Part 3: Association Rules

Ed

2022-06-11

```
# loading the library
library(arules)
```

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

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

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

```
# loading our data
data <- read.transactions('http://bit.ly/SupermarketDatasetII',sep = ",")
```

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

```
data
```

```
## transactions in sparse format with
##  7501 transactions (rows) and
##  119 items (columns)
```

```
# checking class of objects
class(data)
```

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

```
# creating a dataframeof individual items in the dataset
items <- as.data.frame(itemLabels(data))
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
```

```r
# summary of the transactions
summary(data)
```

```
## 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
```

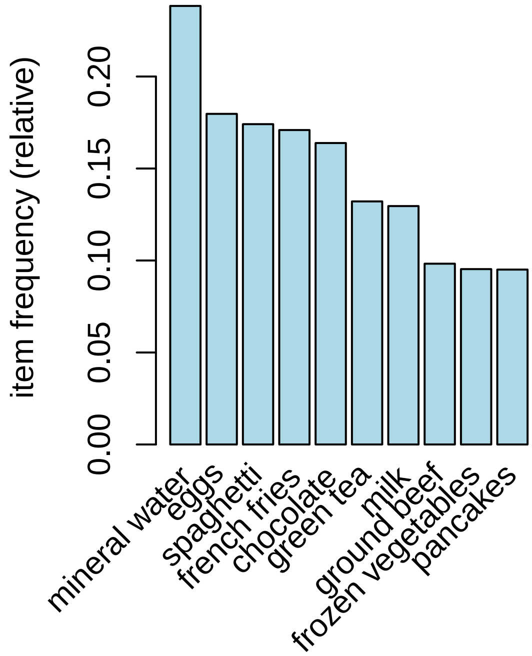
```r
# exploring the frequency of transactions 12 to 15
itemFrequency(data[, 12:15],type = "absolute")
```

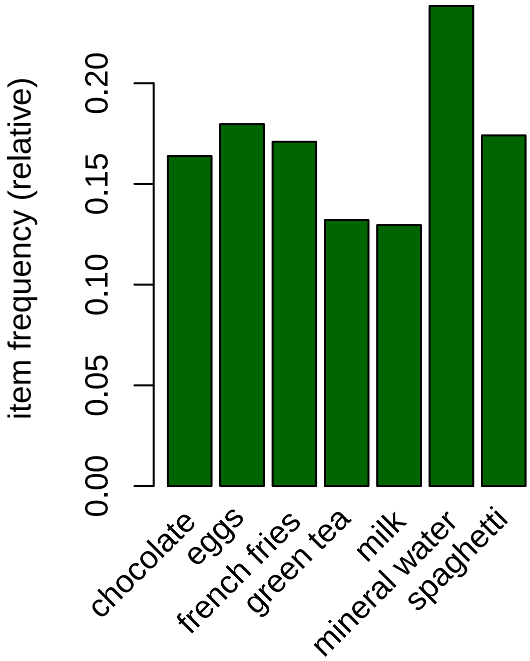
```
##     brownies    bug spray burger sauce      burgers
##          253           65           44          654
```

```r
round(itemFrequency(data[, 12:15],type = "relative")*100,2)
```

```
##     brownies    bug spray burger sauce      burgers
##         3.37         0.87         0.59         8.72
```

```r
# plotting the top 10 most common items and the frequencies
par(mfrow = c(1, 2))
itemFrequencyPlot(data, topN = 10,col="lightblue")
itemFrequencyPlot(data, support = 0.1,col="darkgreen")
```





```
# building a model with min support 0.001 and confidence 0.8
rules1 <- apriori (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].
```

A set of 74 rules

```r
# building model with Min Support as 0.002 and confidence as 0.6.
rules2 <- apriori (data, parameter = list(supp = 0.002, conf = 0.6))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    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 ... [43 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

A set of 43 rules

```r
# building model with Min Support as 0.002 and confidence as 0.6.
rules2 <- apriori (data, 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].
```

A set of 1 rule

```r
# previewing a summary of rule1
summary(rules1)
```

```
## set of 74 rules
##
## rule length distribution (lhs + rhs):sizes
##  3  4  5  6
## 15 42 16  1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   4.000   4.000   4.041   4.000   6.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001067   Min.   :0.8000   Min.   :0.001067   Min.   : 3.356
##  1st Qu.:0.001067   1st Qu.:0.8000   1st Qu.:0.001333   1st Qu.: 3.432
##  Median :0.001133   Median :0.8333   Median :0.001333   Median : 3.795
##  Mean   :0.001256   Mean   :0.8504   Mean   :0.001479   Mean   : 4.823
##  3rd Qu.:0.001333   3rd Qu.:0.8889   3rd Qu.:0.001600   3rd Qu.: 4.877
##  Max.   :0.002533   Max.   :1.0000   Max.   :0.002666   Max.   :12.722
##      count
##  Min.   : 8.000
##  1st Qu.: 8.000
##  Median : 8.500
##  Mean   : 9.419
##  3rd Qu.:10.000
##  Max.   :19.000
##
## mining info:
##  data ntransactions support confidence
##  data          7501   0.001        0.8
##                                                                     call
##  apriori(data = data, parameter = list(supp = 0.001, conf = 0.8))
```

```r
# inspecting the first 5 rules of our rule1 model
inspect(rules1[1:5])
```

```
##     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
##     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
```

If one buys frozen smoothie and spinach, there is a 88.8% chance they will buy mineral water. If someone buys bacon and pancakes, there is a 81.2% that they will buy spaghetti. If someone buys mushroom cream sauce and pasta, there is a 95% chance that they will buy escalope.

```r
# sorting our rules in decreasing order of confidence
rules1 <- sort(rules1, by = "confidence", decreasing = TRUE)
inspect(rules1[1:5])
```

```
##     lhs                        rhs                 support confidence    coverage      lift count
## [1] {french fries,                                                                                
##      mushroom cream sauce,                                                                        
##      pasta}                 => {escalope}      0.001066524       1.00 0.001066524 12.606723     8
## [2] {ground beef,                                                                                 
##      light cream,                                                                                 
##      olive oil}             => {mineral water} 0.001199840       1.00 0.001199840  4.195190     9
## [3] {cake,                                                                                        
##      meatballs,                                                                                   
##      mineral water}         => {milk}          0.001066524       1.00 0.001066524  7.717078     8
## [4] {cake,                                                                                        
##      olive oil,                                                                                   
##      shrimp}                => {mineral water} 0.001199840       1.00 0.001199840  4.195190     9
## [5] {mushroom cream sauce,                                                                        
##      pasta}                 => {escalope}      0.002532996       0.95 0.002666311 11.976387    19
```

Our rules have confidence of 1 thus; If one buys cake, meatballs and mineral water, there is a 100% chance that they will buy milk. If one buys french fries, mushroom cream sauce and pasta there is a 100% chance they will buy escalope. If one buys ground beef, light cream and olive oil or cake, olive oil and shrimp there is a 100% chance they will buy mineral water.