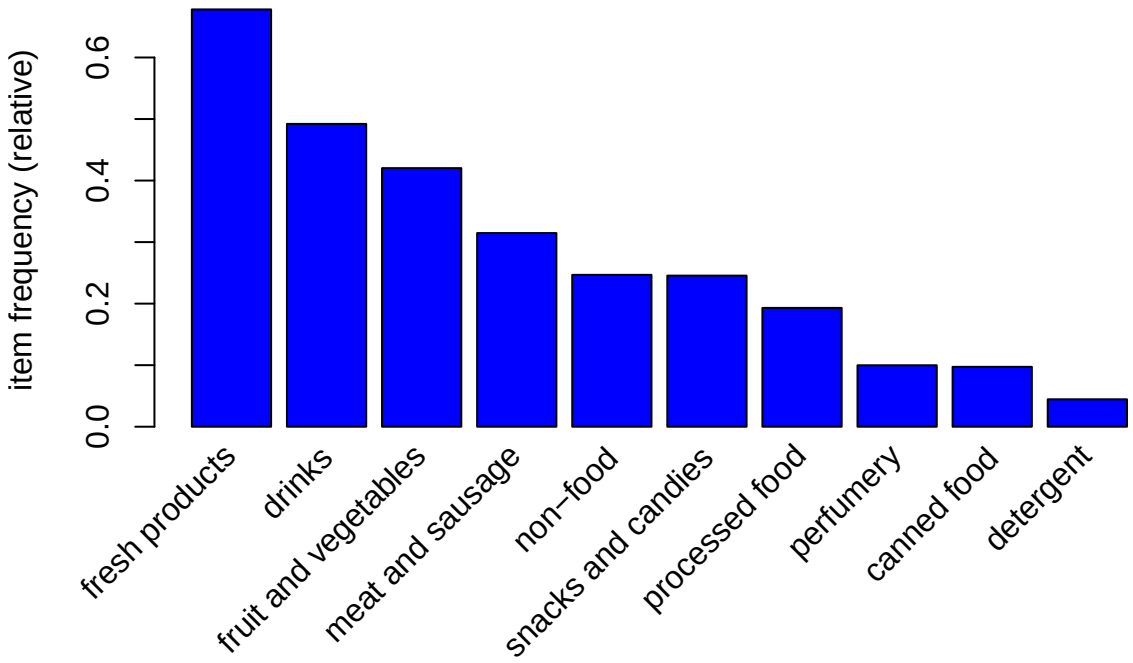
We can see that the transactions dataset contains 9835 transactions of 10 item categories.

Now, let's check all 10 item categories that are bought in those transactions.

```
itemFrequencyPlot(trans_cat2,topN=10,col="blue",type="absolute")
```

