Market Basket Analysis for a French Retail Store data

Submitted by

Soumya Paul	211391
Soumita Bandyopadhyay	211390

Under the Guidance of

Dr. Amit Mitra

Department Of Mathematics And Statistics,

IIT Kanpur

Abstract

To promote business, a necessary part would be to show retailers how to locate related products together and how to cross-promote and recommend items that consumers often put in their shopping carts at the same time. Market basket analysis enables these sales and marketing teams to develop more effective product placement, pricing, and cross-selling strategies. AI and machine learning methods were used to analyze customers' buying habits. In this study, we analyze a french store dataset and find association between different item sets that customers bought together. The apriori algorithm has been used for generating frequent item-sets and generating association rules using the same.

Acknowledgment

It is our pleasure to present a project on "Market Basket Analysis for French Retail Store". Every accomplishment has constant encouragement and advice from valuable and noble minds to guide us in putting our efforts in the right direction to bring out the project. We want to express our sincere gratitude to our instructor **Dr. Amit Mitra** for her constant help and support throughout the completion of the project. Without her valuable guidance and motivation, it was nearly impossible to work on this project as a team and understand the practical aspect of the course "MTH 552AA: Statistical and AI Techniques in Data Mining". Also we are thankful to all faculty members and seniors without whose support at various stages, this project would not have materialized. Finally my earnest thanks go to my friends who were always beside me when I needed them without any excuses.

Soumya Paul

Soumita Bandyopadhyay

Contents

1 Introduction

Market basket analysis, simply put, is a data mining technique used by retailers to increase sales by better understanding customer purchasing habits. It involves analyzing large data sets, for example, purchase history, to find out products that are likely to be purchased together.

The digital records generated by POS systems made it easier for applications to process and analyze large volumes of purchase data as compared to handwritten records kept by the shop owners.

Shopping Basket Analysis tool in Microsoft Excel can be used to analyze transaction data contained in a spreadsheet and perform market basket analysis. The items to be analyzed must be linked with a transaction identification number.

1.1 How does market basket analysis work?

The main aim of Market Basket Analysis is

- to list items that are frequently purchased together,
- represent relationships in terms of association rules,

for example, if a customer buys bread, then he is likely to buy milk as well.

Association rules are usually represented as:

{Bread} -> {Milk}

2 Some terminologies

2.1 Antecedent:

Items or 'item-sets' which are found in the data are antecedents. In the above example, "Bread" is the antecedent.

2.2 Consequent:

A consequent is an item or set of items which are found in association with the antecedent. In the above example, "Milk" is the consequent.

2.3 Support Count:

The number of occurrences of an item-set in the database denoted by σ ({item-set}).

2.4 Support:

Fraction of transactions containing the item-set. It is given by

$$S\left(\text{item-set}\right) = \frac{\sigma\left(\{\text{item-set}\}\right)}{|T|}$$

2.4.1 Drawback

Table 1: Transaction data from a grocery store

Transactions	Items purchased
1	Milk, orange juice, ice cream, brandy, soap
2	Milk, ice cream, brandy
3	Milk, orange juice, detergent
4	Milk, ice cream, pizza, soap
5	Milk, orange juice, soap

Support presents a major drawback when assessing the quality of an association rule. For Table 1 the association rule "if brandy then milk" has support of 40%. However, is the association rule "if brandy then milk" an interesting rule? The answer is yes if this means 40% of customers buy brandy and milk together and nobody buys milk without having bought brandy. However, Table 1 shows that all the transactions contain milk. All customers buy milk and only 40% of those buy brandy. Hence, the association rule "if brandy then milk" is not interesting even if its support is 40%.

2.5 Frequent item-set:

An item-set whose support $\geq$ a predefined threshold value, say *min* sup is called a frequent item-set.

2.6 Confidence:

Let's say our rule is given by $A \implies B$, then confidence is the measure which is used to find how often B appears in transactions containing A, i.e.,

$$\begin{aligned} C\left(A \implies B\right) &= \frac{S\left(A, B\right)}{S\left(A\right)} \\ &= \frac{\frac{\sigma(A,B)}{|T|}}{\frac{\sigma(A)}{|T|}} \\ &= \frac{\sigma\left(A, B\right)}{\sigma\left(A\right)} \end{aligned}$$

2.6.1 Drawback

Confidence is definitely a good criterion to choose interesting rules, but is not a perfect criterion. For Table 1 consider a rule “if ice cream then soap” Its confidence or $\frac{S(\{icecream\},\{soap\})}{S(\{icecream\})}$ is 20% so you may think it is an interesting rule. However, if there is 60% chance (e.g., $S(\{soap\}) = 60\%$) that a randomly chosen transaction contains soap. Hence, ice-cream is not a powerful antecedent for identifying an soap purchase – it has lower than a random chance of identifying an soap purchase. Thus there is no cross-selling opportunity.

2.7 Lift

Lift is a step towards overcoming problems with support and confidence. Consider an association rule “if A then B.” The lift for the rule is defined as

$$\frac{C\left(A \implies B\right)}{S(B)}$$

or

$$\frac{S\left(A,B\right)}{S\left(A\right)S\left(B\right)}$$

.

As shown in the formula, lift is **symmetric** in that the lift for “if A then B” is the same as the lift for “if B then A.”

$S\left(B\right)$ is kind of probability that a randomly chosen transaction contains item B. That is to say, it is an unconditional probability of purchasing the item B regardless of other items purchased. Practitioners often refer to the term "expected confidence" for $S\left(B\right)$ rather than unconditional probability.

Hence, lift is said to measure the difference – measured in ratio – between the confidence of a rule and the expected confidence. For instance, the lift of an association rule “if ice cream then brandy” is 1.67 because the expected confidence is 40% and the confidence is 67%. This means that consumers purchasing ice cream are 1.67 times more likely to buy brandy than randomly selected consumers. In other words, a larger lift means more attractive rules.

2.8 Coverage

Fraction of baskets that have all the items in the Consequent for a particular rule.

