Association Rule Mining

Objective: As a Data Scientist at FDMart Grocery you are asked to help analyze FDMarts transaction database to identify interesting patterns from the database. FDMart specializes in fresh vegetables and fruits. The store is considering expanding its product selection and wants to better understand its customers and their purchasing behavior. The Marketing Analyst has provided the following patterns as a starting point for analyzing the data

Overview of FDMart Grocery dataset:

According to summary mydata, FDMart grocery has 106 items t and total of 64809 transactions. The most frequent item found is fresh vegetables which is present in 30% of the total transaction in the dataset. Second most frequent item is fresh fruit. I set up support=0.01, Confident=0.5 to find association rules of the highest 5 lift. For example, people bought cooking oil and rice and also bought pots and pans together, which reflects strong positive correlation (lift=28.18).

Let X, Y be itemsets, $X \Rightarrow Y$ an association rule and T a set of transactions of a given database.

Support [edit]

Support is an indication of how frequently the itemset appears in the dataset.

The support of X with respect to T is defined as the proportion of transactions t in the dataset which contains the itemset X.

$$\mathrm{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

Confidence [edit]

Confidence is an indication of how often the rule has been found to be true.

The *confidence* value of a rule, $X \Rightarrow Y$, with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y.

Confidence is defined as:

$$\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X)$$

For example, the rule $\{\mathrm{butter, bread}\} \Rightarrow \{\mathrm{milk}\}$ has a confidence of $0.2/0.2 = 1.0$ in the database, which means that for 100% of the transactions conta
well).

Lift [edit]

The *lift* of a rule is defined as:

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X) \times \mathrm{supp}(Y)}$$

or the ratio of the observed support to that expected if X and Y were independent.[citation needed]

For example, the rule $\{\mathrm{milk, bread}\} \Rightarrow \{\mathrm{butter}\}$ has a lift of $\frac{0.2}{0.4 \times 0.4} = 1.25$.

If the rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.

If the lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.

If the lift is < 1, that lets us know the items are substitute to each other. This means that presence of one item has negative effect on presence of other item and vice versa.

The value of lift is that it considers both the support of the rule and the overall data set.[3]

```
library(arules)
library(grid)
library(arulesViz)
```

mydata <-
read.transactions("C:\\Users\\ludai\\Desktop\\BAN620\\Assignment2\\TransactionList1.csv",format="single",sep=",",cols=c(1,2))

class(mydata)

summary(mydata)

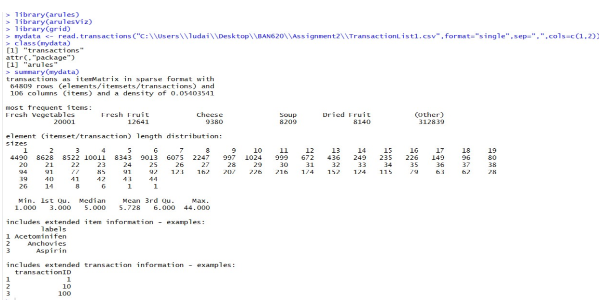
itemFrequencyPlot(mydata, support=0.1, cex.names=0.8)

itemFrequencyPlot(mydata,topN=20,type="absolute")

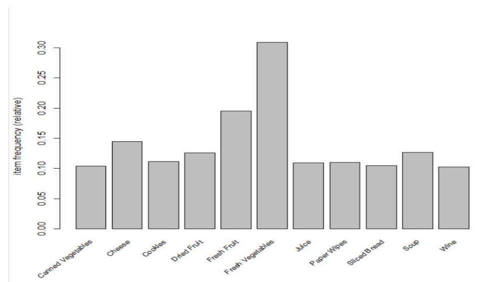
rules <- apriori(mydata, parameter = list(supp = 0.01, conf = 0.5))

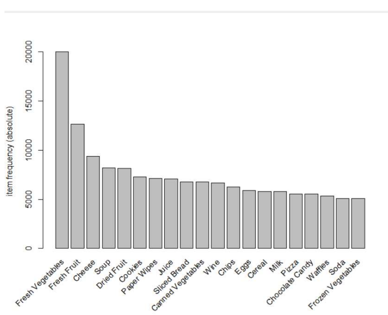
summary(rules)