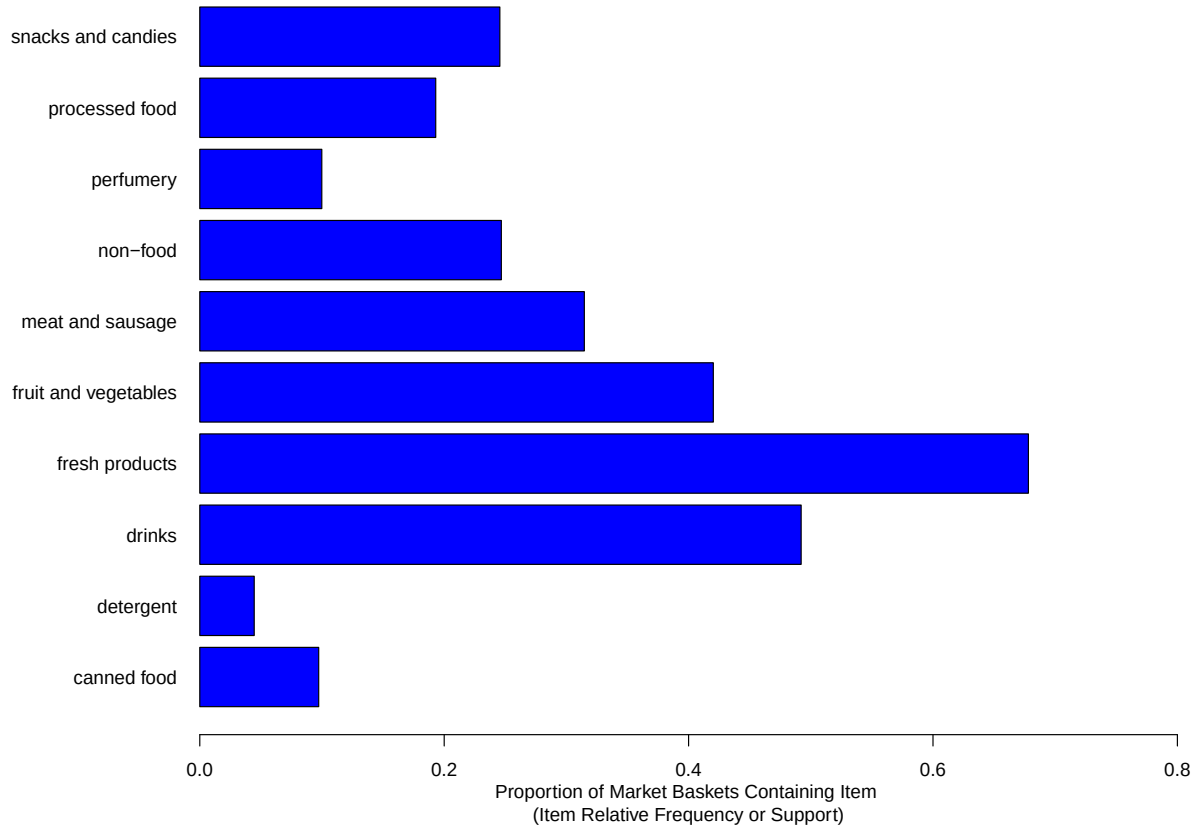



```
itemFrequencyPlot(trans_cat2,topN=10,col="blue",type="relative")
```



Plot item frequency / support for all 10 item categories.

```
itemFrequencyPlot(trans_cat2, cex.names=1, xlim = c(0,0.8),
  type = "relative", horiz = TRUE, col = "blue", las = 1,
  xlab = paste("Proportion of Market Baskets Containing Item",
    "\n(Item Relative Frequency or Support)"))
```



Now, let's call the apriori() method to generate association rules. We set the minimun support as 0.001 and the minimum confidence as 0.05.

```
# Mine frequent itemsets, association rules or association hyperedges using the Apriori algorithm.
first.rules <- apriori(trans_cat2,
                       parameter = list(support = 0.001, confidence = 0.05))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    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 ...[10 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
```

```
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [4294 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Show summary of the 1st set of association rules.

```
summary(first.rules)
```

```
## set of 4294 rules
##
## rule length distribution (lhs + rhs):sizes
##    1    2    3    4    5    6    7    8
##    9   89  360  840 1230 1092  546  128
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   5.000   5.164   6.000   8.000
##
## summary of quality measures:
##     support           confidence          lift
##  Min.   :0.001017   Min.   :0.05293   Min.   :0.9443
##  1st Qu.:0.002034   1st Qu.:0.35795   1st Qu.:1.5999
##  Median :0.003965   Median :0.54195   Median :2.0362
##  Mean   :0.011018   Mean   :0.54449   Mean   :2.3103
##  3rd Qu.:0.009354   3rd Qu.:0.71388   3rd Qu.:2.7167
##  Max.   :0.678088   Max.   :1.00000   Max.   :8.5855
##
## mining info:
##        data ntransactions support confidence
##  trans_cat2          9835   0.001       0.05
```

We notice that the Apriori algorithm detects 4294 rules from the dataset by using the parameters (minimun support=0.001, minimum confidence=0.05). The rule set is still too many to analyze.

In order to reduce the number of association rules generated, we can enlarge the minimum support and confidence setting. Now, let's set minimun support=0.02 and keep minimum confidence=0.05 and call apriori() method again.