3 Association Rule Mining

For any given set of transactions, our aim is to find set of rules such that

1. support $\geq min$ sup
2. confidence $\geq min$ conf, where min conf is the predefined minimum confidence threshold value.

Here (min sup, min conf) depends on how much risk one wants to take.

3.1 Brute Force Method

The Brute Force Method includes listing of all possible association rules and computing support and confidence for each rule. Then we prune as per threshold. Clearly, this method is computationally prohibitive and hence we prefer another method - The Apriori Algorithm.

3.2 The Apriori Algorithm

A relatively faster algorithm was introduced in "Fast Algorithms for Mining Association Rules"by **Agrawal and Srikant.**

3.2.1 2-Step ARM of apriori algorithm

Step-1: Generate all frequent item-sets (with support $\geq min$ sup).

Now, every subset of a frequent item-set is frequent (anti monotone property). Also, if an item-set is not frequent, then every super-set of that set will be infrequent. Prune all such item-sets as there is no need to explore such item-sets for ARM.

Step-2: Generate association rules using these frequent item-sets.

3.2.2 Apriori Step-1 in psuedocodes

- $k = 1$
- Generate frequent item-set of length 1
- Pruning of item-sets of higher orders (i.e. supersets), if necessary.
- Generate item-sets of length $k + 1$ from only the frequent item-sets of length k.
- Compute the support of new candidates. Prune if necessary.
- $k = k + 1$
- Repeat until no frequent item-sets can be found.

3.2.3 Generation (step) of item sets for the next level

Let L_k denote the frequent item sets at level k and c_k denote the set of all candidates at level k. Items in $L_{k-1}$are listed in an order.

Step-1: Self joining $L_{k-1} * L_{k-1}$i.e. joining of 2 items from L_{k-1}.

$$\{p.item_1, p.item_2, ..., p.item_{k-1}\}$$

$$and$$

$$\{q.item_1, q.item_2, ..., q.item_{k-1}\}$$

under the given order insert into c_k, the item-set, $p.item_1, p.item_2, ..., p.item_{k-2}, p.item_{k-1}, q.item_{k-1}$.

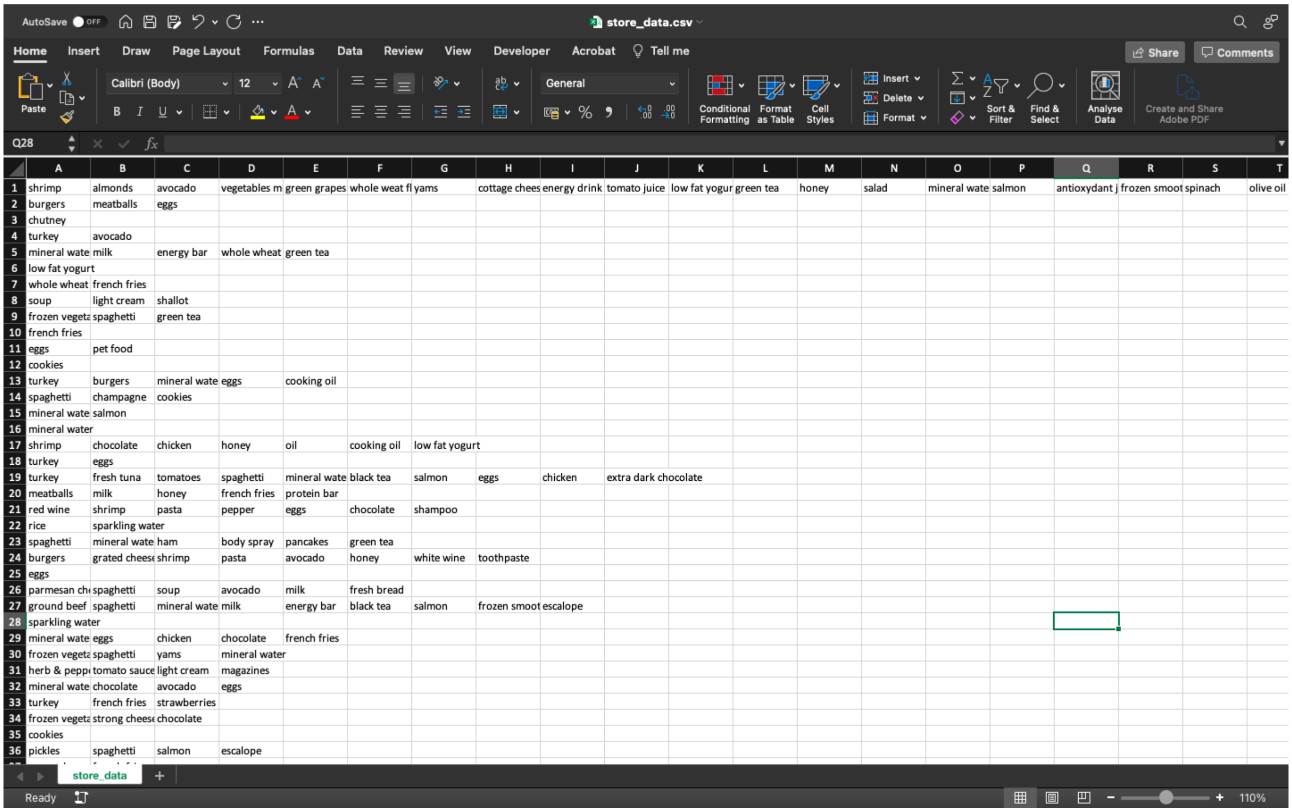
Step-2: Pruning of c_k set. $\forall$ item-sets c in c_k, and $\forall$ $(k-1)$ subsets S of c, (under the given order), if S is not in L_{k-1}, delete c from c_k.

