Association rules

2022-07-31

With this dataset we are going to create association rules that will allow you to identify relationships between products in our transactions.

Loading our libraries.

```
library(arules)
```

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

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

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

loading & previewing data set

```
trans <- read.transactions('http://bit.ly/SupermarketDatasetII',sep = ",")
```

```
## Warning in asMethod(object): removing duplicated items in transactions
```

```
# The duplicates and null values have been removed in this dataset when we were loading the set.
```

```
# verify object's class
class(trans)
```

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

```
# inspecting the first 10 transactions
inspect(trans[1:10])
```

```
##      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}
## [7]  {french fries,
##       whole wheat pasta}
## [8]  {light cream,
##       shallot,
##       soup}
## [9]  {frozen vegetables,
##       green tea,
##       spaghetti}
## [10] {french fries}
```

```
# creating a data frame comprising of the individual items in the data set
# generating a summary of the transactions
items <- as.data.frame(itemLabels(trans))
colnames(items) <- "Item"
head(items, 10)
```

```
##                 Item
## 1            almonds
## 2  antioxydant juice
## 3          asparagus
## 4            avocado
## 5        babies food
## 6              bacon
## 7     barbecue sauce
## 8          black tea
## 9        blueberries
## 10        body spray
```

```
summary(trans)
```

```
## transactions as itemMatrix in sparse format with
##  7501 rows (elements/itemsets/transactions) and
##  119 columns (items) and a density of 0.03288973
##
## most frequent items:
## mineral water          eggs     spaghetti  french fries     chocolate
##          1788          1348          1306          1282          1229
##       (Other)
##         22405
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 1754 1358 1044  816  667  493  391  324  259  139  102   67   40   22   17    4
##   18   19   20
##    1    2    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.914   5.000  20.000
##
## includes extended item information - examples:
##               labels
## 1            almonds
## 2 antioxydant juice
## 3          asparagus
```

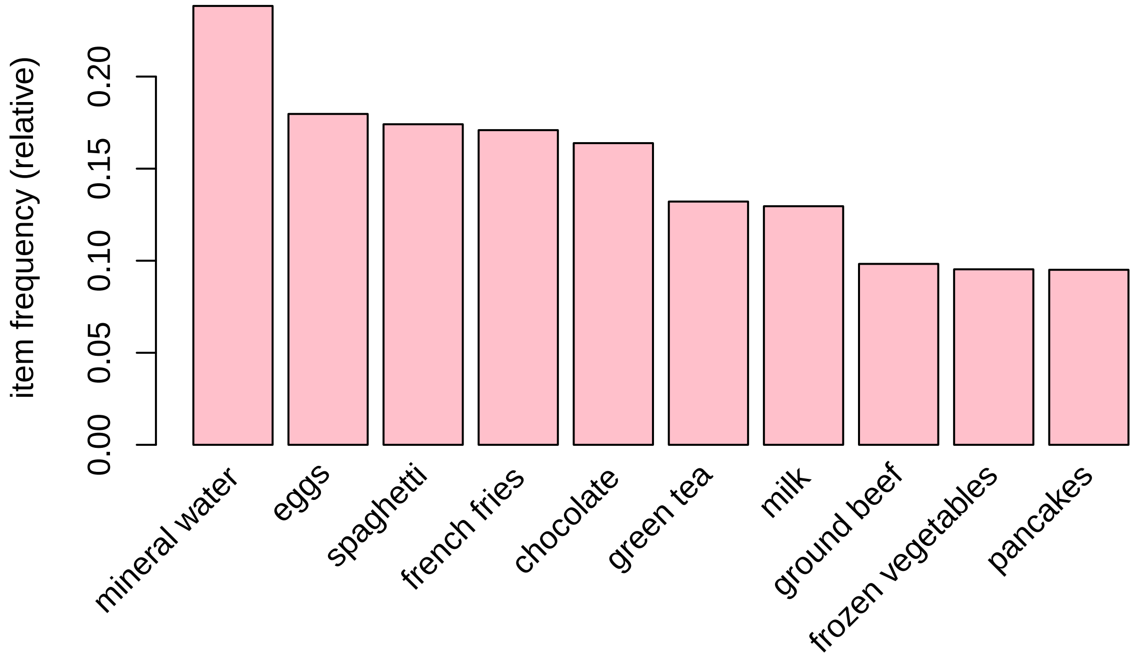
```
# most frequently occuring item in the transactions is mineral water.
```

```
# exploring the frequencies of transactions 20 to 30
itemFrequency(trans[, 20:30],type = "absolute")
```