```
# Mine frequent itemsets, association rules or association hyperedges using the Apriori algorithm.
second.rules <- apriori(trans_cat2,
                        parameter = list(support = 0.02, confidence = 0.05))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5    0.02      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: 196
##
## set item appearances ...[0 item(s)] done [0.00s].
```

```
## set transactions ...[10 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [509 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Show summary of the association rule
summary(second.rules)
```

```
## set of 509 rules
##
## rule length distribution (lhs + rhs):sizes
##   1   2   3   4   5
##   9  77 198 180  45
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   3.344   4.000   5.000
##
## summary of quality measures:
##     support          confidence          lift
##  Min.   :0.02013   Min.   :0.05293   Min.   :0.9443
##  1st Qu.:0.02532   1st Qu.:0.34812   1st Qu.:1.2743
##  Median :0.03538   Median :0.49855   Median :1.4686
##  Mean   :0.05566   Mean   :0.50774   Mean   :1.5057
##  3rd Qu.:0.05928   3rd Qu.:0.67059   3rd Qu.:1.7105
##  Max.   :0.67809   Max.   :0.95067   Max.   :2.3370
##
## mining info:
##        data ntransactions support confidence
##  trans_cat2          9835    0.02       0.05
```

Now, we get 509 rules, much less than the 1st set of 4294 rules. However, some rules have lift values less than 1.0.

Generally, a lift value less than 1.0 impllies that the RHS item(s) is unlikely to be bought when the LHS item(s) are bought. In contrast, a lift greater than 1.0 implies that the RHS item(s) is likely to be bought when the item(s) on the LHS are bought. So, we will later subset the second.rules to select only rules with high fidelity and confidence.

A picture says a thousand words. For now, we can visualize the association rules. To do that, we first need to load two packages: "arulesViz"" for association rules plot, and "RColorBrewer" for generating color palettes for graphs. Note that "arulesViz" requires the 'grid' package to be loaded also.

```
library(grid)
library(arulesViz)  # data visualization of association rules
```