3.2.4 Step 2 of apriori algorithm

Generate association rules using frequent item-sets. Find all non empty subsets F of L and output each rule $F \implies \{L - F\}$that satisfies the threshold on confidence (min conf)

4 Dataset Exploration

We have fetched our Market Basket Dataset from Github. At a glance we can see that the carts/transactions are represented as rows and the items are distributed in the columns for each transaction. Different products are given in 7501 transactions over the course of a week at a French retail store.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	shrimp	almonds	avocado	vegetables m	green grapes	whole weat fl	yams	cottage chees	energy drink	tomato juice	low fat yogur	green tea	honey	salad	mineral wate	salmon	antioxydant j	frozen smoot	spinach	olive oil
2	burgers	meatballs	eggs																	
3	chutney																			
4	turkey	avocado																		
5	mineral wate	milk	energy bar	whole wheat	green tea															
6	low fat yogurt																			
7	whole wheat	french fries																		
8	soup	light cream	shallot																	
9	frozen vegeta	spaghetti	green tea																	
10	french fries																			
11	eggs	pet food																		
12	cookies																			
13	turkey	burgers	mineral wate	eggs	cooking oil															
14	spaghetti	champagne	cookies																	
15	mineral wate	salmon																		
16	mineral water																			
17	shrimp	chocolate	chicken	honey	oil	cooking oil	low fat yogurt													
18	turkey	eggs																		
19	turkey	fresh tuna	tomatoes	spaghetti	mineral wate	black tea	salmon	eggs	chicken	extra dark chocolate										
20	meatballs	milk	honey	french fries	protein bar															
21	red wine	shrimp	pasta	pepper	eggs	chocolate	shampoo													
22	rice	sparkling water																		
23	spaghetti	mineral wate	ham	body spray	pancakes	green tea														
24	burgers	grated chees	shrimp	pasta	avocado	honey	white wine	toothpaste												
25	eggs																			
26	parmesan ch	spaghetti	soup	avocado	milk	fresh bread														
27	ground beef	spaghetti	mineral wate	milk	energy bar	black tea	salmon	frozen smoot	escalope											
28	sparkling water																			
29	mineral wate	eggs	chicken	chocolate	french fries															
30	frozen vegeta	spaghetti	yams	mineral water																
31	herb & pepp	tomato sauce	light cream	magazines																
32	mineral wate	chocolate	avocado	eggs																
33	turkey	french fries	strawberries																	
34	frozen vegeta	strong chees	chocolate																	
35	cookies																			
36	pickles	spaghetti	salmon	escalope																

Figure 1: Dataset

4.1 Dimension of the data

```r
library(arules) #Provides the infrastructure for representing, manipulating and analyzing transaction
#data and patterns (frequent item-sets and association rules). Also provides C implementations of
#the association mining algorithms Apriori and Eclat. Hahsler, Gruen and Hornik (2005)
library(arulesViz) #Extends package 'arules' with various visualization techniques for association
#rules and item-sets. The package also includes several interactive visualizations for rule exploration.
#Michael Hahsler (2017)
mba_data<-read.transactions("/Volumes/NO NAME/MBA/store_data.csv",format="basket",sep=",",cols=NULL)
dim(mba_data) #

[1] 7501  119
```

So, there are 7501 transaction and the maximum number of items purchased is 119.

4.2 Head of the Dataset

First six transactions are given as

```
inspect(head(mba_data))
```

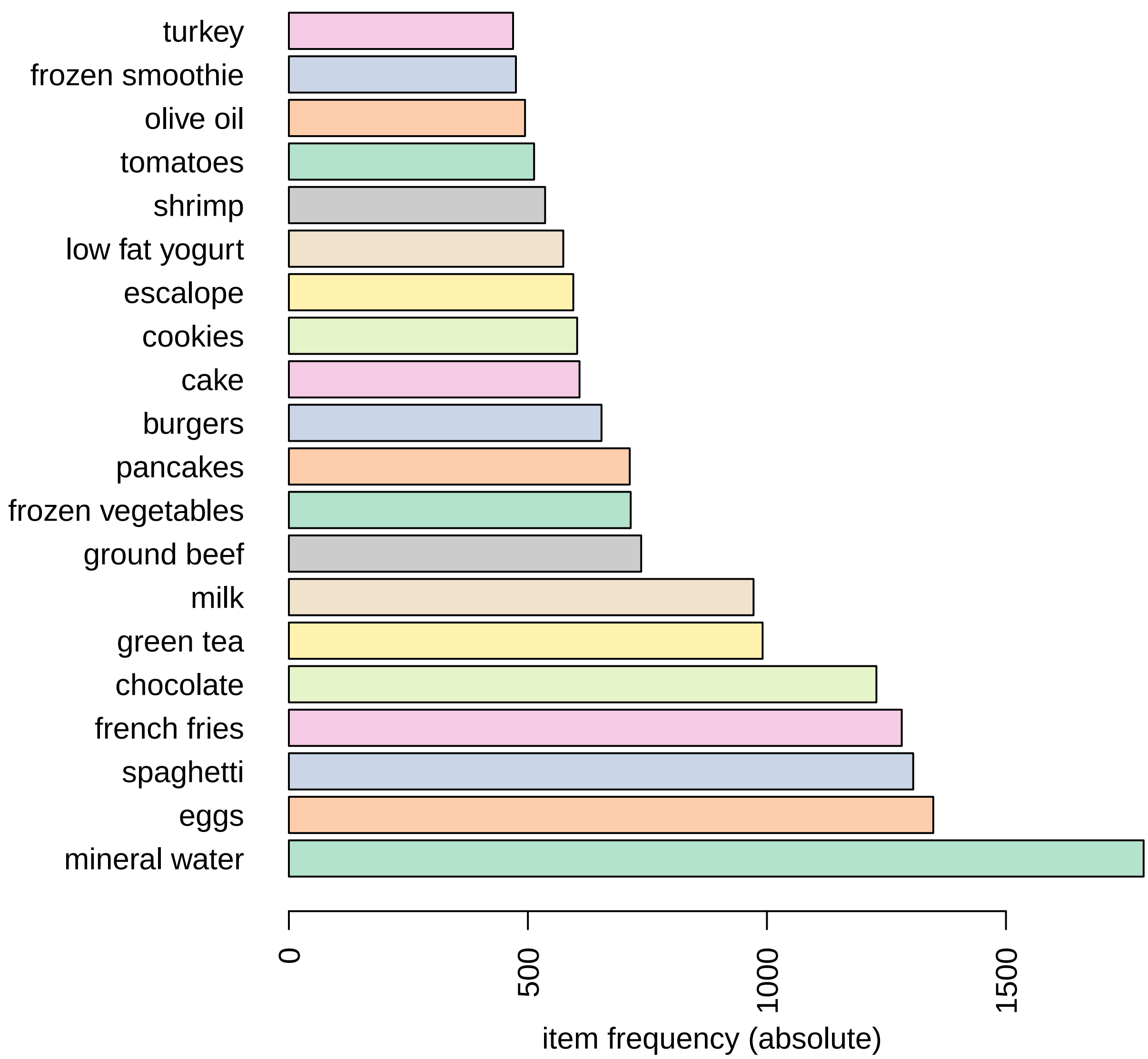
```
    items
[1] {almonds,
     antioxydant juice,
     avocado,
     cottage cheese,
     energy drink,
     frozen smoothie,
     green grapes,
     green tea,
     honey,
     low fat yogurt,
     mineral water,
     olive oil,
     salad,
     salmon,
     shrimp,
     spinach,
     tomato juice,
     vegetables mix,
     whole weat flour,
     yams}
[2] {burgers,
     eggs,
     meatballs}
[3] {chutney}
[4] {avocado,
     turkey}
[5] {energy bar,
     green tea,
     milk,
     mineral water,
     whole wheat rice}
```

```
[6] {low fat yogurt}
```

