Association Rule in R

Karimi Gichunge

11/13/2020

Research Question: Identify relationships between variables

```
# Loading the arules library
#
library(arules)
```

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

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

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

```
#Loading the data in form of a transactions dataset
items <- read.transactions("C:/Users/Karimi/Downloads/items.csv", sep = ",")
```

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

```
head(items)
```

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

```
#Checking class of our dataset
class(items)
```

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

```
items.list<-as.data.frame(itemLabels(items))
colnames(items.list) <- "Item"
items.list
```

```
##                    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
## 11              bramble
## 12             brownies
## 13            bug spray
## 14         burger sauce
## 15              burgers
## 16               butter
## 17                 cake
## 18           candy bars
## 19              carrots
## 20          cauliflower
## 21              cereals
```

```
## 22             champagne
## 23               chicken
## 24                 chili
## 25             chocolate
## 26       chocolate bread
## 27               chutney
## 28                 cider
## 29   clothes accessories
## 30               cookies
## 31          cooking oil
## 32                 corn
## 33        cottage cheese
## 34                 cream
## 35          dessert wine
## 36              eggplant
## 37                  eggs
## 38            energy bar
## 39          energy drink
## 40              escalope
## 41  extra dark chocolate
## 42             flax seed
## 43          french fries
## 44           french wine
## 45           fresh bread
## 46            fresh tuna
## 47         fromage blanc
## 48       frozen smoothie
## 49     frozen vegetables
## 50       gluten free bar
## 51         grated cheese
## 52           green beans
## 53          green grapes
## 54             green tea
## 55           ground beef
## 56                 gums
## 57                  ham
## 58     hand protein bar
## 59         herb & pepper
## 60                honey
## 61              hot dogs
## 62               ketchup
## 63           light cream
## 64            light mayo
## 65        low fat yogurt
## 66             magazines
## 67         mashed potato
## 68            mayonnaise
## 69             meatballs
## 70                melons
## 71                  milk
## 72         mineral water
## 73                  mint
## 74        mint green tea
## 75               muffins
## 76  mushroom cream sauce
## 77               napkins
## 78           nonfat milk
## 79               oatmeal
## 80                   oil
## 81             olive oil
## 82              pancakes
## 83       parmesan cheese
## 84                 pasta
## 85                pepper
## 86              pet food
## 87               pickles
## 88           protein bar
## 89              red wine
## 90                  rice
## 91                 salad
## 92                salmon
## 93                  salt
## 94              sandwich
```

```
## 95               shallot
## 96               shampoo
## 97                shrimp
## 98                  soda
## 99                  soup
## 100             spaghetti
## 101      sparkling water
## 102              spinach
## 103         strawberries
## 104        strong cheese
## 105                  tea
## 106         tomato juice
## 107         tomato sauce
## 108             tomatoes
## 109           toothpaste
## 110               turkey
## 111       vegetables mix
## 112          water spray
## 113           white wine
## 114     whole weat flour
## 115    whole wheat pasta
## 116     whole wheat rice
## 117                 yams
## 118          yogurt cake
## 119             zucchini
```

There is a total of 119 items in the purchase list.

```
# Previewing our first 6 transactions
#
inspect(items[1:6])
```

```
##     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}
```

From above we see that our first transaction is very lengthy as compared to the rest, and transaction 6 only contains a single item.

```
# Checking which items were mostly purchased and distribution of item sets
summary(items)
```

```
## 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
```

Mineral water was the most purchased item, it was in 1708 transactions, followed by eggs then spaghetti. 1754 transactions consisted of only a sinle item in the basket which is the highest basket size, followed by 2 items all the way down to 1 transaction containing 18 and another 20 items.

```
#Viewing the frequency of items purchased
itemFrequency(items[, 1:10],type = "absolute")
```

```
##           almonds antioxydant juice         asparagus           avocado
##               153                67                36               250
##       babies food             bacon    barbecue sauce         black tea
##                34                65                81               107
##       blueberries        body spray
##                69                86
```

```
#Getting the support of the top ten items
round(itemFrequency(items[, 1:10],type = "relative")*100,2)
```

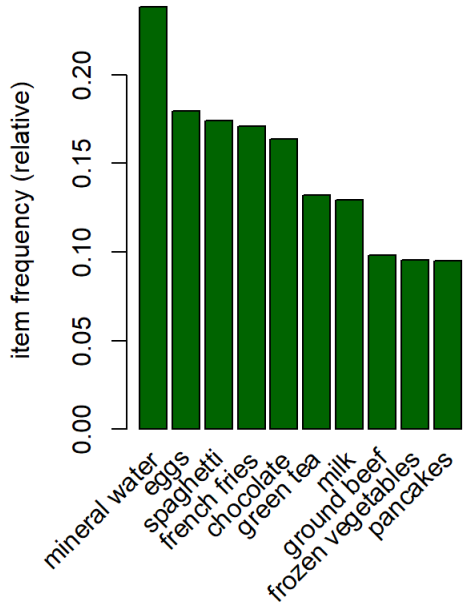
```
##           almonds antioxydant juice         asparagus           avocado
##              2.04              0.89              0.48              3.33
##       babies food             bacon    barbecue sauce         black tea
##              0.45              0.87              1.08              1.43
##       blueberries        body spray
##              0.92              1.15
```