```
## Warning: package 'arulesViz' was built under R version 3.3.3
```

```
library(RColorBrewer)  # color palettes for plots
```

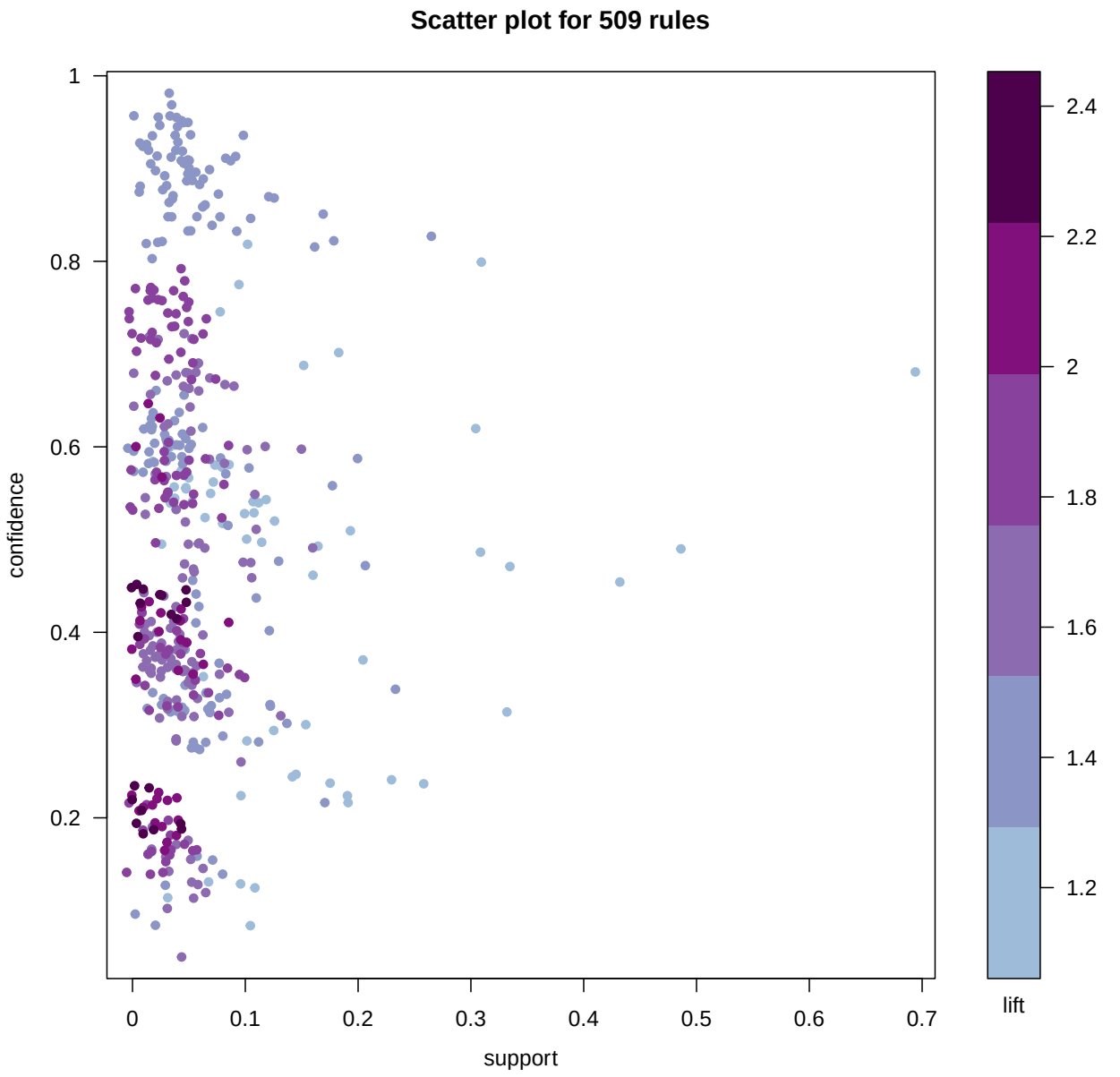
Grouped matrix-based visualization of all association rules.

```
# grouped matrix of rules
plot(second.rules,
     method="grouped",
     control=list(col = rev(brewer.pal(9, "Blues")[4:9]), main = ""))
```

Draw all 509 rules in a scatter plot.

```
# Data visualization of association rules in scatter plot
plot(second.rules,
     control=list(jitter=2, col = rev(brewer.pal(9, "BuPu")[4:9])),
```

```
     shading = "lift")
```



From the scatter plot, while most items with very high lift values (>1.6) are bought less frequently (less than 10% of all transactions), other items with relatively lower lift values (1.2 - 1.6) are so popular that they have the potential to significantly influence profit margins.

Therefore, we will consider a subset of second.rules which are the rules for items that are bought atleast 10% of the time in all transactions (support = 0.10) with very strong association rules (i.e. lift >= 1.0)

```
second_sub <- subset(second.rules, lift >= 1.0)
second_sub <- subset(second_sub, support >= 0.10)