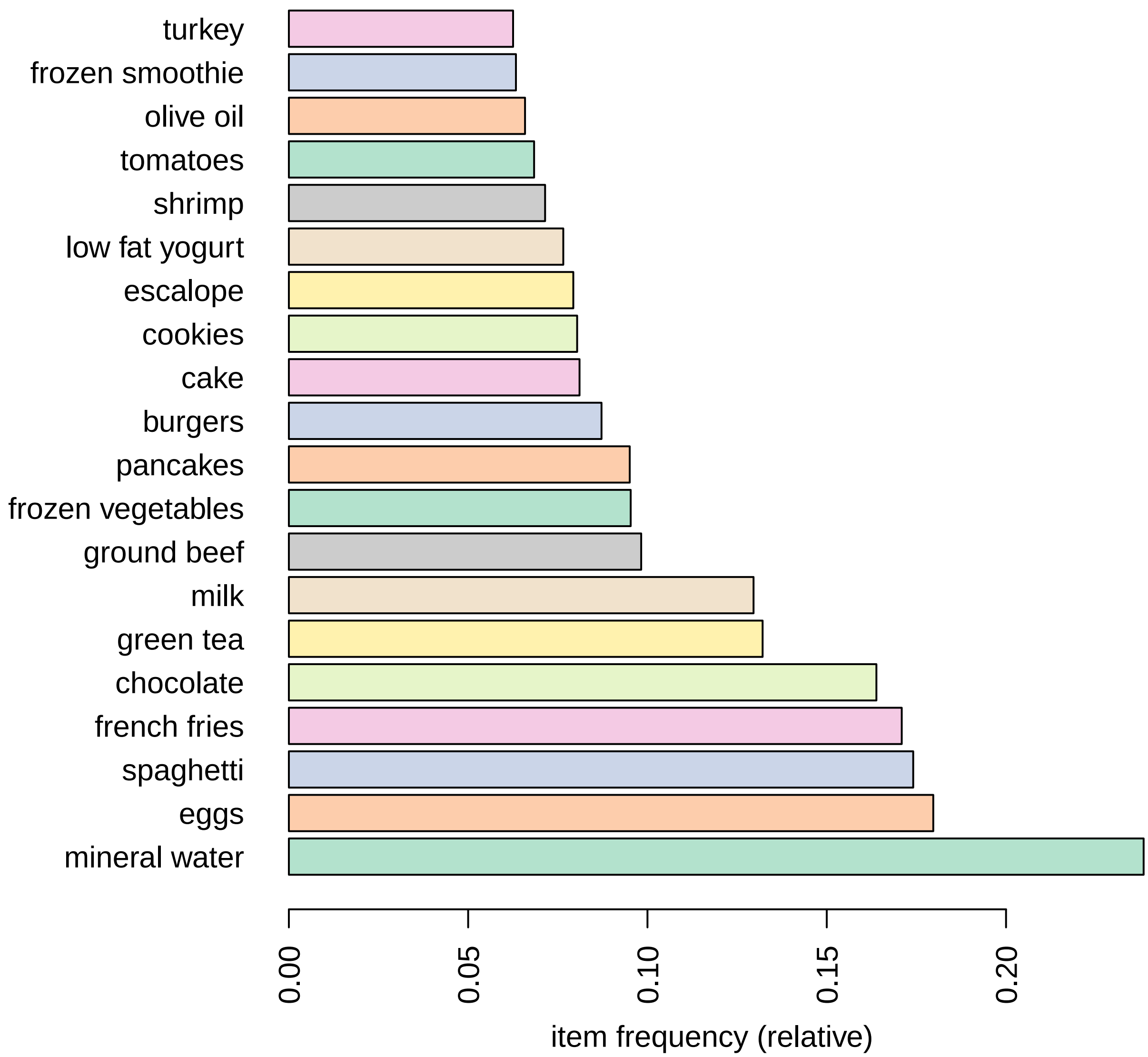
4.3 Most frequent item-sets

We can visualize the top 20 item-sets which are frequent by a horizontal frequency plot in both the "absolute" and "relative" scale.

```
library(RColorBrewer)
itemFrequencyPlot(mba_data,topN=20,type="absolute",horiz=TRUE,col = brewer.pal(8, 'Pastel2'))
```



```
itemFrequencyPlot(mba_data,topN=20,type="relative",horiz=TRUE,col = brewer.pal(8, 'Pastel2'))
```



So, Mineral water is the most frequent item.