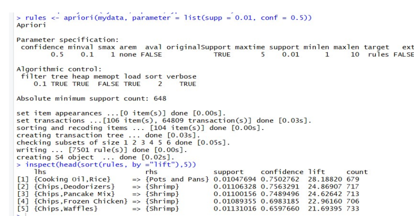
inspect(head(sort(rules, by ="lift"),5))



```
> library(arules)
> library(arulesViz)
> library(grid)
> mydata <- read.transactions("C:\\Users\\ludai\\Desktop\\BAN620\\Assignment2\\TransactionList1.csv",format="single",sep=",",cols=c(1,2))
> class(mydata)
[1] "transactions"
attr(,"package")
[1] "arules"
> summary(mydata)
transactions as itemMatrix in sparse format with
 64809 rows (elements/itemsets/transactions) and
 106 columns (items) and a density of 0.05403541

most frequent items:
Fresh Vegetables      Fresh Fruit           Cheese              Soup       Dried Fruit           (Other)
           20001            12641             9380              8209              8140            312839

element (itemset/transaction) length distribution:
sizes
    1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19
 4490  8628  8522 10011  8343  9013  6075  2247   997  1024   999   672   436   249   235   226   149    96    80
   20    21    22    23    24    25    26    27    28    29    30    31    32    33    34    35    36    37    38
   94    91    77    85    91    92   123   162   207   226   216   174   152   124   115    79    63    62    28
   39    40    41    42    43    44
   26    14     8     6     1     1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   5.000   5.728   6.000  44.000

includes extended item information - examples:
       labels
1 Acetominifen
2    Anchovies
3      Aspirin

includes extended transaction information - examples:
  transactionID
1             1
2            10
3           100
>
```







```
> rules <- apriori(mydata, parameter = list(supp = 0.01, conf = 0.5))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.5    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 648

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[106 item(s), 64809 transaction(s)] done [0.03s].
sorting and recoding items ... [104 item(s)] done [0.00s].
creating transaction tree ... done [0.03s].
checking subsets of size 1 2 3 4 5 6 done [0.05s].
writing ... [7501 rule(s)] done [0.00s].
creating S4 object  ... done [0.02s].
> inspect(head(sort(rules, by ="lift"),5))
    lhs                        rhs               support    confidence lift     count
[1] {Cooking Oil,Rice}      => {Pots and Pans} 0.01047694 0.7502762  28.18820 679
[2] {Chips,Deodorizers}     => {Shrimp}        0.01106328 0.7563291  24.86907 717
[3] {Chips,Pancake Mix}     => {Shrimp}        0.01100156 0.7489496  24.62642 713
[4] {Chips,Frozen Chicken} => {Shrimp}        0.01089355 0.6983185  22.96160 706
[5] {Chips,Waffles}         => {Shrimp}        0.01131016 0.6597660  21.69395 733
>
```

Analyze Question1: Purchase patterns related to beverages (Wine, Beer etc.)

Based on 16 rules (set wine on the right-hand side), we find that wine is combined with items like sauces, fresh vegetables, spices, and candles, but wine and beer are not associated. This means people may not buy wine and beer together.

In order to further explain this pattern about wine and beer, I set wine and beer in the left-hand side and beer on the right side respectively, the results (36 rules and 13 rules) indicate that Beer is mostly purchase with gums, pizza, frozen food items, eggs and Jam while wine is frequently purchased with fresh chicken, vegetables, and candles.

According to mining rules, we find 1.) wine and beer very rarely buy together. They have no correlation. 2.) Beer is mostly purchase with gums, pizza, eggs, chips, and frozen food items, while wine is frequently purchased with fresh vegetables, fresh chicken and candles. 3.) Beer is bought mostly in small baskets where there is less items. 4.) wine and candles have positive correlation. The people who buy candles are 62% likely to buy wine.

(See the following output in R)

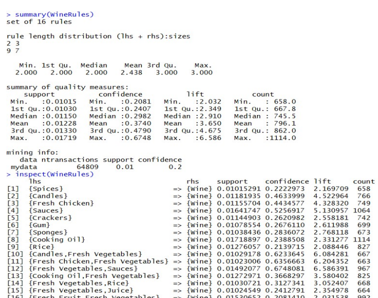
Find subset of rules that has Wine on the right hand side

WineRules <- subset(rules, subset = rhs %pin% "Wine")

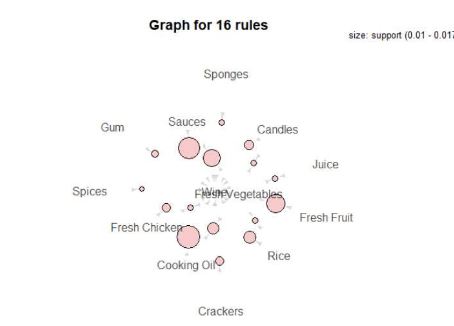
summary(WineRules)

inspect(WineRules)