inspect(second_sub[1:20])
```

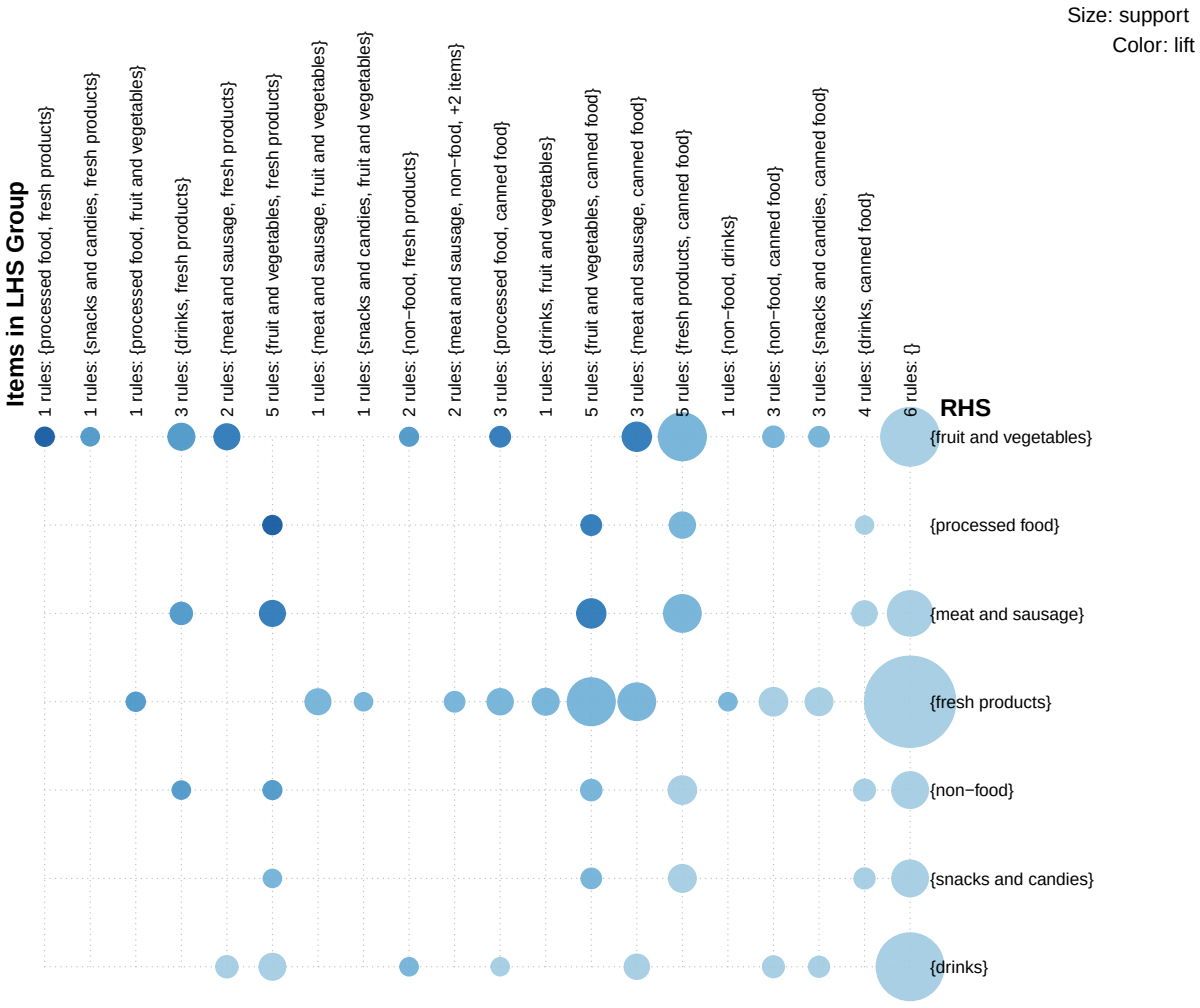
```
##      lhs                         rhs                    support    confidence
```

