Datamining Project 3

Paichana Kularb 57090015

Modeling

Load libraries and data

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 3.4.2
```

```
library(arulesViz)
data("Groceries")
```

Summary of the data

```
summary(Groceries)
```

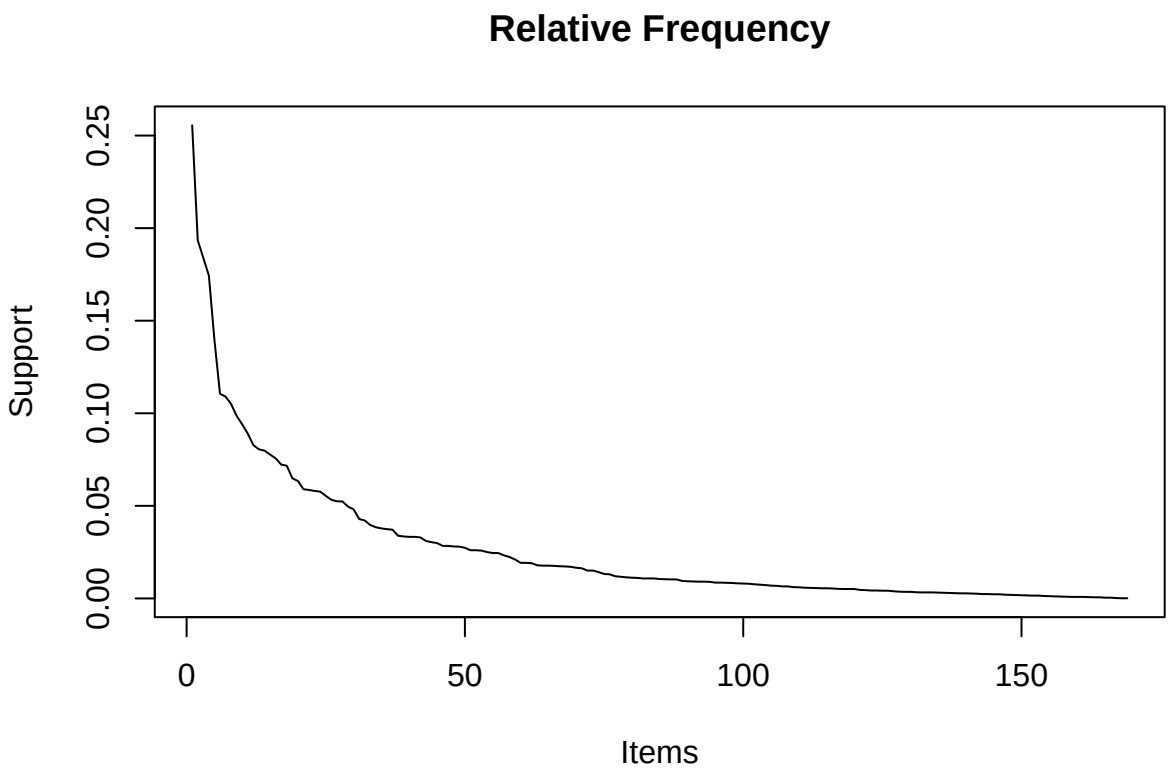
```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             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
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55
##   16   17   18   19   20   21   22   23   24   26   27   28   29   32
##   46   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  level2           level1
## 1 frankfurter sausage meat and sausage
## 2     sausage sausage meat and sausage
## 3  liver loaf sausage meat and sausage
```

Get the frequency of each item

