Assignment Part B - 3

43320_Devashish_Prasad_Q11

30/10/2020

Read Dataset and print summery

```
df<-read.transactions("Groceries.csv",sep=",")
inspect(df[1:5])
```

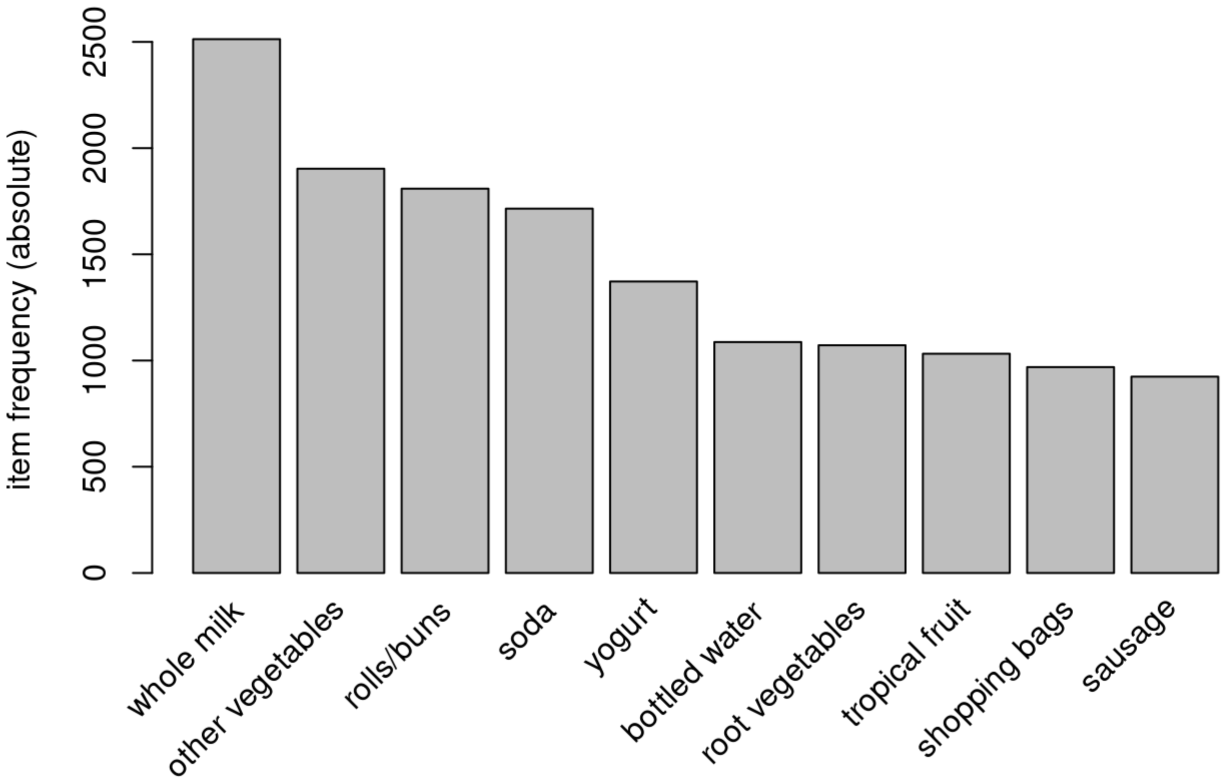
```
##     items
## [1] {citrus fruit,
##      margarine,
##      ready soups,
##      semi-finished bread}
## [2] {coffee,
##      tropical fruit,
##      yogurt}
## [3] {whole milk}
## [4] {cream cheese,
##      meat spreads,
##      pip fruit,
##      yogurt}
## [5] {condensed milk,
##      long life bakery product,
##      other vegetables,
##      whole milk}
```

```
summary(df)
```

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46
##   17   18   19   20   21   22   23   24   26   27   28   29   32
##   29   14   14    9   11    4    6    1    1    1    1    3    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
```

Frequency plot of top 10 items

```
itemFrequencyPlot(df,topN=10,type="absolute")
```



Get, summerize and print the rules

