Association Rules

RuthNguli

2022-04-01

Looading libraries

```
# calling libraries
#
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(arules)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

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

Reading data

```
# Loading and reading data
#
supa <- read.transactions("http://bit.ly/SupermarketDatasetII")
```

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

```
# previewing the class of the data
#
class(supa)
```

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

```
# previewing the first 5 items
#
inspect(head(supa, 5))
```

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

```
# previewing the structure of data
#
str(supa)
```

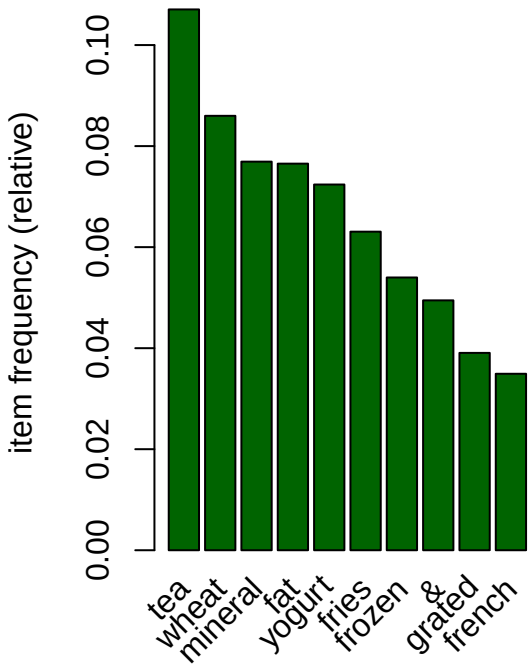
```
## Formal class 'transactions' [package "arules"] with 3 slots
##   ..@ data       :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
##   .. .. ..@ i       : int [1:23299] 1087 1614 1705 1732 1993 2101 2105 2358 2444 3463 ...
##   .. .. ..@ p       : int [1:7502] 0 15 16 17 18 24 27 31 33 36 ...
##   .. .. ..@ Dim     : int [1:2] 5729 7501
##   .. .. ..@ Dimnames:List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : NULL
##   .. .. ..@ factors : list()
##   ..@ itemInfo   :'data.frame':  5729 obs. of  1 variable:
##   .. ..$ labels: chr [1:5729] "&" "accessories" "accessories,antioxydant" "accessories,champagne,fres
##   ..@ itemsetInfo:'data.frame':  0 obs. of  0 variables
```

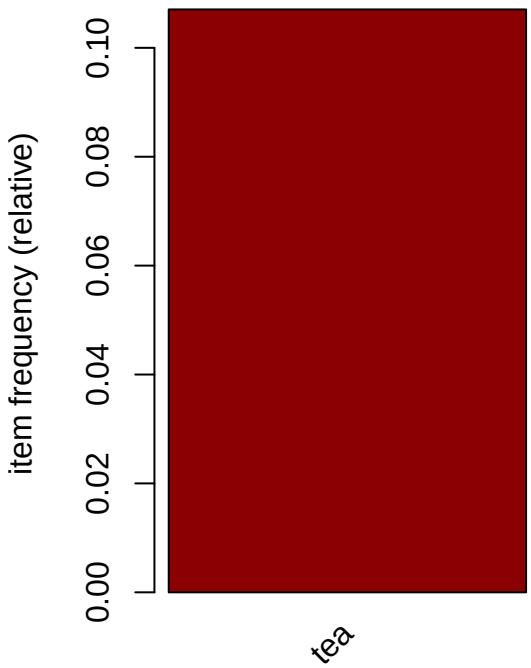
```
# looking at summary of data
#
summary(supa)
```