plot(WineRules,method="graph",interactive=FALSE,shading=NA)



```
> summary(WineRules)
set of 16 rules

rule length distribution (lhs + rhs):sizes
2 3
9 7

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   2.000   2.438   3.000   3.000

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.01015   Min.   :0.2081   Min.   :2.032   Min.   : 658.0
 1st Qu.:0.01030   1st Qu.:0.2407   1st Qu.:2.349   1st Qu.: 667.8
 Median :0.01150   Median :0.2982   Median :2.910   Median : 745.5
 Mean   :0.01228   Mean   :0.3740   Mean   :3.650   Mean   : 796.1
 3rd Qu.:0.01330   3rd Qu.:0.4790   3rd Qu.:4.675   3rd Qu.: 862.0
 Max.   :0.01719   Max.   :0.6748   Max.   :6.586   Max.   :1114.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(WineRules)
     lhs                                  rhs    support    confidence lift     count
[1]  {Spices}                          => {Wine} 0.01015291 0.2222973  2.169709  658
[2]  {Candles}                         => {Wine} 0.01181935 0.4633999  4.522964  766
[3]  {Fresh Chicken}                   => {Wine} 0.01155704 0.4434577  4.328320  749
[4]  {Sauces}                          => {Wine} 0.01641747 0.5256917  5.130957 1064
[5]  {Crackers}                        => {Wine} 0.01144903 0.2620982  2.558181  742
[6]  {Gum}                             => {Wine} 0.01078554 0.2676110  2.611988  699
[7]  {Sponges}                         => {Wine} 0.01038436 0.2836072  2.768118  673
[8]  {Cooking Oil}                     => {Wine} 0.01718897 0.2388508  2.331277 1114
[9]  {Rice}                            => {Wine} 0.01276057 0.2139715  2.088446  827
[10] {Candles,Fresh Vegetables}        => {Wine} 0.01029178 0.6233645  6.084281  667
[11] {Fresh Chicken,Fresh Vegetables}  => {Wine} 0.01023006 0.6356663  6.204352  663
[12] {Fresh Vegetables,Sauces}         => {Wine} 0.01492077 0.6748081  6.586391  967
[13] {Cooking Oil,Fresh Vegetables}    => {Wine} 0.01272971 0.3668297  3.580402  825
[14] {Fresh Vegetables,Rice}           => {Wine} 0.01030721 0.3127341  3.052407  668
[15] {Fresh Vegetables,Juice}          => {Wine} 0.01024549 0.2412791  2.354978  664
[16] {Fresh Fruit,Fresh Vegetables}    => {Wine} 0.01530652 0.2081410  2.031538  992
```



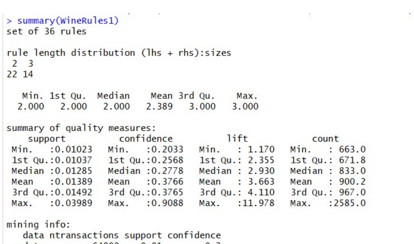
Find subset of rules that has Wine and Beer in the left hand side.

WineRules1 <- subset(rules, subset = lhs %ain% "Wine"|lhs %ain% "Beer")

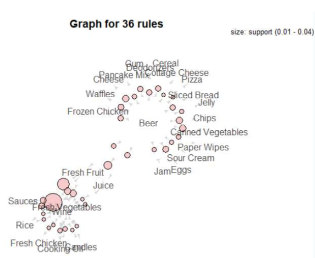
summary(WineRules1)

inspect(WineRules1)

plot(WineRules1,method="graph",interactive=FALSE,shading=NA)



```
> summary(WineRules1)
set of 36 rules

rule length distribution (lhs + rhs):sizes
 2  3
22 14

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   2.000   2.389   3.000   3.000

summary of quality measures:
    support           confidence          lift             count
 Min.   :0.01023   Min.   :0.2033   Min.   : 1.170   Min.   : 663.0
 1st Qu.:0.01037   1st Qu.:0.2568   1st Qu.: 2.355   1st Qu.: 671.8
 Median :0.01285   Median :0.2778   Median : 2.930   Median : 833.0
 Mean   :0.01389   Mean   :0.3766   Mean   : 3.663   Mean   : 900.2
 3rd Qu.:0.01492   3rd Qu.:0.3765   3rd Qu.: 4.110   3rd Qu.: 967.0
 Max.   :0.03989   Max.   :0.9088   Max.   :11.978   Max.   :2585.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
```



```
> inspect(WineRules1)
     lhs                        rhs                  support    confidence lift      count
[1]  {Beer}                  => {Gum}                0.01370180 0.2723091   6.756539  888
[2]  {Beer}                  => {Sour Cream}         0.01100156 0.2186446   4.609674  713
[3]  {Beer}                  => {Pizza}              0.01023006 0.2033119   2.373705  663
[4]  {Beer}                  => {Deodorizers}        0.01293030 0.2569764   6.200440  838
[5]  {Beer}                  => {Cottage Cheese}     0.01038436 0.2063784   3.723602  673
[6]  {Beer}                  => {Jam}                0.01104785 0.2195646   3.256238  716
[7]  {Beer}                  => {Jelly}              0.01083183 0.2152714   3.000973  702
[8]  {Beer}                  => {Frozen Chicken}     0.01365551 0.2713891   4.035902  885
[9]  {Beer}                  => {Chips}              0.01607801 0.3195339   3.304399 1042
[10] {Beer}                  => {Eggs}               0.01433443 0.2848819   3.144229  929
[11] {Beer}                  => {Pancake Mix}        0.01277600 0.2539098   4.709686  828
[12] {Beer}                  => {Waffles}            0.01404126 0.2790555   3.402692  910
[13] {Beer}                  => {Paper Wipes}        0.01181935 0.2348973   2.132137  766
[14] {Beer}                  => {Canned Vegetables}  0.01530652 0.3042012   2.915554  992
[15] {Beer}                  => {Cereal}             0.01382524 0.2747623   3.068069  896
[16] {Beer}                  => {Sliced Bread}       0.01399497 0.2781355   2.655914  907
[17] {Beer}                  => {Juice}              0.01396411 0.2775222   2.540746  905
[18] {Beer}                  => {Cheese}             0.01533738 0.3048145   2.106047  994
[19] {Beer}                  => {Fresh Fruit}        0.01237482 0.2459368   1.260891  802
[20] {Beer}                  => {Fresh Vegetables}   0.01816106 0.3609322   1.169524 1177
[21] {Wine}                  => {Fresh Fruit}        0.02632350 0.2569277   1.317240 1706
[22] {Wine}                  => {Fresh Vegetables}   0.03988644 0.3893072   1.261468 2585
[23] {Candles,Wine}          => {Fresh Vegetables}   0.01029178 0.8707572   2.821504  667
[24] {Fresh Vegetables,Wine} => {Candles}            0.01029178 0.2580271  10.116441  667
[25] {Fresh Chicken,Wine}    => {Fresh Vegetables}   0.01023006 0.8851802   2.868239  663
[26] {Fresh Vegetables,Wine} => {Fresh Chicken}      0.01023006 0.2564797   9.841440  663
[27] {Sauces,Wine}           => {Fresh Vegetables}   0.01492077 0.9088346   2.944886  967
[28] {Fresh Vegetables,Wine} => {Sauces}             0.01492077 0.3740812  11.978177  967
[29] {Cooking Oil,Wine}      => {Fresh Vegetables}   0.01272971 0.7405745   2.399675  825
[30] {Fresh Vegetables,Wine} => {Cooking Oil}        0.01272971 0.3191489   4.434761  825
[31] {Rice,Wine}             => {Fresh Vegetables}   0.01030721 0.8077388   2.617306  668
[32] {Fresh Vegetables,Wine} => {Rice}               0.01030721 0.2584139   4.333130  668
[33] {Juice,Wine}            => {Fresh Vegetables}   0.01024549 0.7272727   2.356573  664
[34] {Fresh Vegetables,Wine} => {Juice}              0.01024549 0.2568665   2.351641  664
[35] {Fresh Fruit,Wine}      => {Fresh Vegetables}   0.01530652 0.5814771   1.884153  992
[36] {Fresh Vegetables,Wine} => {Fresh Fruit}        0.01530652 0.3837524   1.967456  992
```