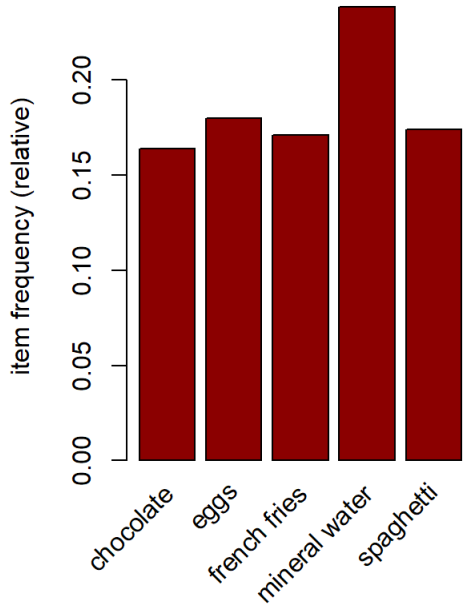
From the list above, it evident that for the first 10 items in the stores list(arranged alphabetically), a good number of purchases contained an avocado and almonds in the basket. Hence the high support percentage for both items.

```
# Displaying top 10 most frequent items
# and the items whose relative importance is at least 10%
#
par(mfrow = c(1, 2))

# plot the frequency of   top N items
itemFrequencyPlot(items, topN = 10,col="darkgreen")

# Setting the minsup for getting the frequent itemset
itemFrequencyPlot(items, support = 0.15,col="darkred")
```






```
# Building a model based on association rules
# using the apriori function
# ---
# We use Min Support as 0.001 and confidence as 0.8
# ---
#
rules <- apriori (items, parameter = list(supp = 0.001, conf = 0.85))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.85    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 [0.00s].
## writing ... [29 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules
```

```
## set of 29 rules
```

Increasing support to 0.002 decreases the set of rules to only 2 which is outrageous, so we eave it at 0.001 since it gives us 74rules when combined by a confidence of 0.8. But since we have 119 items, it also doesn't make much sense to have 74rules, so we increase the confidence to 0.85 which gives us 29rules.

```
#Summary of our model
summary(rules)
```

```
## set of 29 rules
##
## rule length distribution (lhs + rhs):sizes
##  3  4  5  6
##  5 19  4  1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   4.000   4.000   4.034   4.000   6.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001067   Min.   :0.8571   Min.   :0.001067   Min.   : 3.596
##  1st Qu.:0.001067   1st Qu.:0.8889   1st Qu.:0.001200   1st Qu.: 3.729
##  Median :0.001067   Median :0.8889   Median :0.001200   Median : 4.946
##  Mean   :0.001273   Mean   :0.9070   Mean   :0.001407   Mean   : 6.050
##  3rd Qu.:0.001333   3rd Qu.:0.9091   3rd Qu.:0.001466   3rd Qu.: 7.717
##  Max.   :0.002533   Max.   :1.0000   Max.   :0.002666   Max.   :12.722
##      count
##  Min.   : 8.000
##  1st Qu.: 8.000
##  Median : 8.000
##  Mean   : 9.552
##  3rd Qu.:10.000
##  Max.   :19.000
##
## mining info:
##   data ntransactions support confidence
##  items          7501   0.001       0.85
```