```
## transactions as itemMatrix in sparse format with
##  7501 rows (elements/itemsets/transactions) and
##  5729 columns (items) and a density of 0.0005421748
##
## most frequent items:
##     tea   wheat mineral     fat  yogurt (Other)
##     803     645     577     574     543   20157
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   15   16
## 1603 2007 1382  942  651  407  228  151   70   39   13    5    1    1    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.106   4.000  16.000
##
## includes extended item information - examples:
##                    labels
## 1                       &
## 2             accessories
## 3 accessories,antioxydant
```

Frequent purchased items are tea, Wheat, mineral, fat & yoghurt

```
# visualizing top 10 most common items and items with atleast 10% importance
#
par(mfrow = c(1,2))
itemFrequencyPlot(supa, topN = 10,col="darkgreen")
itemFrequencyPlot(supa, support = 0.1,col="darkred")
```





Association Analysis

```
# Building a model based on association rules
# using the apriori function
# using Min Support as 0.001 and confidence as 0.8

rules <- apriori (supa, 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 ...[5729 item(s), 7501 transaction(s)] done [0.04s].
```

```
## sorting and recoding items ... [354 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [271 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules
```

```
## set of 271 rules
```

We have a set o f 271 rules

```
# looking at the summarry of rules
#
summary(rules)
```

```
## set of 271 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 107 144  20
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   3.000   2.679   3.000   4.000
##
## summary of quality measures:
##     support           confidence       coverage             lift
##  Min.   :0.001067   Min.   :0.800   Min.   :0.001067   Min.   :  7.611
##  1st Qu.:0.001200   1st Qu.:0.931   1st Qu.:0.001200   1st Qu.: 11.630
##  Median :0.001600   Median :1.000   Median :0.001600   Median : 13.068
##  Mean   :0.002834   Mean   :0.963   Mean   :0.002973   Mean   : 22.372
##  3rd Qu.:0.002666   3rd Qu.:1.000   3rd Qu.:0.002800   3rd Qu.: 20.218
##  Max.   :0.068391   Max.   :1.000   Max.   :0.076523   Max.   :613.718
##      count
##  Min.   :  8.00
##  1st Qu.:  9.00
##  Median : 12.00
##  Mean   : 21.26
##  3rd Qu.: 20.00
##  Max.   :513.00
##
## mining info:
##  data ntransactions support confidence
##  supa          7501   0.001        0.8
##                                                                  call
##  apriori(data = supa, parameter = list(supp = 0.001, conf = 0.8))
```

The 271 rules are distributed depending on items , the rules have 2,3 to 4 items

```
# Observing the first 5 rules built in our model
#
inspect(rules[1:5])
```

```
##     lhs                                rhs      support     confidence
## [1] {cookies,low}                   => {yogurt} 0.001066524 1
## [2] {cookies,low}                   => {fat}    0.001066524 1
## [3] {extra}                         => {dark}   0.001066524 1
## [4] {burgers,whole}                 => {wheat}  0.001199840 1
## [5] {fries,escalope,pasta,mushroom} => {cream}  0.001066524 1
##     coverage    lift     count
## [1] 0.001066524 13.81400 8
## [2] 0.001066524 13.06794 8
## [3] 0.001066524 83.34444 8
## [4] 0.001199840 11.62946 9
## [5] 0.001066524 47.77707 8
```

We observe that from the first rule if people who buys cookies and low are 100% likely to buy yogurt.

```
# Ordering these rules by confidence
#
rules<-sort(rules, by="confidence", decreasing=TRUE)
inspect(rules[1:5])
```

```
##     lhs                                rhs      support     confidence
## [1] {cookies,low}                   => {yogurt} 0.001066524 1
## [2] {cookies,low}                   => {fat}    0.001066524 1
## [3] {extra}                         => {dark}   0.001066524 1
## [4] {burgers,whole}                 => {wheat}  0.001199840 1
## [5] {fries,escalope,pasta,mushroom} => {cream}  0.001066524 1
##     coverage    lift     count
## [1] 0.001066524 13.81400 8
## [2] 0.001066524 13.06794 8
## [3] 0.001066524 83.34444 8
## [4] 0.001199840 11.62946 9
## [5] 0.001066524 47.77707 8
```

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