4.4 Supports of different item-sets

```
head(itemFrequency((mba_data))) #first 6 supports alphabetically

          almonds antioxydant juice          asparagus           avocado
```

```
      0.020397280       0.008932142       0.004799360       0.033328889
      babies food             bacon
      0.004532729       0.008665511
itemFrequency(mba_data)[67] #support of the 67th item alphabetically
mashed potato
  0.004132782
itemFrequency(mba_data)["mineral water"] #support of mineral water
mineral water
    0.2383682
support(itemsets(list(c("mineral water","salad")),itemLabels=mba_data),mba_data) #support
[1] 0.00119984
#of {mineral water,salad}
```

So,

- almond, antioxidant juice, asparagus, avocado, babies food and bacon have supports 0.020397280, 0.008932142, 0.004799360, 0.033328889, 0.004532729 and 0.008665511 respectively.
- Support of the 67th item alphabetically, which is meshed potato, is 0.004132782.
- Support of mineral water is 0.2383682
- Support of item-set {mineral water, salad} is 0.00119984

5 Data Analysis

5.1 Apriori Algorithm

Let's see association rules with minimum support = 0.001 and minimum confidence level = 0.8 and plot the rules.

```
#Apriori Algorithm
data_rules<-apriori(mba_data,parameter= list(supp=0.001,conf=0.8))
Apriori
```

```
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5   0.001      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: 7

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[119 item(s), 7501 transaction(s)] done [0.00s].
sorting and recoding items ... [116 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.01s].
writing ... [74 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

inspect(head(data_rules))

    lhs                              rhs             support     confidence
[1] {frozen smoothie, spinach}    => {mineral water} 0.001066524 0.8888889
[2] {bacon, pancakes}             => {spaghetti}     0.001733102 0.8125000
[3] {nonfat milk, turkey}         => {mineral water} 0.001199840 0.8181818
[4] {ground beef, nonfat milk}    => {mineral water} 0.001599787 0.8571429
[5] {mushroom cream sauce, pasta} => {escalope}      0.002532996 0.9500000
[6] {milk, pasta}                 => {shrimp}        0.001599787 0.8571429
    coverage    lift      count
[1] 0.001199840  3.729058  8
[2] 0.002133049  4.666587 13
[3] 0.001466471  3.432428  9
[4] 0.001866418  3.595877 12
[5] 0.002666311 11.976387 19
```