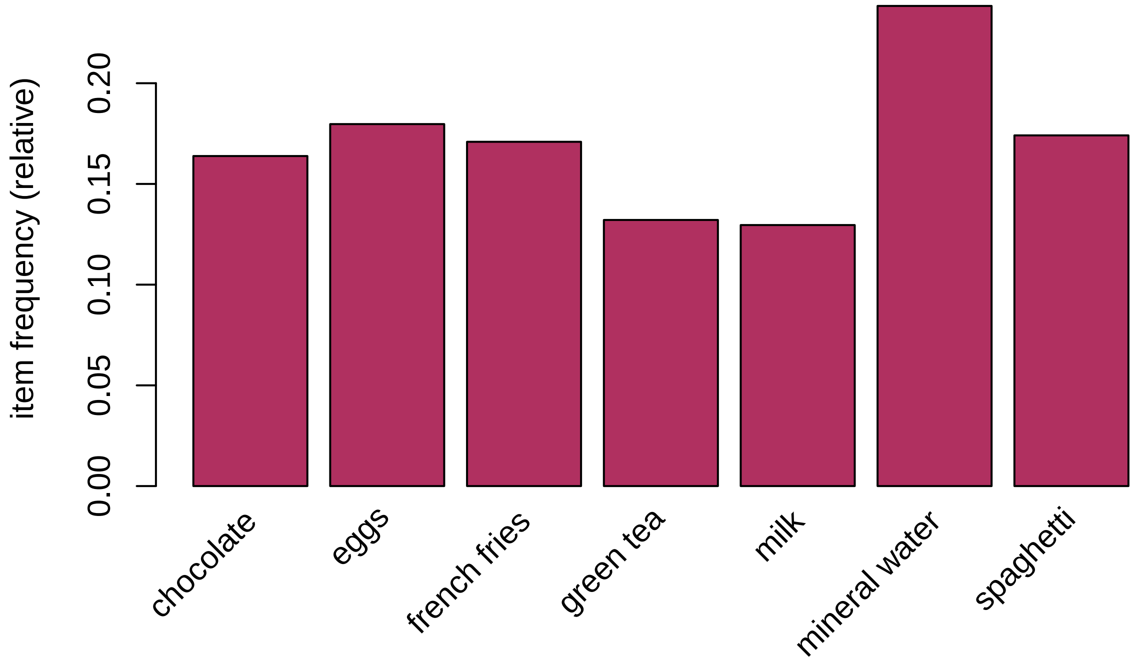
```
##         cauliflower             cereals           champagne             chicken
##                  36                 193                 351                 450
##               chili           chocolate     chocolate bread             chutney
##                  46                1229                  32                  31
##               cider clothes accessories             cookies
##                  79                  63                 603
```

```
par(mfrow = c(1, 2))
```

```
# plot the frequency of items
itemFrequencyPlot(trans, topN = 10,col="pink")
```



```
itemFrequencyPlot(trans, support = 0.1,col="maroon")
```



```
# The order remains the same from our summary: Mineral water to eggs to sphaghetti and so on
```

```
# model 1 using apriori function and min support 0.001 and confidence 0.9
rules1 <- apriori (trans, parameter = list(supp = 0.001, conf = 0.9))
```

```
## 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 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 ... [11 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules1
```

```
## set of 11 rules
```

```
# WE obtain a set of 11 rules.
```

```
# Inspect the top ten products
inspect(rules1[1:10])
```

```
##      lhs                           rhs                  support confidence    coverage      lift count
## [1]  {mushroom cream sauce,
##       pasta}                     => {escalope}      0.002532996  0.9500000 0.002666311 11.976387    19
## [2]  {red wine,
##       soup}                      => {mineral water} 0.001866418  0.9333333 0.001999733  3.915511    14
## [3]  {french fries,
##       mushroom cream sauce,
##       pasta}                     => {escalope}      0.001066524  1.0000000 0.001066524 12.606723     8
## [4]  {eggs,
##       mineral water,
##       pasta}                     => {shrimp}        0.001333156  0.9090909 0.001466471 12.722185    10
## [5]  {ground beef,
##       light cream,
##       olive oil}                 => {mineral water} 0.001199840  1.0000000 0.001199840  4.195190     9
## [6]  {cake,
##       meatballs,
##       mineral water}             => {milk}          0.001066524  1.0000000 0.001066524  7.717078     8
## [7]  {herb & pepper,
##       mineral water,
##       rice}                      => {ground beef}   0.001333156  0.9090909 0.001466471  9.252498    10
## [8]  {ground beef,
##       pancakes,
##       whole wheat rice}          => {mineral water} 0.001333156  0.9090909 0.001466471  3.813809    10
## [9]  {cake,
##       olive oil,
##       shrimp}                    => {mineral water} 0.001199840  1.0000000 0.001199840  4.195190     9
## [10] {frozen vegetables,
##       milk,
##       spaghetti,
##       turkey}                    => {mineral water} 0.001199840  0.9000000 0.001333156  3.775671     9
```

```