```
## [1]  {}                     => {snacks and candies}   0.2455516 0.2455516
## [2]  {}                     => {non-food}             0.2467717 0.2467717
## [3]  {}                     => {meat and sausage}     0.3146924 0.3146924
## [4]  {}                     => {drinks}               0.4921200 0.4921200
## [5]  {}                     => {fruit and vegetables} 0.4202339 0.4202339
## [6]  {}                     => {fresh products}       0.6780885 0.6780885
## [7]  {processed food}       => {drinks}               0.1005592 0.5208004
## [8]  {drinks}               => {processed food}       0.1005592 0.2043388
## [9]  {processed food}       => {fruit and vegetables} 0.1191662 0.6171669
## [10] {fruit and vegetables} => {processed food}       0.1191662 0.2835713
## [11] {processed food}       => {fresh products}       0.1631927 0.8451817
## [12] {fresh products}       => {processed food}       0.1631927 0.2406658
## [13] {snacks and candies}   => {drinks}               0.1224199 0.4985507
## [14] {drinks}               => {snacks and candies}   0.1224199 0.2487603
## [15] {snacks and candies}   => {fruit and vegetables} 0.1183528 0.4819876
## [16] {fruit and vegetables} => {snacks and candies}   0.1183528 0.2816356
## [17] {snacks and candies}   => {fresh products}       0.1755974 0.7151139
## [18] {fresh products}       => {snacks and candies}   0.1755974 0.2589594
## [19] {non-food}             => {drinks}               0.1286223 0.5212196
## [20] {drinks}               => {non-food}             0.1286223 0.2613636
##      lift
## [1]  1.000000
## [2]  1.000000
## [3]  1.000000
## [4]  1.000000
## [5]  1.000000
## [6]  1.000000
## [7]  1.058279
## [8]  1.058279
## [9]  1.468627
## [10] 1.468627
## [11] 1.246418
## [12] 1.246418
## [13] 1.013067
## [14] 1.013067
## [15] 1.146951
## [16] 1.146951
## [17] 1.054603
## [18] 1.054603
## [19] 1.059131
## [20] 1.059131
```

The results above and the plot below show that the most popular products are drinks, fresh products, fruits and vegetables and these are bought by more than 40% of all customers. For example, almost 68% of customers have fresh products in their shopping baskets, regardless of what else they bought. Atleast 49% of all customers buy both fresh products and drinks whilst 42% of customers buy fruits and vegetables.

The grouped matrix-based visualization below shows all 53 association rules.

```
# grouped matrix of rules
plot(second_sub,
     method="grouped",
     control=list(col = rev(brewer.pal(9, "Blues")[4:9]), main = ""))
```



Now, we will sort out the rules in decreasing order according to their association (lift).

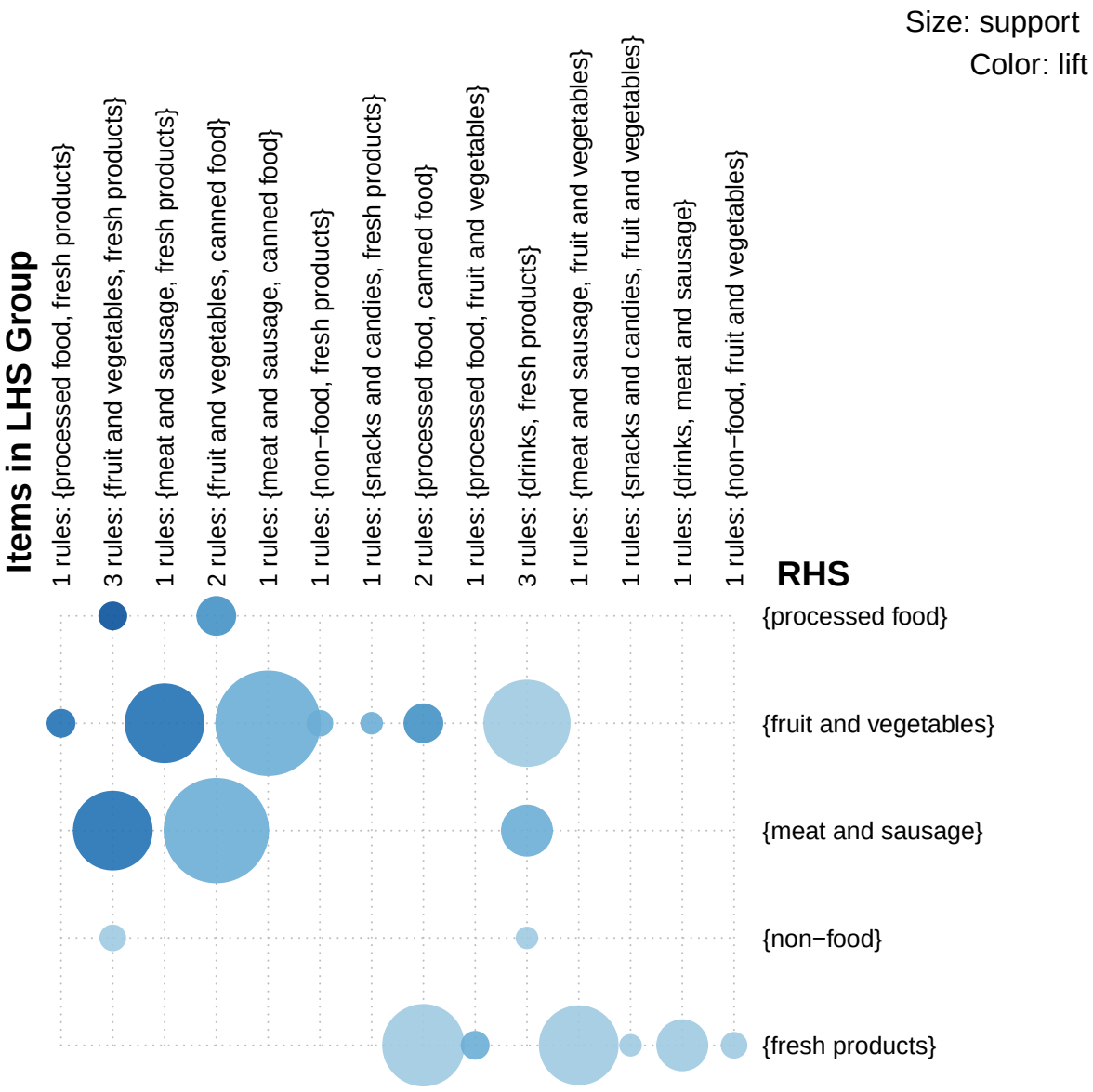
```
# Sort by lift.
top.second_sub <- second_sub %>% sort(decreasing = TRUE, by = "lift") %>% head(53)