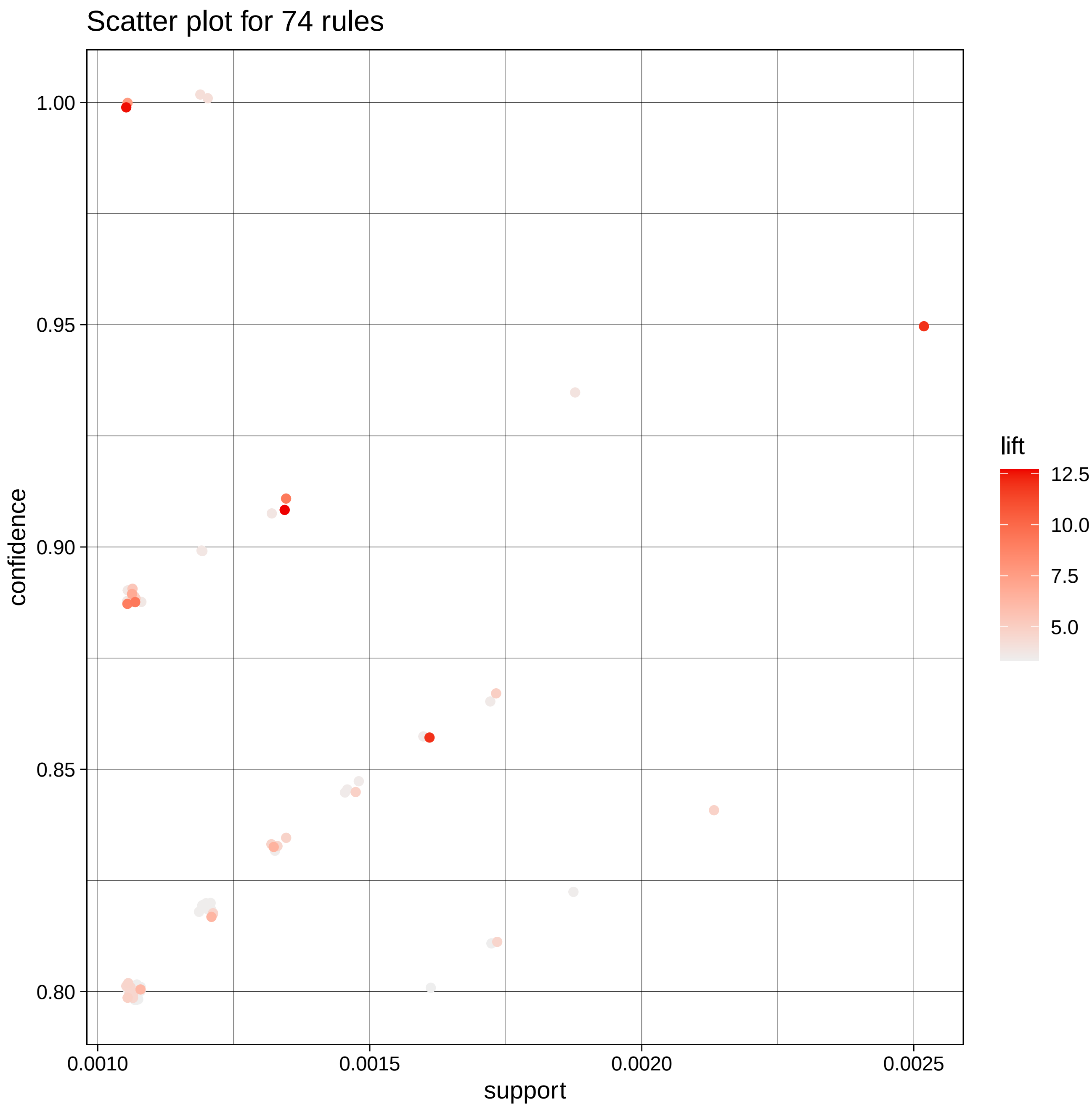
```
[6] 0.001866418 11.995203 12
```

So, the conclusions are

- There are 8 baskets which satisfy the {frozen smoothie, spinach} -> {mineral water}
- 0.1066524 % of baskets satisfy the {frozen smoothie, spinach} -> {mineral water}
- In 88.88889% of the cases {mineral water} appears in transactions containing {frozen smoothie, spinach}
- 0.001199840 is the fraction of baskets that have the items {frozen smoothie, spinach}
- Consumers purchasing {frozen smoothie, spinach} are 3.729058 times more likely to buy {mineral water} than randomly selected consumers
- We can see that the association rule {mushroom cream sauce, pasta} -> {escalope} has the maximum confidence as well as support

5.1.1 Scatter plot

A straight-forward visualization of association rules is to use a scatter plot with two interest measures on the axes. Here we plot support in the x-axis and confidence in the y-axis.



Conclusion: We can say that there is one rule whose support and confidence is moderately high. So, that rule is more most interesting than any other rules as it also possesses high lift value. The rule is {mashroom cream sauce, pasta} $\implies$ {escalope}. We can see it from interactive plot.

5.1.2 Grouped matrix -Based visualization

Let us denote the set of association rules as

$$\mathcal{R} = \{(A_1, C_1, \theta_1), (A_2, C_2, \theta_2), \ldots, (A_n, C_n, \theta_n)\}$$

where A_i's are the antecedents, C_i's are their corresponding consequents and θ_i's are their corresponding selected interest measure.

Let's start with the matrix M, which contains the values of a selected interest measure of the rules in set $\mathcal{R}$. The columns and rows of matrix M are the unique antecedents and consequents in $\mathcal{R}$, respectively. Now grouping rules becomes the problem of grouping columns or rows in M. Since for most applications the consequents in mined rules are restricted to a single item there is no problem with combinatorial explosion and we can restrict ourself to only grouping antecedents (i.e., columns in M). However, note that the same grouping method can be used also for consequents. We have used k-means clustering for this.

We use the interest measure lift, but other interest measures can be used as well. The idea behind lift is that antecedents that are statistically dependent on the same consequents (i.e., have a high lift value) are similar and thus should be grouped together.

To visualize we use a balloon plot with antecedent groups as columns and consequents as rows. The color of the balloons represent the aggregated interest measure in the group with a certain consequent and the size of the balloon shows the aggregated support. The default aggregation function is the median value in the group. The number of antecedents and the most important (frequent) items in the group are displayed as the labels for the columns. Furthermore, the columns and rows in the plot are reordered such that the aggregated interest measure is decreasing from top down and from left to right, placing the most interesting group in the top left corner.

```
plot(data_rules, method="grouped")
```