```
##     lhs                  rhs            support     confidence coverage
## [1] {&, fresh}        => {tuna,herb}    0.001199840 0.9        0.001333156
## [2] {parmesan, wheat} => {cheese,whole} 0.001333156 1.0        0.001333156
## [3] {fat, tea}        => {yogurt,green} 0.004666045 1.0        0.004666045
## [4] {&, grated}       => {cheese,herb}  0.004666045 1.0        0.004666045
## [5] {bar,hand}        => {protein}      0.001199840 1.0        0.001199840
##     lift     count
## [1] 613.7182  9
## [2] 258.6552 10
## [3] 197.3947 35
## [4] 153.0816 35
## [5] 144.2500  9
```

Ordering by lift and looking at the first rule, people who buy fresh also are 90% likely to buy tuna & herb.

```
# Ordering these rules by support in decreasing order
#
rules<-sort(rules, by="support", decreasing=TRUE)
inspect(rules[1:5])
```

```
##     lhs          rhs      support    confidence coverage   lift     count
## [1] {yogurt} => {fat}    0.06839088 0.9447514  0.07239035 12.34596 513
## [2] {fat}    => {yogurt} 0.06839088 0.8937282  0.07652313 12.34596 513
## [3] {herb}   => {&}      0.03092921 1.0000000  0.03092921 20.21833 232
## [4] {whole}  => {wheat}  0.01893081 0.9466667  0.01999733 11.00922 142
## [5] {rice}   => {wheat}  0.01226503 0.9583333  0.01279829 11.14490  92
```

looking at the first rule , people who buy yogurt are 94% likely to buy fat.

```
# Creating a subset of rules concerning Wheat
#
Wheat <- subset(rules, subset = rhs %pin% "wheat")

# previewing the first 5
#
inspect(Wheat[1:5])
```

```
##     lhs                   rhs     support     confidence coverage    lift
## [1] {whole}            => {wheat} 0.018930809 0.9466667  0.019997334 11.00922
## [2] {rice}             => {wheat} 0.012265031 0.9583333  0.012798294 11.14490
## [3] {water,whole}      => {wheat} 0.005599253 1.0000000  0.005599253 11.62946
## [4] {vegetables,whole} => {wheat} 0.004932676 1.0000000  0.004932676 11.62946
## [5] {pasta,ground}     => {wheat} 0.004532729 1.0000000  0.004532729 11.62946
##     count
## [1] 142
## [2]  92
## [3]  42
## [4]  37
## [5]  34
```

Looking at the first rule people who buy wheat are 94% likely to have bought whole The fifth rule shows that people who buy wheat are 95% likely to have bought rice as well.

```
# Looking at what other items people who previously bought wheat might as well buy
#
Wheat <- subset(rules, subset = lhs %pin% "wheat")

# Previewing the top 5 rules
#
inspect(Wheat[1:5])
```

```
##     lhs                       rhs            support     confidence coverage
## [1] {wheat, yogurt}       => {fat}          0.006932409 1.0000000  0.006932409
## [2] {fat, wheat}          => {yogurt}       0.006932409 0.8125000  0.008532196
## [3] {herb, wheat}         => {&}            0.003466205 1.0000000  0.003466205
## [4] {pepper,whole, wheat} => {&}            0.002266364 0.9444444  0.002399680
```

```
## [5] {parmesan, wheat}     => {cheese,whole} 0.001333156 1.0000000  0.001333156
##     lift      count
## [1]  13.06794 52
## [2]  11.22387 52
## [3]  20.21833 26
## [4]  19.09509 17
## [5] 258.65517 10
```

the 1 st rule shows that people who Wheat had bought yogurt and are 100% likely to add fat in their items list.

The 5th rule states that people who bought wheat had parmesan and are 100% likely to buy cheese and whole as well.