```
rules = apriori(df, parameter = list(supp = 0.03, conf = 0.3))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.3    0.1    1 none FALSE            TRUE       5    0.03      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: 295
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [44 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [14 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules)
```

```
## set of 14 rules
##
## rule length distribution (lhs + rhs):sizes
##  2
## 14
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2       2       2       2       2       2
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.03010   Min.   :0.3079   Min.   :0.07168   Min.   :1.205
##  1st Qu.:0.03249   1st Qu.:0.3298   1st Qu.:0.09021   1st Qu.:1.475
##  Median :0.03910   Median :0.3802   Median :0.10696   Median :1.575
##  Mean   :0.04260   Mean   :0.3759   Mean   :0.11484   Mean   :1.604
##  3rd Qu.:0.04853   3rd Qu.:0.4027   3rd Qu.:0.13226   3rd Qu.:1.759
##  Max.   :0.07483   Max.   :0.4496   Max.   :0.19349   Max.   :2.247
##      count
##  Min.   :296.0
##  1st Qu.:319.5
##  Median :384.5
##  Mean   :419.0
##  3rd Qu.:477.2
##  Max.   :736.0
##
## mining info:
##  data ntransactions support confidence
##    df          9835    0.03        0.3
```

```
inspect(rules)
```

```
##      lhs                        rhs                support    confidence
## [1]  {whipped/sour cream} => {whole milk}       0.03223183 0.4496454
## [2]  {pip fruit}          => {whole milk}       0.03009659 0.3978495
## [3]  {pastry}             => {whole milk}       0.03324860 0.3737143
## [4]  {citrus fruit}       => {whole milk}       0.03050330 0.3685504
## [5]  {sausage}            => {rolls/buns}       0.03060498 0.3257576
## [6]  {bottled water}      => {whole milk}       0.03436706 0.3109476
## [7]  {tropical fruit}     => {other vegetables} 0.03589222 0.3420543
## [8]  {tropical fruit}     => {whole milk}       0.04229792 0.4031008
## [9]  {root vegetables}    => {other vegetables} 0.04738180 0.4347015
## [10] {root vegetables}    => {whole milk}       0.04890696 0.4486940
## [11] {yogurt}             => {other vegetables} 0.04341637 0.3112245
## [12] {yogurt}             => {whole milk}       0.05602440 0.4016035
## [13] {rolls/buns}         => {whole milk}       0.05663447 0.3079049
## [14] {other vegetables}   => {whole milk}       0.07483477 0.3867578
##      coverage   lift     count
## [1]  0.07168277 1.759754 317
## [2]  0.07564820 1.557043 296
## [3]  0.08896797 1.462587 327
## [4]  0.08276563 1.442377 300
## [5]  0.09395018 1.771048 301
## [6]  0.11052364 1.216940 338
## [7]  0.10493137 1.767790 353
## [8]  0.10493137 1.577595 416
## [9]  0.10899847 2.246605 466
## [10] 0.10899847 1.756031 481
## [11] 0.13950178 1.608457 427
## [12] 0.13950178 1.571735 551
## [13] 0.18393493 1.205032 557
## [14] 0.19349263 1.513634 736
```

Sort the rules by confidence

```
rules = sort(rules, by = "confidence")
options(digits = 2)
inspect(rules)
```