```
freq = itemFrequency(Groceries,'relative')
freq = sort(freq,decreasing = TRUE)
```

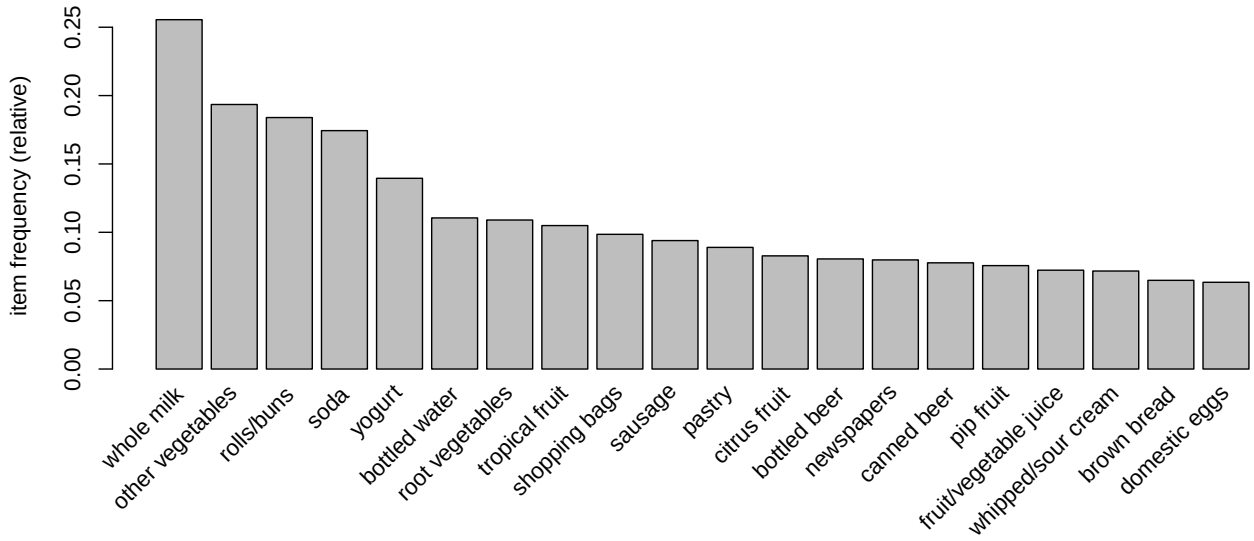
Visualize the all relative frequency of items

```
x = c(1:length(freq))
plot(x,freq,type = 'l',xlab = 'Items',ylab = 'Support',main = "Relative Frequency")
```



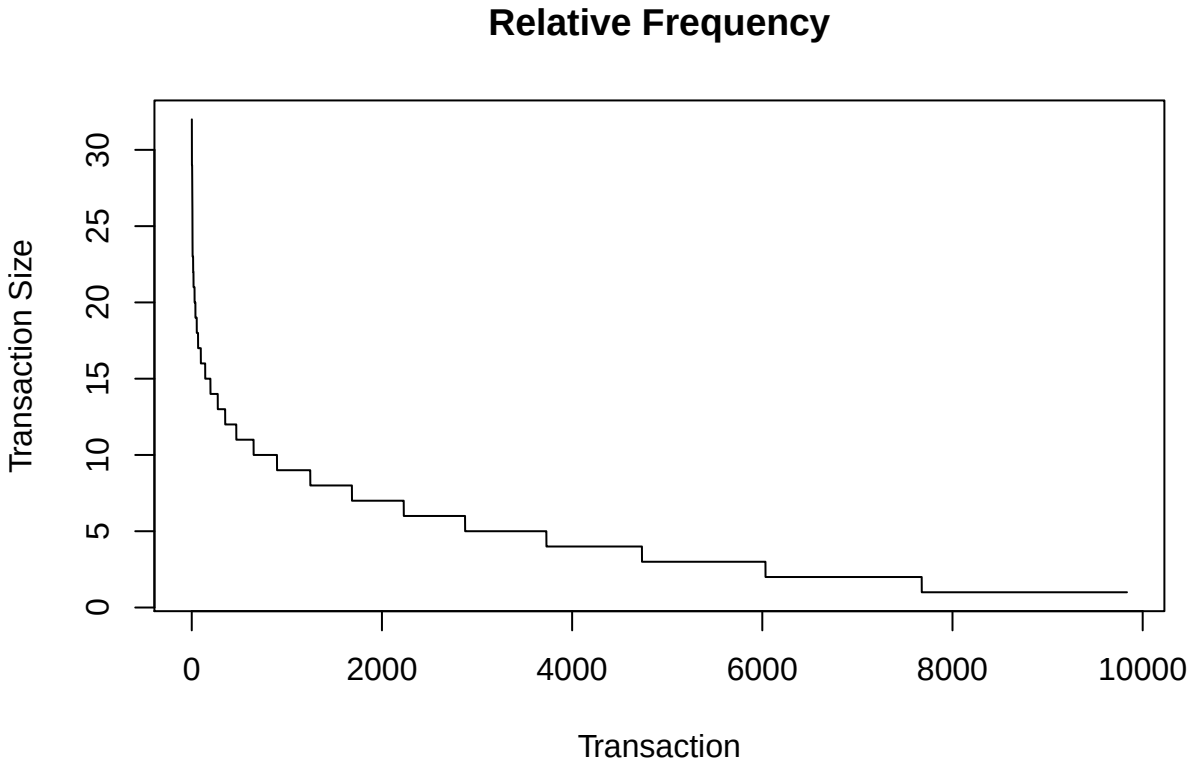
Visualize the top 20 items in terms of frequency