generating rules for beer on RHS

BeerRule<-apriori(data=mydata, parameter=list(supp=0.01,conf = 0.15,minlen=2),

appearance = list(default="lhs",rhs="Beer"),

control = list(verbose=F))

Sorting Beerrule by confidence in descending order

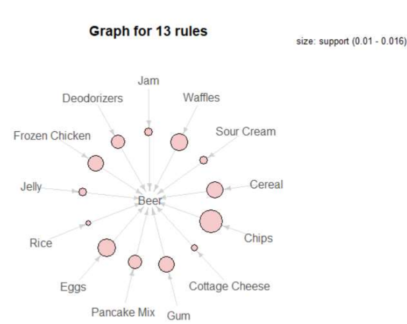
Beerrules1<-sort(BeerRule, decreasing=TRUE,by="confidence")

summary(Beerrules1)

inspect(Beerrules1)

plot(BeerRule,method="graph",interactive=FALSE,shading=NA)

```
> summary(Beerrules1)
set of 13 rules

rule length distribution (lhs + rhs):sizes
 2
13

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      2       2       2       2       2       2

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.01006   Min.   :0.1510   Min.   :3.001   Min.   : 652.0
 1st Qu.:0.01100   1st Qu.:0.1638   1st Qu.:3.256   1st Qu.: 713.0
 Median :0.01293   Median :0.1712   Median :3.403   Median : 838.0
 Mean   :0.01267   Mean   :0.2035   Mean   :4.043   Mean   : 820.9
 3rd Qu.:0.01383   3rd Qu.:0.2319   3rd Qu.:4.610   3rd Qu.: 896.0
 Max.   :0.01608   Max.   :0.3400   Max.   :6.757   Max.   :1042.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01       0.15
```




```
> inspect(Beerrules1)
     lhs                 rhs    support    confidence lift     count
[1]  {Gum}            => {Beer} 0.01370180 0.3399694  6.756539  888
[2]  {Deodorizers}    => {Beer} 0.01293030 0.3119881  6.200440  838
[3]  {Pancake Mix}    => {Beer} 0.01277600 0.2369777  4.709686  828
[4]  {Sour Cream}     => {Beer} 0.01100156 0.2319453  4.609674  713
[5]  {Frozen Chicken} => {Beer} 0.01365551 0.2030748  4.035902  885
[6]  {Cottage Cheese} => {Beer} 0.01038436 0.1873608  3.723602  673
[7]  {Waffles}        => {Beer} 0.01404126 0.1712135  3.402692  910
[8]  {Rice}           => {Beer} 0.01006033 0.1686934  3.352607  652
[9]  {Chips}          => {Beer} 0.01607801 0.1662678  3.304399 1042
[10] {Jam}            => {Beer} 0.01104785 0.1638444  3.256238  716
[11] {Eggs}           => {Beer} 0.01433443 0.1582084  3.144229  929
[12] {Cereal}         => {Beer} 0.01382524 0.1543763  3.068069  896
[13] {Jelly}          => {Beer} 0.01083183 0.1510002  3.000973  702
```

Analyze question2: Canned vs Fresh

Fresh food has mainly fresh vegetables, fresh fruits. Canned food is mainly canned vegetables and canned fruits.