```
##      lhs                     rhs                support confidence coverage
## [1]  {whipped/sour cream} => {whole milk}       0.032   0.45       0.072
## [2]  {root vegetables}    => {whole milk}       0.049   0.45       0.109
## [3]  {root vegetables}    => {other vegetables} 0.047   0.43       0.109
## [4]  {tropical fruit}     => {whole milk}       0.042   0.40       0.105
## [5]  {yogurt}             => {whole milk}       0.056   0.40       0.140
## [6]  {pip fruit}          => {whole milk}       0.030   0.40       0.076
## [7]  {other vegetables}   => {whole milk}       0.075   0.39       0.193
## [8]  {pastry}             => {whole milk}       0.033   0.37       0.089
## [9]  {citrus fruit}       => {whole milk}       0.031   0.37       0.083
## [10] {tropical fruit}     => {other vegetables} 0.036   0.34       0.105
## [11] {sausage}            => {rolls/buns}       0.031   0.33       0.094
## [12] {yogurt}             => {other vegetables} 0.043   0.31       0.140
## [13] {bottled water}      => {whole milk}       0.034   0.31       0.111
## [14] {rolls/buns}         => {whole milk}       0.057   0.31       0.184
##      lift count
## [1]  1.8  317
## [2]  1.8  481
## [3]  2.2  466
## [4]  1.6  416
## [5]  1.6  551
## [6]  1.6  296
## [7]  1.5  736
## [8]  1.5  327
## [9]  1.4  300
## [10] 1.8  353
## [11] 1.8  301
## [12] 1.6  427
## [13] 1.2  338
## [14] 1.2  557
```

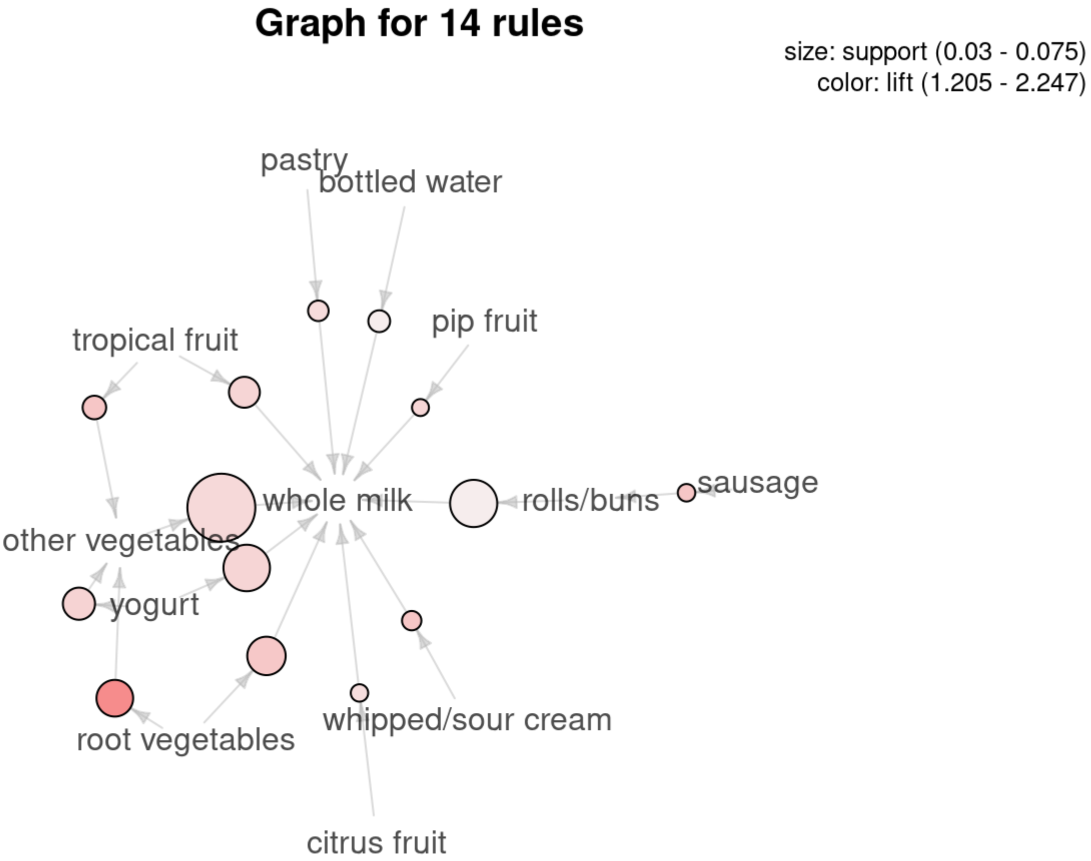
Inspect the reduntant rules, if redundancy rules<-rules[!is.reduntant(rules)]