```
itemFrequencyPlot(Groceries,topN=20,type="relative")
```



Visualize the size of each transaction

```
transSize = sort(size(Groceries),decreasing = TRUE)
x = c(1:length(size(Groceries)))
plot(x,transSize,type = 'l',xlab = 'Transaction',ylab = 'Transaction Size',main = "Relative Frequency")
```



The rules are mine using the apriori algorithm:

Minimum support is 0.001 because the dataset is very sparse. The rules does not appear frequently in the dataset.

Minimum confidence is set to 0.75 because I want to be 75% sure that the rule is correct based on the past data.

```
minSup = 0.001
minCon = 0.75
rules = apriori(Groceries,parameter = list(supp = minSup,conf = minCon))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.75    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 done [0.02s].
```

```
## writing ... [777 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

777 rules are created

Contingency table that shows how many times an item is purchased together can be shown by

```
tb = crossTable(Groceries, sort=TRUE)
tb[1:5,1:5]
```

```
##                  whole milk other vegetables rolls/buns soda yogurt
## whole milk             2513              736        557  394    551
## other vegetables        736             1903        419  322    427
## rolls/buns              557              419       1809  377    338
## soda                    394              322        377 1715    269
## yogurt                  551              427        338  269   1372
```

Remove redundent rules, "A rule is redundant if a more general rules with the same or a higher confidence exists." quoted from https://cran.r-project.org/web/packages/arules/arules.pdf

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

```
## set of 739 rules
```

The top 10 rules based on the confidence level can be shown with the following script

```
rules = sort(rules,by=c('confidence'),decreasing = TRUE)
inspect(rules[1:10])
```

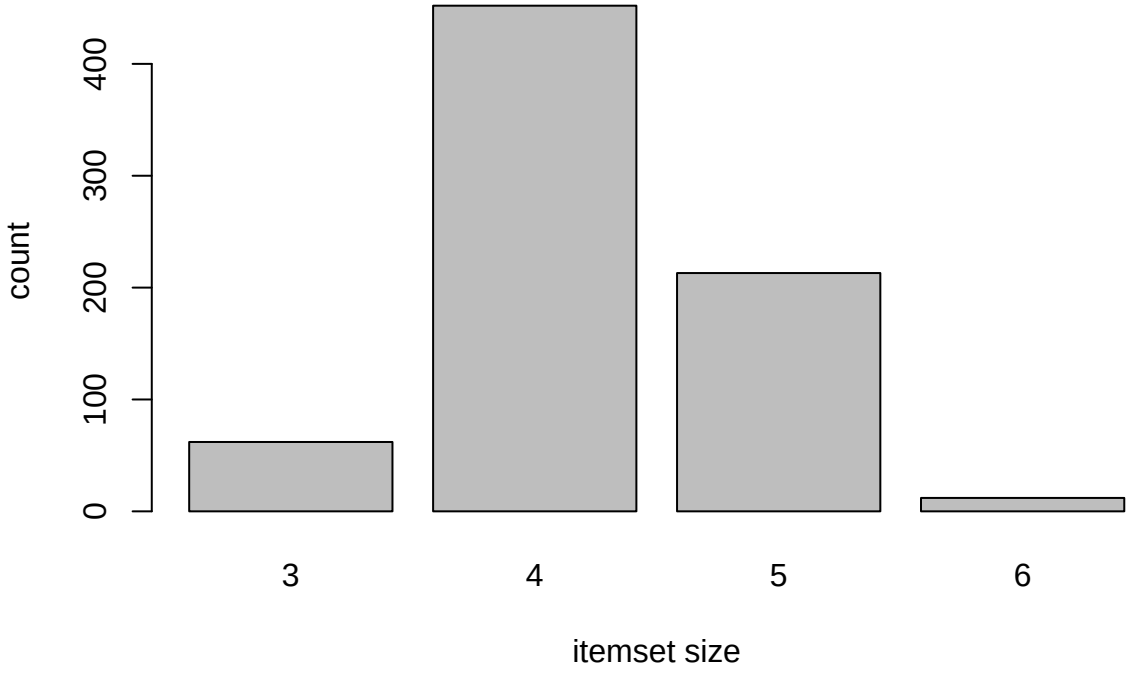
```
##      lhs                     rhs                    support confidence     lift count
## [1]  {rice,
##       sugar}              => {whole milk}       0.001220132          1 3.913649    12
## [2]  {canned fish,
##       hygiene articles}   => {whole milk}       0.001118454          1 3.913649    11
## [3]  {root vegetables,
##       butter,
##       rice}               => {whole milk}       0.001016777          1 3.913649    10
## [4]  {root vegetables,
##       whipped/sour cream,
##       flour}              => {whole milk}       0.001728521          1 3.913649    17
## [5]  {butter,
##       soft cheese,
##       domestic eggs}      => {whole milk}       0.001016777          1 3.913649    10
## [6]  {citrus fruit,
##       root vegetables,
##       soft cheese}        => {other vegetables} 0.001016777          1 5.168156    10
## [7]  {pip fruit,
##       butter,
##       hygiene articles}   => {whole milk}       0.001016777          1 3.913649    10
## [8]  {root vegetables,
##       whipped/sour cream,
##       hygiene articles}   => {whole milk}       0.001016777          1 3.913649    10
## [9]  {pip fruit,
##       root vegetables,
##       hygiene articles}   => {whole milk}       0.001016777          1 3.913649    10
## [10] {cream cheese ,
##       domestic eggs,
```

```
##       sugar}              => {whole milk}       0.001118454          1 3.913649    11
```