We find with 864 item sets having fresh vegetables and 133 itemset with fresh fruits on the right-hand side of the itemset. (see summary for Fresh_Rules and Fresh_Rules1)

Based on the summary for fresh _rule2, fresh Fruit and fresh vegetables are positively correlated. People buy these two items also buy items like pasta, wine, rice, juice, and cheese. For example,

{Fresh Fruit, Fresh Vegetables} => {Pasta} supp=0.01510593 conf=0.2054133 lift=3.375414

Rules created for fresh vegetable and canned vegetable on the left-hand side (see summary for canned_Rules) have 203 itemset having canned vegetables and fresh vegetables together.

Canned vegetables and fresh vegetables are positively correlated. People buy these mostly with those items that are used for cooking meals for dinner and lunch e: g oil, pasta, rice, cheese, jelly, sour cream and wine.

We didn't find Canned fruits are frequent item and its sale may be independent of fresh food.

See the following output in R

Subrules for Fresh Vegetables on the rhs

Fresh_Rules <- subset(rules, subset = rhs %pin% "Fresh Vegetables")

summary(Fresh_Rules)

inspect(Fresh_Rules[1:20])

```
> summary(Fresh_Rules)
set of 864 rules

rule length distribution (lhs + rhs):sizes
  1   2   3   4   5
  1  74 296 460  33

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   4.000   3.521   4.000   5.000

summary of quality measures:
    support          confidence          lift             count
 Min.   :0.01001   Min.   :0.2198   Min.   :0.7122   Min.   :  649.0
 1st Qu.:0.01102   1st Qu.:0.6880   1st Qu.:2.2292   1st Qu.:  714.0
 Median :0.01129   Median :0.8294   Median :2.6874   Median :  732.0
 Mean   :0.01323   Mean   :0.7751   Mean   :2.5116   Mean   :  857.4
 3rd Qu.:0.01269   3rd Qu.:0.9089   3rd Qu.:2.9450   3rd Qu.:  822.5
 Max.   :0.30861   Max.   :0.9742   Max.   :3.1566   Max.   :20001.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(Fresh_Rules[1:20])
     lhs                    rhs                support    confidence lift      count
[1]  {}                  => {Fresh Vegetables} 0.30861454 0.3086145  1.0000000 20001
[2]  {Canned Fruit}      => {Fresh Vegetables} 0.01418013 0.4240886  1.3741692   919
[3]  {Deli Salads}       => {Fresh Vegetables} 0.01220509 0.2878457  0.9327030   791
[4]  {Personal Hygiene}  => {Fresh Vegetables} 0.01692666 0.3269747  1.0594921  1097
[5]  {Plastic Utensils}  => {Fresh Vegetables} 0.01127930 0.2515485  0.8150896   731
[6]  {Spices}            => {Fresh Vegetables} 0.01904057 0.4168919  1.3508498  1234
[7]  {Popcorn}           => {Fresh Vegetables} 0.01089355 0.2406271  0.7797012   706
[8]  {Aspirin}           => {Fresh Vegetables} 0.01185021 0.3221477  1.0438512   768
[9]  {Candles}           => {Fresh Vegetables} 0.01651005 0.6473079  2.0974641  1070
[10] {Fresh Chicken}     => {Fresh Vegetables} 0.01609344 0.6175252  2.0009594  1043
[11] {Pots and Pans}     => {Fresh Vegetables} 0.01660263 0.6237681  2.0211883  1076
[12] {Tofu}              => {Fresh Vegetables} 0.01154161 0.5006693  1.6223129   748
[13] {Fashion Magazines} => {Fresh Vegetables} 0.01274514 0.5481088  1.7760304   826
[14] {Popsicles}         => {Fresh Vegetables} 0.01555340 0.2593927  0.8405070  1008
[15] {Hard Candy}        => {Fresh Vegetables} 0.01538367 0.4253413  1.3782283   997
[16] {Sauces}            => {Fresh Vegetables} 0.02211113 0.7080040  2.2941367  1433
[17] {Oysters}           => {Fresh Vegetables} 0.01018377 0.4153556  1.3458717   660
[18] {Dips}              => {Fresh Vegetables} 0.01606258 0.2440225  0.7907032  1041
[19] {Sugar}             => {Fresh Vegetables} 0.01968862 0.5085692  1.6479105  1276
[20] {Tools}             => {Fresh Vegetables} 0.01612430 0.4477292  1.4507716  1045
```