```
rules[is.redundant(rules)]
```

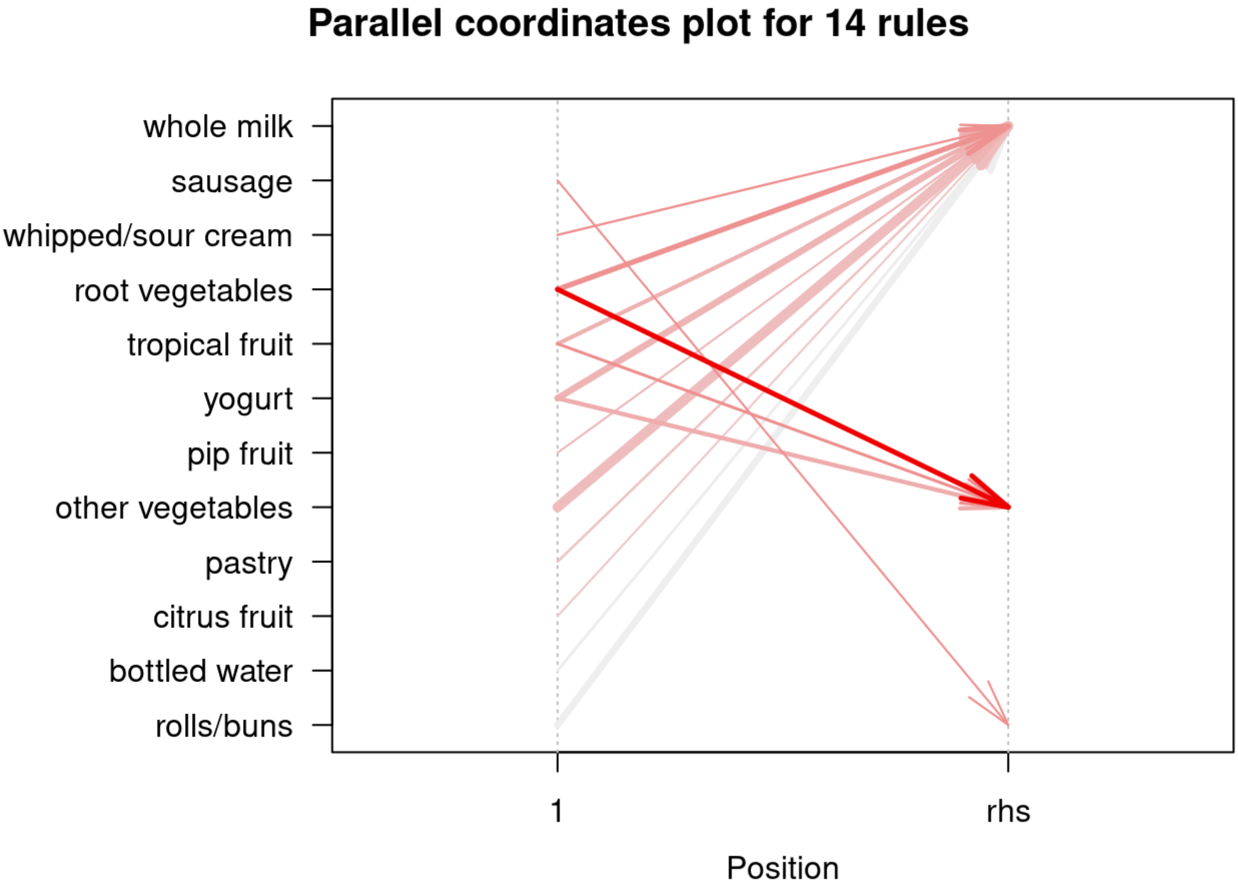
```
## set of 0 rules
```

Plot the graphs for the rules