Most of the right handside of the rule is whole milk because whole milk occur in most transactions. And these rules involving whole milk will be above the minimum support.

The size of the item set in the rules can be shown by

```
barplot(table(size(rules)), xlab="itemset size", ylab="count")
```



Most of the rules generated have the size of 4

Evaluation

10 most interesting rules are

Rule 1

Highest confidence rule can be shown by

```
rules = sort(rules,by=c('confidence','lift','support','count'),decreasing = TRUE)
inspect(rules[1])
```

```
##     lhs                    rhs                    support confidence     lift count
## [1] {citrus fruit,
##      tropical fruit,
##      root vegetables,
##      whipped/sour cream} => {other vegetables} 0.001220132          1 5.168156    12
```

The rule above have the most confidence value which is 1. The transaction seems to be including varieties of fruits and vegetable which is sensible. But in my opinion whipped/sour cream should not belong in this rule. I could not think of a reason by whipped/sour cream -> other vegetables

Rule 2

Highest lift rule can be shown by

```
rules = sort(rules,by=c('lift','confidence','support','count'),decreasing = TRUE)
inspect(rules[1])
```

```
##     lhs                          rhs            support     confidence
## [1] {liquor,red/blush wine} => {bottled beer} 0.001931876 0.9047619
##     lift     count
## [1] 11.23527 19
```

The occurence of beer and {liquor,red/blush wine} is dependent to each other since the value of lift around 11.2. . This make sense since when buying alcohol berverages we buy more than one type.

Rule 3

Highest support rule can be shown by

```
rules = sort(rules,by=c('support','lift','confidence','count'),decreasing = TRUE)
inspect(rules[1])
```

```
##     lhs                  rhs                    support confidence     lift count
## [1] {citrus fruit,
##      tropical fruit,
##      root vegetables} => {other vegetables} 0.004473818  0.7857143 4.060694    44
```

The highest support is still low this is could be because the data contain many transactions. The rule shows that if a transaction contain citrus fruit, tropical fruit and root vegetables it is likely to contain other vegetables. This rule have the most support value because in groceries people often buy vetegatble and fruits. This rule is also a broader rule of the 1st rule and could be represent by it. It is not redundent because the more general rule confidence level is higher.

Rule 4

I like drinking milk and wanted to know what which item purchased leads to purchasing milk.

```
rhsMilk = subset(rules, subset = rhs %in% "whole milk" )
rhsMilk = sort(rhsMilk,by=c('lift','support','confidence','count'),decreasing = TRUE)
inspect(rhsMilk[1])
```

```
##     lhs                     rhs              support confidence     lift count
## [1] {root vegetables,
##      whipped/sour cream,
##      flour}              => {whole milk} 0.001728521          1 3.913649    17
```

{Whipped cream , flour} => whole milk make sense because these are ingredients to bake cakes. But root vetegable doesn't really goes into this category.