# Display the top 20 rules
inspect(top.second_sub[1:20])
```

```
##      lhs                          rhs                        support confidence     lift
## [1]  {fresh products,
##       fruit and vegetables} => {processed food}       0.1078800  0.3222965 1.669187
## [2]  {fresh products,
##       processed food}       => {fruit and vegetables} 0.1078800  0.6610592 1.573075
## [3]  {fresh products,
##       fruit and vegetables} => {meat and sausage}     0.1604474  0.4793439 1.523214
```

```
## [4]  {fresh products,
##       meat and sausage}     => {fruit and vegetables} 0.1604474  0.6332263 1.506843
## [5]  {processed food}       => {fruit and vegetables} 0.1191662  0.6171669 1.468627
## [6]  {fruit and vegetables} => {processed food}       0.1191662  0.2835713 1.468627
## [7]  {meat and sausage}     => {fruit and vegetables} 0.1869853  0.5941842 1.413937
## [8]  {fruit and vegetables} => {meat and sausage}     0.1869853  0.4449552 1.413937
## [9]  {fresh products,
##       non-food}             => {fruit and vegetables} 0.1056431  0.5827257 1.386670
## [10] {fresh products,
##       snacks and candies}   => {fruit and vegetables} 0.1013726  0.5773017 1.373763
## [11] {fruit and vegetables,
##       processed food}       => {fresh products}       0.1078800  0.9052901 1.335062
## [12] {drinks,
##       fresh products}       => {meat and sausage}     0.1317743  0.4181994 1.328915
## [13] {drinks,
##       fresh products}       => {non-food}             0.1013726  0.3217167 1.303702
## [14] {fresh products,
##       fruit and vegetables} => {non-food}             0.1056431  0.3156136 1.278970
## [15] {drinks,
##       fresh products}       => {fruit and vegetables} 0.1681749  0.5337206 1.270056
## [16] {fruit and vegetables,
##       meat and sausage}     => {fresh products}       0.1604474  0.8580750 1.265432
## [17] {fruit and vegetables,
##       snacks and candies}   => {fresh products}       0.1013726  0.8565292 1.263153
## [18] {drinks,
##       meat and sausage}     => {fresh products}       0.1317743  0.8476128 1.250003
## [19] {fruit and vegetables,
##       non-food}             => {fresh products}       0.1056431  0.8460912 1.247759
## [20] {processed food}       => {fresh products}       0.1631927  0.8451817 1.246418
```

We will go ahead to visualize the top 20 association rules as grouped matrix, which will be easier to analyze.