```
plot(rules, method = "graph")
```

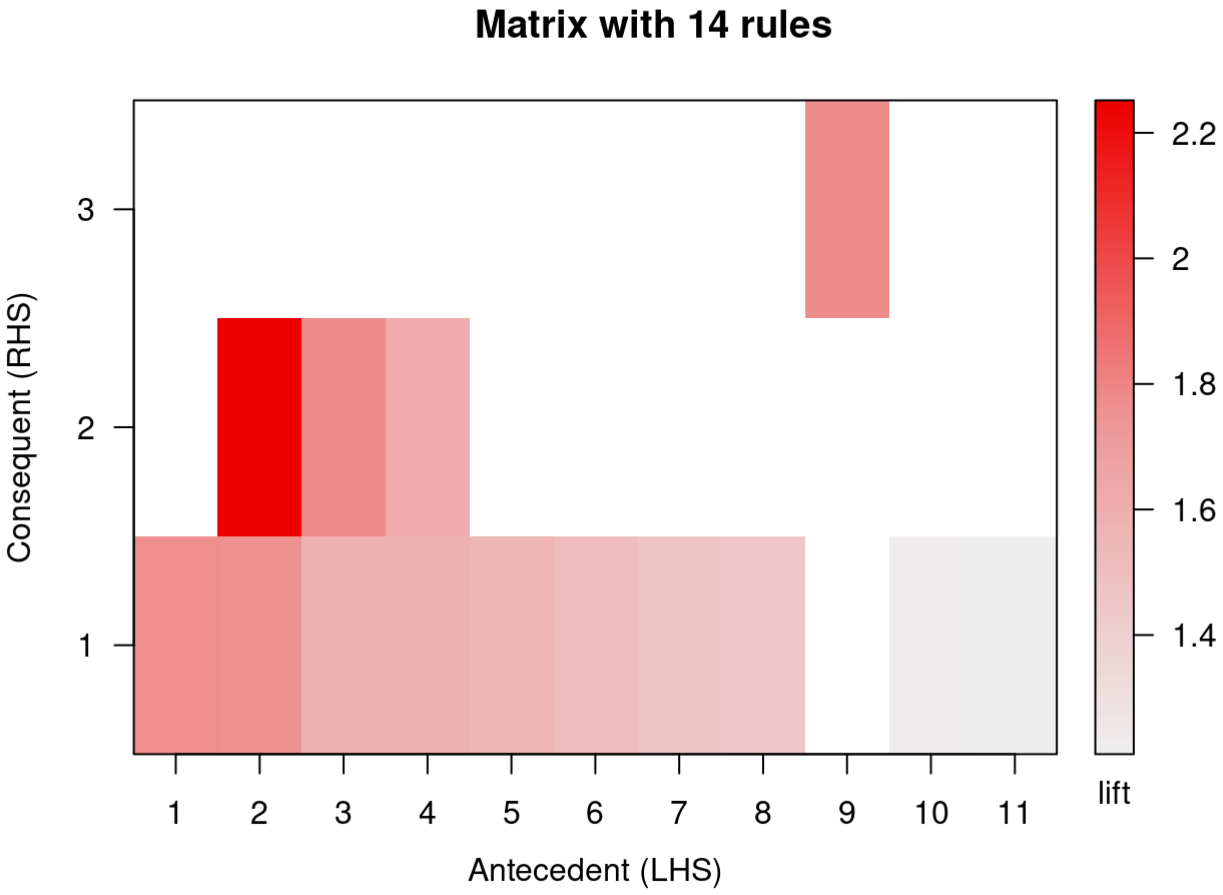


```
plot(rules, method = "paracoord")
```



```
plot(rules, method = "matrix", control = list(reorder = "none"))
```

```
## Itemsets in Antecedent (LHS)
##  [1] "{whipped/sour cream}" "{root vegetables}"    "{tropical fruit}"
##  [4] "{yogurt}"             "{pip fruit}"          "{other vegetables}"
##  [7] "{pastry}"             "{citrus fruit}"       "{sausage}"
## [10] "{bottled water}"      "{rolls/buns}"
## Itemsets in Consequent (RHS)
## [1] "{whole milk}"       "{other vegetables}" "{rolls/buns}"
```


Matrix with 14 rules
Consequent (RHS)
Antecedent (LHS)
lift