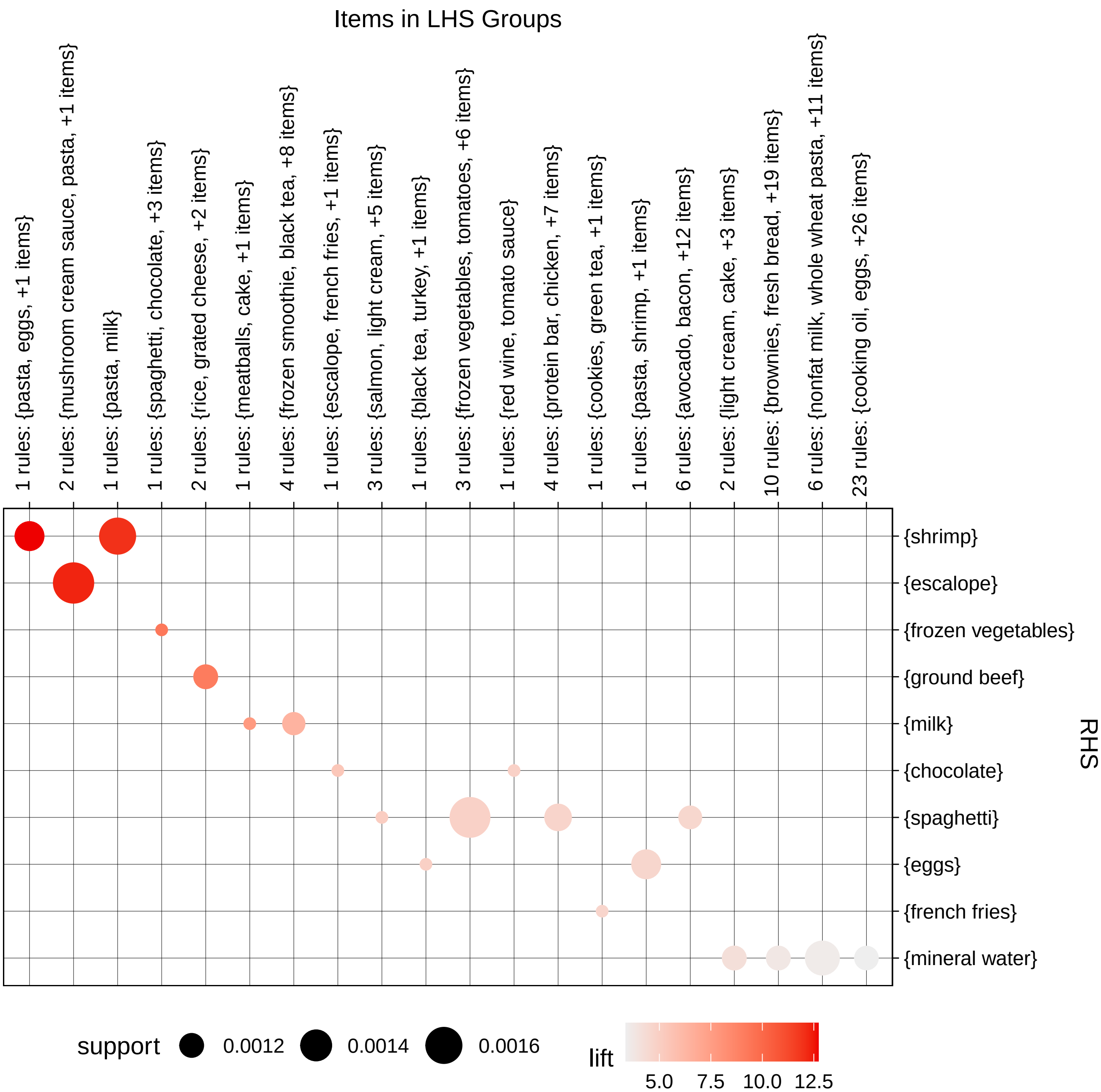


Figure 2: Group matrix-based visualization

Conclusion : The group of most interesting rules according to lift (the default measure) are shown in the top-left corner of the plot. There are 1 rule which contain "pasta, egg" and 1 other item in the antecedent and the consequent is "shrimp."

5.1.3 Sorting by quality measures

```
## ------Sorting by quality measure ----------##
#Sorting by lift values
inspect(sort(data_rules, by="lift", decreasing=T)[1:4])

    lhs                         rhs            support confidence    coverage     lift count
[1] {eggs,
     mineral water,
     pasta}                  => {shrimp}   0.001333156  0.9090909 0.001466471 12.72218    10
[2] {french fries,
     mushroom cream sauce,
     pasta}                  => {escalope} 0.001066524  1.0000000 0.001066524 12.60672     8
[3] {milk,
     pasta}                  => {shrimp}   0.001599787  0.8571429 0.001866418 11.99520    12
[4] {mushroom cream sauce,
     pasta}                  => {escalope} 0.002532996  0.9500000 0.002666311 11.97639    19

#Sorting by support values
inspect(sort(data_rules, by="support", decreasing=TRUE)[1:4])

    lhs                         rhs                 support confidence    coverage      lift count
[1] {mushroom cream sauce,
     pasta}                  => {escalope}      0.002532996  0.9500000 0.002666311 11.976387    19
[2] {frozen vegetables,
     olive oil,
     tomatoes}               => {spaghetti}     0.002133049  0.8421053 0.002532996  4.836624    16
[3] {red wine,
     soup}                   => {mineral water} 0.001866418  0.9333333 0.001999733  3.915511    14
[4] {frozen vegetables,
     olive oil,
     shrimp}                 => {mineral water} 0.001866418  0.8235294 0.002266364  3.454862    14

#Sorting by confidence values
inspect(sort(data_rules, by="confidence", decreasing=TRUE)[1:4])

    lhs                         rhs                 support confidence    coverage      lift count
```

```
[1] {french fries,
     mushroom cream sauce,
     pasta}                 => {escalope}      0.001066524          1 0.001066524 12.606723      8
[2] {ground beef,
     light cream,
     olive oil}             => {mineral water} 0.001199840          1 0.001199840  4.195190      9
[3] {cake,
     meatballs,
     mineral water}         => {milk}          0.001066524          1 0.001066524  7.717078      8
[4] {cake,
     olive oil,
     shrimp}                => {mineral water} 0.001199840          1 0.001199840  4.195190      9
```

Conclusion:

1. The rule {eggs, mineral water, pasta} $\implies$ {shrimp} has the highest lift value of 12.72218 which implies that consumers purchasing eggs, mineral water and pasta together are 12.72218 times more likely to buy shrimp than randomly selected consumers.

2. The rule {mushroom cream sauce, pasta} $\implies$ {escalope} has the highest support value of 0.002532996 which implies that the itemset {mushroom cream sauce, pasta, escalope} has about 0.2532% chance of occuring in the whole transaction set.

3. The rule {french fries, mushroom cream sauce,pasta} => {escalope} has the confidence value of 1 which implies that escalope always appears in transactions containing french fries, mushroom cream sauce and pasta together.