```
# grouped matrix of rules
plot(top.second_sub[1:20],
     method="grouped",
     control=list(col = rev(brewer.pal(9, "Blues")[4:9]), main = ""))
```



From the rules displayed and the above visualization, the following conclusions can be drawn:

1. Customers who buy fruits, vegetables and fresh products are very likely to buy processed foods also. Such customers make up about 10.7% of the total customers.
2. Customers who buy fruits, vegetables and fresh products are also very likely to buy meat and sausage. About 16% of all customers fall into this category.

3. 18.6% of customers will most likely buy fruit and vegetables as well as meat and sausage.
4. Generally, the top 8 rules indicate strong associations between four item categories namely:

- fresh products,
- fruit and vegetables
- meat and sausage
- processed food

5. Customers who buy processed foods and canned foods are also very likey to purchase fruits and vegetables.
6. Similarly, customers who purchase drinks and fresh products are likely to purchase meat and sausage.
7. Non-food items, Snacks and candies are among the least popular items in the store.

Draw the graph of the top 20 association rules.

```
plot(top.second_sub[1:20], method="graph",
  control=list(type="items"),
  shading = "lift")
```

CONCLUSION

The results and analyses above suggest the following:

1.The most popular products at this store are: - drinks - fresh products and - fruits and vegetables

These items are bought by more than 40% of all customers. For example, almost 68% of customers have fresh products in their shopping baskets, regardless of what else they buy. Atleast 49% of all customers buy both fresh products and drinks whilst 42% of customers buy fruits and vegetables.

2. Customers who buy fruits, vegetables and fresh products are very likely to buy processed foods also. Such customers make up about 10.7% of the total customers.
3. Customers who buy fruits, vegetables and fresh products are also very likely to buy meat and sausage. About 16% of all customers fall into this category.
4. 18.6% of customers will most likely buy fruit and vegetables as well as meat and sausage.
5. Customers who buy processed foods and canned foods are also very likey to purchase fruits and vegetables.
6. Similarly, customers who purchase drinks and fresh products are likely to purchase meat and sausage.
7. Generally, the top 8 rules indicate strong associations between four item categories namely:

- fresh products,
- fruit and vegetables
- meat and sausage
- processed food

The result suggests that these items / item categories are almost always bought together by most customers.

Based on the above analyses, the following business recommendations are suggested:

1. Drinks, fresh products and fruits & vegetables are the most popular items at the store. 40% to 67% of customers will buy one or more of these items at any given visit to the store. Therefore, these items must be placed so that they are very accessible and can easily be seen by all customers who visit the store.
2. Also, since these items are already popular, there is no need for promotional discounts for them. Promotional discounts may be targeted at non-food items, snacks and candies, which are among the least popular products at the store.
3. About 15% to 20% of customers buy fresh products, fruit & vegetables, meat & sausage and processed food together. Therefore, these items should all be placed very close together in the store. They may be positioned either on the same shelf or in aisles that are in very close proximity. For the online store, advertisements of these items should be targeted at customers who buy some of the other items in the group. In the physical store, cashiers should prompt / recommend products in these groups to customers who buy atleast 2 other items in the groups.
4. Since these 4 item categories are often bought together, promotional discounts may be applied to just one of the four item sets at a time, instead of all four.