Our model shows that a basket size of 4 has the highest length at 19, a basket size of 3 with 5 transctions

```
#Previewing the basket combinations of our model
inspect(rules)
```

```
##      lhs                       rhs                     support confidence    coverage      lift count
## [1]  {frozen smoothie,
##       spinach}              => {mineral water}     0.001066524  0.8888889 0.001199840  3.729058     8
## [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]  {french fries,
##       mushroom cream sauce,
##       pasta}                => {escalope}          0.001066524  1.0000000 0.001066524 12.606723     8
## [7]  {eggs,
##       mineral water,
##       pasta}                => {shrimp}            0.001333156  0.9090909 0.001466471 12.722185    10
## [8]  {black tea,
##       spaghetti,
##       turkey}               => {eggs}              0.001066524  0.8888889 0.001199840  4.946258     8
## [9]  {ground beef,
##       light cream,
##       olive oil}            => {mineral water}     0.001199840  1.0000000 0.001199840  4.195190     9
## [10] {light cream,
##       mineral water,
##       shrimp}               => {spaghetti}         0.001066524  0.8888889 0.001199840  5.105326     8
## [11] {cake,
##       meatballs,
##       milk}                 => {mineral water}     0.001066524  0.8888889 0.001199840  3.729058     8
## [12] {cake,
##       meatballs,
##       mineral water}        => {milk}              0.001066524  1.0000000 0.001066524  7.717078     8
## [13] {herb & pepper,
##       mineral water,
##       rice}                 => {ground beef}       0.001333156  0.9090909 0.001466471  9.252498    10
## [14] {grated cheese,
##       mineral water,
```

```
##       rice}                 => {ground beef}       0.001066524  0.8888889 0.001199840  9.046887     8
## [15] {cake,
##       olive oil,
##       whole wheat pasta}    => {mineral water}     0.001066524  0.8888889 0.001199840  3.729058     8
## [16] {escalope,
##       hot dogs,
##       mineral water}        => {milk}              0.001066524  0.8888889 0.001199840  6.859625     8
## [17] {brownies,
##       eggs,
##       ground beef}          => {mineral water}     0.001066524  0.8888889 0.001199840  3.729058     8
## [18] {chicken,
##       fresh bread,
##       pancakes}             => {mineral water}     0.001066524  0.8888889 0.001199840  3.729058     8
## [19] {ground beef,
##       salmon,
##       shrimp}               => {spaghetti}         0.001066524  0.8888889 0.001199840  5.105326     8
## [20] {burgers,
##       milk,
##       salmon}               => {spaghetti}         0.001066524  0.8888889 0.001199840  5.105326     8
## [21] {chocolate,
##       soup,
##       turkey}               => {mineral water}     0.001066524  0.8888889 0.001199840  3.729058     8
## [22] {ground beef,
##       pancakes,
##       whole wheat rice}     => {mineral water}     0.001333156  0.9090909 0.001466471  3.813809    10
## [23] {escalope,
##       french fries,
##       shrimp}               => {chocolate}         0.001066524  0.8888889 0.001199840  5.425188     8
## [24] {cake,
##       olive oil,
##       shrimp}               => {mineral water}     0.001199840  1.0000000 0.001199840  4.195190     9
## [25] {frozen vegetables,
##       milk,
##       spaghetti,
##       turkey}               => {mineral water}     0.001199840  0.9000000 0.001333156  3.775671     9
## [26] {chocolate,
##       frozen vegetables,
##       olive oil,
##       shrimp}               => {mineral water}     0.001199840  0.9000000 0.001333156  3.775671     9
## [27] {frozen vegetables,
##       ground beef,
##       mineral water,
##       shrimp}               => {spaghetti}         0.001733102  0.8666667 0.001999733  4.977693    13
## [28] {chocolate,
##       frozen vegetables,
##       shrimp,
##       spaghetti}            => {mineral water}     0.001733102  0.8666667 0.001999733  3.635831    13
## [29] {chocolate,
##       ground beef,
##       milk,
##       mineral water,
##       spaghetti}            => {frozen vegetables} 0.001066524  0.8888889 0.001199840  9.325253     8
```

- If a person buys mushroom cream sauce and pasta, their is a 95% chance that they will buy escalope
- If a person purchases french fries,mushroom cream sauce and pasta then there is a 100% chance that they will buy escalope. * There is also a 100% chance that if a person buys cake, meatballs and mineral water they will also buy milk and so on.

```
rules<-sort(rules, by="lift", decreasing=TRUE)
inspect(rules[1:5])
```

```
##     lhs                       rhs                    support confidence    coverage      lift count
## [1] {eggs,
##      mineral water,
##      pasta}                => {shrimp}            0.001333156  0.9090909 0.001466471 12.722185    10
## [2] {french fries,
##      mushroom cream sauce,
##      pasta}                => {escalope}          0.001066524  1.0000000 0.001066524 12.606723     8
## [3] {milk,
##      pasta}                => {shrimp}            0.001599787  0.8571429 0.001866418 11.995203    12
## [4] {mushroom cream sauce,
##      pasta}                => {escalope}          0.002532996  0.9500000 0.002666311 11.976387    19
## [5] {chocolate,
##      ground beef,
##      milk,
##      mineral water,
##      spaghetti}            => {frozen vegetables} 0.001066524  0.8888889 0.001199840  9.325253     8
```

Conclusion The lift for the above items are very high hence a great correlation between the itemsets and the combination commodity. Hence Carrefour marketing team can make marketing plans using this. For example they can arrange iteems in an order that prioritizes this purchases. E.g they can have mushroom cream sauce, pasta and escalope on the same lane. They can have mineral water at the very front where it is easily accessible and so on.