Subrules for Fresh Fruit on the rhs

Fresh_Rules1 <- subset(rules, subset = rhs %pin% "Fresh Fruit")

summary(Fresh_Rules1)

inspect(Fresh_Rules1[1:20])

```
> summary(Fresh_Rules1)
set of 133 rules

rule length distribution (lhs + rhs):sizes
 2  3  4
49 83  1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.639   3.000   4.000

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01003   Min.   :0.2009   Min.   :1.030   Min.   : 650.0
 1st Qu.:0.01086   1st Qu.:0.3237   1st Qu.:1.659   1st Qu.: 704.0
 Median :0.01202   Median :0.4008   Median :2.055   Median : 779.0
 Mean   :0.01463   Mean   :0.4431   Mean   :2.272   Mean   : 948.1
 3rd Qu.:0.01629   3rd Qu.:0.5385   3rd Qu.:2.761   3rd Qu.:1056.0
 Max.   :0.07354   Max.   :0.8639   Max.   :4.429   Max.   :4766.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(Fresh_Rules1[1:20])
     lhs                    rhs           support    confidence lift     count
[1]  {Spices}            => {Fresh Fruit} 0.01405669 0.3077703  1.577904  911
[2]  {Candles}           => {Fresh Fruit} 0.01245197 0.4882033  2.502964  807
[3]  {Fresh Chicken}     => {Fresh Fruit} 0.01265256 0.4854944  2.489076  820
[4]  {Fashion Magazines} => {Fresh Fruit} 0.01016834 0.4372926  2.241951  659
[5]  {Hard Candy}        => {Fresh Fruit} 0.01158790 0.3203925  1.642617  751
[6]  {Sauces}            => {Fresh Fruit} 0.01228224 0.3932806  2.016306  796
[7]  {Tools}             => {Fresh Fruit} 0.01092441 0.3033419  1.555200  708
[8]  {Pasta}             => {Fresh Fruit} 0.02163280 0.3554767  1.822489 1402
[9]  {Bologna}           => {Fresh Fruit} 0.01041522 0.2085909  1.069422  675
[10] {TV Dinner}         => {Fresh Fruit} 0.01604715 0.3209877  1.645668 1040
[11] {Conditioner}       => {Fresh Fruit} 0.01526023 0.5079610  2.604259  989
[12] {Mouthwash}         => {Fresh Fruit} 0.01632489 0.3866959  1.982547 1058
[13] {Coffee}            => {Fresh Fruit} 0.01444244 0.2355310  1.207541  936
[14] {Shrimp}            => {Fresh Fruit} 0.01077011 0.3541350  1.815611  698
[15] {Lightbulbs}        => {Fresh Fruit} 0.01853138 0.2870459  1.471652 1201
[16] {Peanut Butter}     => {Fresh Fruit} 0.01317718 0.2475362  1.269091  854
[17] {Cleaners}          => {Fresh Fruit} 0.01785246 0.3346833  1.715884 1157
[18] {Cooking Oil}       => {Fresh Fruit} 0.01502878 0.2088336  1.070667  974
[19] {Yogurt}            => {Fresh Fruit} 0.01706553 0.3645353  1.868932 1106
[20] {Muffins}           => {Fresh Fruit} 0.01629403 0.2158626  1.106704 1056
```

Subrule for both Fresh Fruit and Fresh Vegetable on the lhs

Fresh_Rules2 <- subset(rules, subset = lhs %ain% c("Fresh Fruit", "Fresh Vegetables")) summary(Fresh_Rules2)

inspect(Fresh_Rules2)

```
> summary(Fresh_Rules2)
set of 7 rules

rule length distribution (lhs + rhs):sizes
3 4
5 2

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   3.000   3.000   3.286   3.500   4.000

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01048   Min.   :0.2048   Min.   : 1.506   Min.   : 679.0
 1st Qu.:0.01277   1st Qu.:0.2068   1st Qu.: 1.953   1st Qu.: 827.5
 Median :0.01511   Median :0.2180   Median : 3.375   Median : 979.0
 Mean   :0.01434   Mean   :0.3369   Mean   : 4.873   Mean   : 929.4
 3rd Qu.:0.01567   3rd Qu.:0.4141   3rd Qu.: 6.845   3rd Qu.:1015.5
 Max.   :0.01793   Max.   :0.6936   Max.   :11.630   Max.   :1162.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(Fresh_Rules2[1:20])
Error in slot(x, s)[i] : subscript out of bounds
> inspect(Fresh_Rules2)
    lhs                                   rhs      support    confidence lift      count
[1] {Fresh Fruit,Fresh Vegetables}       => {Pasta}  0.01510593 0.2054133  3.375414  979
[2] {Fresh Fruit,Fresh Vegetables}       => {Wine}   0.01530652 0.2081410  2.031538  992
[3] {Fresh Fruit,Fresh Vegetables}       => {Rice}   0.01792961 0.2438103  4.088254 1162
[4] {Fresh Fruit,Fresh Vegetables}       => {Juice}  0.01505964 0.2047839  1.874818  976
[5] {Fresh Fruit,Fresh Vegetables}       => {Cheese} 0.01603172 0.2180025  1.506239 1039
[6] {Fresh Fruit,Fresh Vegetables,Pasta} => {Rice}   0.01047694 0.6935649 11.629818  679
[7] {Fresh Fruit,Fresh Vegetables,Rice}  => {Pasta}  0.01047694 0.5843373  9.602008  679
>
```