Rule 5

To diplay others rules where the items in the RHS which is different from the above(sort by lift)

```
r = subset(rules, subset = !(rhs %in% "whole milk" | rhs %in% "bottled beer" | rhs %in% "other vegetable
r = sort(r,by=c('lift','support','confidence','count'),decreasing = TRUE)
inspect(r[1])
```

```
##     lhs                          rhs                    support confidence   lift count
## [1] {citrus fruit,
##      other vegetables,
```

```
##      soda,
##      fruit/vegetable juice} => {root vegetables} 0.001016777  0.9090909 8.3404    10
```

Rules 6

Rules that doesn't involve whole milk, bottled beer and vetegables which is not interesting since rules above already include these items

```
r = subset(rules, subset = !(rhs %in% "whole milk" | rhs %in% "bottled beer" | rhs %in% "other vegetable
lhs %in% "whole milk" | lhs %in% "bottled beer" | lhs %in% "other vegetables" | lhs %in% "other vegetabl
r = sort(r,by=c('confidence','support','lift','count'),decreasing = TRUE)
inspect(r[1])
```

```
##     lhs                                  rhs      support     confidence
## [1] {sausage,pip fruit,sliced cheese} => {yogurt} 0.001220132 0.8571429
##     lift     count
## [1] 6.144315 12
```

I wanted to cover all items in the rhs, all rhs items can be shown by

```
df = data.frame(lhs = labels(lhs(rules), setStart = "", setEnd = ""),
           rhs = labels(rhs(rules), setStart = "", setEnd = ""))
summary(df$rhs)
```

```
##     bottled beer other vegetables       rolls/buns  root vegetables
##                1              230                1               14
##             soda   tropical fruit       whole milk           yogurt
##                2                5              449               37
```

Rules 7

Soda which are the remaing rhs items that have not be shown above can be shown by

```
r = subset(rules, subset = rhs %in% "soda" )
r = sort(r,by=c('lift','support','confidence','count'),decreasing = TRUE)
inspect(r[1])
```

```
##     lhs                          rhs    support     confidence lift
## [1] {coffee,misc. beverages} => {soda} 0.001016777 0.7692308  4.411303
##     count
## [1] 10
```

Rules 8

Tropical fruit which are the remaing rhs items that have not be shown above can be shown by

```
r = subset(rules, subset = rhs %in% "tropical fruit" )
r = sort(r,by=c('lift','support','confidence','count'),decreasing = TRUE)
inspect(r[1])
```

```
##     lhs                          rhs                  support confidence     lift count
## [1] {citrus fruit,
##      grapes,
##      fruit/vegetable juice} => {tropical fruit} 0.001118454  0.8461538 8.063879    11
```

Rules 9

Rolls/Buns fruit which are the remaing rhs items that have not be shown above can be shown by

```
r = subset(rules, subset = rhs %in% "rolls/buns" )
r = sort(r,by=c('lift','support','confidence','count'),decreasing = TRUE)
inspect(r[1])
```

```
##     lhs                           rhs          support     confidence
## [1] {spread cheese,newspapers} => {rolls/buns} 0.001220132 0.75
##     lift     count
## [1] 4.077529 12
```

Rules 10

The value of conviction is set to negative because the lower the better. Conviction measures lhs appears without rhs if they were dependent. The transaction with the lowest conviction can be shown by

```
quality(rules) = cbind(quality(rules),-interestMeasure(rules,c("gini","conviction"),transactions = Groc
r = sort(rules,by=c('conviction','lift','support','confidence','count'),decreasing = TRUE)
inspect(r[1])
```

```
##     lhs                             rhs          support    confidence
## [1] {tropical fruit,curd,yogurt} => {whole milk} 0.00396543 0.75
##     lift     count gini         conviction
## [1] 2.935237 39    -0.002599356 -2.977936
```

Deployment

The rules could be used by the store owner as a guide to where items should be located on the shelf. The information could also be use to suggest additional products to the user according to the user's purchase. This could improve the sales of the products which will result in a increase revenue.

The customer can benefit from this rule also, by suggestion of product he/she might want.

The effectiveness of the rules depends on the minimum support and minimum confidence.