5.1.4 Graph-Based visualization

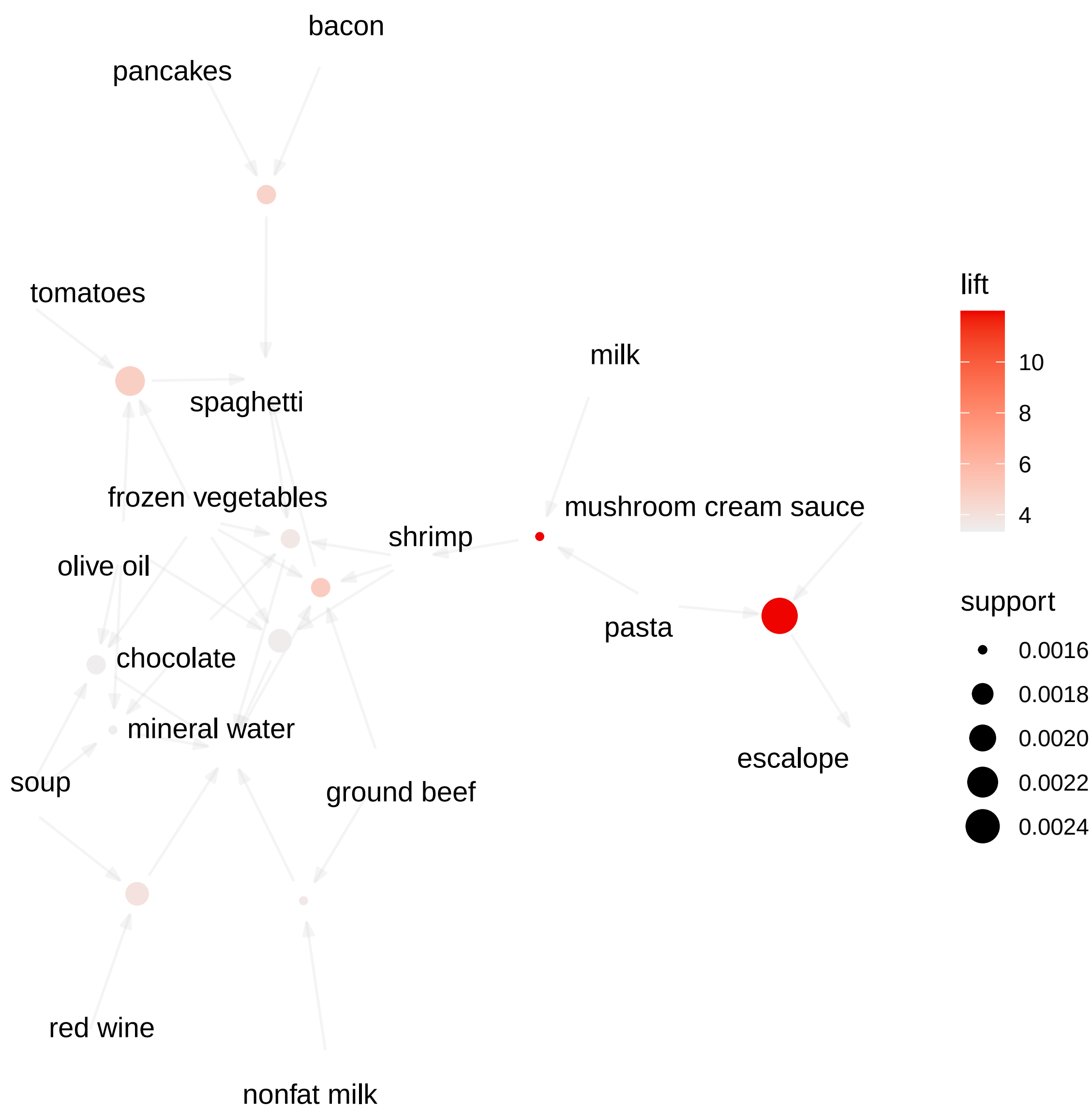
Graph-based techniques visualize association rules using vertices and edges where vertices annotated with item labels represent items, and item-sets or rules are represented as a second set of vertices. Items are linked to groups of items/rules using arrows. For rules arrows pointing from items to rule vertices indicate LHS items and an arrow from a rule to an item indicates the RHS. Measurements of interest are typically added to the graph using the color or size of the vertices representing items-sets/rules. This kind of visualization offers a very clear representation of rules but they tend to easily become cluttered and thus are only viable for very small sets of rules.

So, here we are increasing our minimum support value a little bit and set it as 0.0015 keeping the confidence level as earlier so that we can have less number of rules which leads to clean visualization.

```
#Apriori Algorithm
data_rules_simple<-apriori(mba_data,parameter= list(supp=0.0015,conf=0.8))

Apriori


Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5  0.0015      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: 11


set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[119 item(s), 7501 transaction(s)] done [0.00s].
sorting and recoding items ... [116 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [11 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].

#Asociation rules with various measures
inspect(data_rules_simple)

     lhs                           rhs                   support confidence    coverage      lift count
[1]  {bacon,
      pancakes}               => {spaghetti}     0.001733102  0.8125000 0.002133049  4.666587    13
[2]  {ground beef,
      nonfat milk}            => {mineral water} 0.001599787  0.8571429 0.001866418  3.595877    12
[3]  {mushroom cream sauce,
      pasta}                  => {escalope}      0.002532996  0.9500000 0.002666311 11.976387    19
[4]  {milk,
```

```
      pasta}                  => {shrimp}        0.001599787  0.8571429 0.001866418 11.995203     12
[5]  {red wine,
      soup}                   => {mineral water} 0.001866418  0.9333333 0.001999733  3.915511     14
[6]  {frozen vegetables,
      olive oil,
      soup}                   => {mineral water} 0.001733102  0.8125000 0.002133049  3.408592     13
[7]  {chocolate,
      olive oil,
      soup}                   => {mineral water} 0.001599787  0.8000000 0.001999733  3.356152     12
[8]  {frozen vegetables,
      olive oil,
      tomatoes}               => {spaghetti}     0.002133049  0.8421053 0.002532996  4.836624     16
[9]  {frozen vegetables,
      olive oil,
      shrimp}                 => {mineral water} 0.001866418  0.8235294 0.002266364  3.454862     14
[10] {frozen vegetables,
      ground beef,
      mineral water,
      shrimp}                 => {spaghetti}     0.001733102  0.8666667 0.001999733  4.977693     13
[11] {chocolate,
      frozen vegetables,
      shrimp,
      spaghetti}              => {mineral water} 0.001733102  0.8666667 0.001999733  3.635831     13

#Graph based visualization
plot(data_rules_simple,method="graph",edgeCol="black",cex=0.7,alpha=1)

Available control parameters (with default values):
layout  =  stress
circular  =  FALSE
ggraphdots  =  NULL
edges  =  <environment>
nodes  =  <environment>
nodetext  =  <environment>
colors  =  c("#EE0000FF", "#EEEEEEFF")
```

```
engine  =  ggplot2
max  =  100
verbose  =  FALSE
```



Conclusion: From the above figure we can see that the rule {mushroom cream sauce, pasta} $\implies$ {escalope} has a moderately high lift of 11.976 . The lift being 11.976 implies that consumers purchasing mushroom cream sauce and pasta together are 11.976 times more likely to buy escalope than randomly selected consumers.

References

[1] R. C. Blattberg, B.-D. Kim, and S. A. Neslin, *Market Basket Analysis*. New York, NY: Springer New York, 2008, pp. 339–351. [Online]. Available: https://doi.org/10.1007/978-0-387-72579-6_13

[2] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '93. New York, NY, USA: Association for Computing Machinery, 1993, p. 207â216. [Online]. Available: https://doi.org/10.1145/170035.170072

[3] M. Hahsler and S. Chelluboina, "Visualizing association rules: Introduction to the r-extension package arulesviz," 02 2015.