Subrule for fresh Vegetable and Canned Vegetables on lhs.

canned_Rules <- subset(rules, subset = lhs %ain% c("Fresh Vegetables", "Canned Vegetables"))

summary(canned_Rules)

inspect(canned_Rules[1:20])

```
> summary(canned_Rules)
set of 203 rules

rule length distribution (lhs + rhs):sizes
  3   4
 21 182

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   4.000   4.000   3.897   4.000   4.000

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01021   Min.   :0.2571   Min.   : 1.836   Min.   : 662.0
 1st Qu.:0.01095   1st Qu.:0.7353   1st Qu.: 7.094   1st Qu.: 709.5
 Median :0.01111   Median :0.7533   Median :10.252   Median : 720.0
 Mean   :0.01134   Mean   :0.7122   Mean   :10.160   Mean   : 735.0
 3rd Qu.:0.01128   3rd Qu.:0.7705   3rd Qu.:12.981   3rd Qu.: 731.0
 Max.   :0.01649   Max.   :0.8193   Max.   :19.019   Max.   :1069.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
```

```
> inspect(canned_Rules[1:20])
     lhs                                   rhs               support    confidence lift     count
[1]  {Canned Vegetables,Fresh Vegetables} => {Shrimp}         0.01027635 0.2586408 8.504439  666
[2]  {Canned Vegetables,Fresh Vegetables} => {Peanut Butter}  0.01095527 0.2757282 5.179613  710
[3]  {Canned Vegetables,Fresh Vegetables} => {Sour Cream}     0.01451959 0.3654369 7.704489  941
[4]  {Canned Vegetables,Fresh Vegetables} => {Shampoo}        0.01021463 0.2570874 4.228826  662
[5]  {Canned Vegetables,Fresh Vegetables} => {Rice}           0.01425728 0.3588350 6.017008  924
[6]  {Canned Vegetables,Fresh Vegetables} => {Deli Meats}     0.01063124 0.2675728 3.576227  689
[7]  {Canned Vegetables,Fresh Vegetables} => {Deodorizers}    0.01370180 0.3448544 8.320799  888
[8]  {Canned Vegetables,Fresh Vegetables} => {Cottage Cheese} 0.01431900 0.3603883 6.502341  928
[9]  {Canned Vegetables,Fresh Vegetables} => {Milk}           0.01146446 0.2885437 3.229746  743
[10] {Canned Vegetables,Fresh Vegetables} => {Jam}            0.01458131 0.3669903 5.442626  945
[11] {Canned Vegetables,Fresh Vegetables} => {Jelly}          0.01459674 0.3673786 5.121412  946
[12] {Canned Vegetables,Fresh Vegetables} => {Frozen Chicken} 0.01459674 0.3673786 5.463387  946
[13] {Canned Vegetables,Fresh Vegetables} => {Chips}          0.01198908 0.3017476 3.120466  777
[14] {Canned Vegetables,Fresh Vegetables} => {Pancake Mix}    0.01391782 0.3502913 6.497432  902
[15] {Canned Vegetables,Fresh Vegetables} => {Waffles}        0.01482819 0.3732039 4.550700  961
[16] {Canned Vegetables,Fresh Vegetables} => {Paper Wipes}    0.01498249 0.3770874 3.422781  971
[17] {Canned Vegetables,Fresh Vegetables} => {Cereal}         0.01504421 0.3786408 4.228003  975
[18] {Canned Vegetables,Fresh Vegetables} => {Sliced Bread}   0.01484362 0.3735922 3.567429  962
[19] {Canned Vegetables,Fresh Vegetables} => {Juice}          0.01513679 0.3809709 3.487829  981
[20] {Canned Vegetables,Fresh Vegetables} => {Cheese}         0.01649462 0.4151456 2.868355 1069
```

Analyze question3: Small vs large transactions – how are they different?

Based on summary for rulesSmall, small baskets having less than or equal to 2 items are 787. Few items have strong positive correlation. For example,

{Candles} => {Fresh Chicken} 0.01035366 0.4059286 15.5757381

{Fresh Chicken} => {Candles} 0.01035366 0.3972765 15.5757381

{Candles} => {Sauces} 0.01027651 0.4029038 12.9008845

Based on summary for rulesLarge, large baskets having more than or equal to 5 items are 400. Fresh vegetables, fresh chicken, juice and sliced bread are found to be positively correlated with deodorizer (See the output of inspect ruleLarge with the highest 5 lift)

rule for small baskets with item less than or equal to 2

rulesSmall <- subset(rules, subset = size(rules) <=2)