# This shows in the first basket, with mushroom sauce and pasta, the next top pick would be an escalope
# The second basket with red wine, the next top choice will be mineral water.
```

```
# Building a apriori model with Min Support as 0.002 and confidence as 0.9
rules2 <- apriori (trans,parameter = list(supp = 0.002, conf = 0.9))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5   0.002      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: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.00s].
## sorting and recoding items ... [115 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [1 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules2
```

```
## set of 1 rules
```

```
# We ontain a set of 1 rules
```

```
inspect(rules2[1])
```

```
##     lhs                              rhs        support     confidence
## [1] {mushroom cream sauce, pasta} => {escalope} 0.002532996 0.95
##     coverage    lift     count
## [1] 0.002666311 11.97639 19
```

```
# With one set of rule we have one basket of mushroom cream sauce and pasta top pick being an escalope
```

```
# model 3 with Min Support as 0.001 and confidence as 0.7.
rules3 <- apriori (trans, parameter = list(supp = 0.001, conf = 0.7))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    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 ... [200 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules3
```

```
## set of 200 rules
```

```
inspect(rules3[1:10])
```

```
##      lhs                                rhs             support     confidence
## [1]  {frozen smoothie, spinach}      => {mineral water} 0.001066524 0.8888889
## [2]  {spaghetti, spinach}            => {mineral water} 0.001333156 0.7142857
## [3]  {olive oil, strong cheese}      => {spaghetti}     0.001066524 0.7272727
## [4]  {milk, strong cheese}           => {mineral water} 0.001599787 0.7058824
## [5]  {green beans, ground beef}      => {spaghetti}     0.001066524 0.7272727
## [6]  {green grapes, salmon}          => {mineral water} 0.001066524 0.7272727
## [7]  {blueberries, pancakes}         => {mineral water} 0.001066524 0.7272727
## [8]  {blueberries, eggs}             => {mineral water} 0.001599787 0.7500000
## [9]  {ground beef, whole weat flour} => {mineral water} 0.001066524 0.7272727
## [10] {bacon, pancakes}               => {spaghetti}     0.001733102 0.8125000
##      coverage    lift     count
## [1]  0.001199840 3.729058  8
## [2]  0.001866418 2.996564 10
## [3]  0.001466471 4.177085  8
## [4]  0.002266364 2.961311 12
## [5]  0.001466471 4.177085  8
## [6]  0.001466471 3.051047  8
## [7]  0.001466471 3.051047  8
## [8]  0.002133049 3.146393 12
## [9]  0.001466471 3.051047  8
## [10] 0.002133049 4.666587 13
```

```
# With this model, the rules have gone up to 200,
# The top pick basket being the frozen smoothie basket with mineral water as next pick at 88% confidenc
```