#summary for ruleSmallSize

summary(rulesSmall)

inspect(rulesSmall[1:20])

```
> summary(rulesSmall)
set of 787 rules

rule length distribution (lhs + rhs):sizes
  1   2
  1 786

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   2.000   1.999   2.000   2.000

summary of quality measures:
    support           confidence          lift              count
 Min.   :0.01001   Min.   :0.2001   Min.   : 0.7122   Min.   :  649.0
 1st Qu.:0.01235   1st Qu.:0.2341   1st Qu.: 2.2477   1st Qu.:  800.5
 Median :0.01575   Median :0.2775   Median : 3.1529   Median : 1021.0
 Mean   :0.01782   Mean   :0.3033   Mean   : 3.6580   Mean   : 1154.7
 3rd Qu.:0.02092   3rd Qu.:0.3527   3rd Qu.: 4.7097   3rd Qu.: 1356.0
 Max.   :0.30861   Max.   :0.7080   Max.   :15.5760   Max.   :20001.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(rulesSmall[1:20])
     lhs                     rhs                support    confidence lift       count
[1]  {}                   => {Fresh Vegetables} 0.30861454 0.3086145   1.0000000 20001
[2]  {Canned Fruit}       => {Fresh Vegetables} 0.01418013 0.4240886   1.3741692   919
[3]  {Deli Salads}        => {Fresh Vegetables} 0.01220509 0.2878457   0.9327030   791
[4]  {Personal Hygiene}   => {Fresh Vegetables} 0.01692666 0.3269747   1.0594921  1097
[5]  {Plastic Utensils}   => {Fresh Vegetables} 0.01127930 0.2515485   0.8150896   731
[6]  {Spices}             => {Wine}             0.01015291 0.2222973   2.1697087   658
[7]  {Spices}             => {Fresh Fruit}      0.01405669 0.3077703   1.5779039   911
[8]  {Spices}             => {Fresh Vegetables} 0.01904057 0.4168919   1.3508498  1234
[9]  {Popcorn}            => {Fresh Vegetables} 0.01089355 0.2406271   0.7797012   706
[10] {Dried Meat}         => {Gum}              0.01056952 0.3728906   9.2521691   685
[11] {Gum}                => {Dried Meat}       0.01056952 0.2622511   9.2521691   685
[12] {Dried Meat}         => {Pizza}            0.01112500 0.3924878   4.5823705   721
[13] {Dried Meat}         => {Chips}            0.01080097 0.3810561   3.9406196   700
[14] {Aspirin}            => {Juice}            0.01010662 0.2747483   2.5153502   655
[15] {Aspirin}            => {Fresh Vegetables} 0.01185021 0.3221477   1.0438512   768
[16] {Pancakes}           => {Cereal}           0.01002947 0.3752887   4.1905728   650
[17] {Pancakes}           => {Sliced Bread}     0.01012205 0.3787529   3.6167078   656
[18] {Candles}            => {Fresh Chicken}    0.01035350 0.4059286  15.5759784   671
[19] {Fresh Chicken}      => {Candles}          0.01035350 0.3972765  15.5759784   671
[20] {Candles}            => {Sauces}           0.01027635 0.4029038  12.9010835   666
>
```

Subrule for Large baskets with item more than or equal to 5

rulesLarge <- subset(rules, subset = size(rules) >= 5)

summary(rulesLarge)

inspect(head(sort(rulesLarge, by ="lift"),5))

```
> summary(rulesLarge)
set of 400 rules

rule length distribution (lhs + rhs):sizes
  5
400

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      5       5       5       5       5       5

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01001   Min.   :0.6788   Min.   : 2.236   Min.   :649.0
 1st Qu.:0.01027   1st Qu.:0.7999   1st Qu.: 7.634   1st Qu.:665.8
 Median :0.01054   Median :0.8173   Median : 8.375   Median :683.0
 Mean   :0.01063   Mean   :0.8162   Mean   : 9.841   Mean   :688.6
 3rd Qu.:0.01083   3rd Qu.:0.8345   3rd Qu.:12.220   3rd Qu.:702.0
 Max.   :0.01258   Max.   :0.8884   Max.   :19.659   Max.   :815.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(head(sort(rulesLarge, by ="lift"),5))
    lhs                                                          rhs           support    confidence lift
[1] {Cottage Cheese,Fresh Vegetables,Frozen Chicken,Sliced Bread} => {Deodorizers} 0.01038436 0.8147700  19.65913
[2] {Fresh Vegetables,Frozen Chicken,Juice,Sliced Bread}         => {Deodorizers} 0.01058495 0.8108747  19.56514
[3] {Fresh Vegetables,Frozen Chicken,Pancake Mix,Sliced Bread}   => {Deodorizers} 0.01019920 0.8100490  19.54522
[4] {Cereal,Fresh Vegetables,Frozen Chicken,Sliced Bread}        => {Deodorizers} 0.01036893 0.8076923  19.48836
[5] {Frozen Chicken,Juice,Pancake Mix,Sliced Bread}              => {Deodorizers} 0.01018377 0.8068460  19.46794
    count
[1] 673
[2] 686
[3] 661
[4] 672
[5] 660
```

Analyze question4: Find one other interesting pattern

We know people like to eat cereal with milk as their breakfasts, so I want to mine rules for milk and cereal. I set milk and cereal on left side and right side respectively. The results indicate that milk and cereal have strong positive correlation (more than 80% confidence, and Lift >2). I also did interesting rules for milk and eggs, which have positive correlation. I find that the basket with milk and cereal also include some items like sliced bread, Jam, cheese, juice, which may be prepared for breakfast.

Milk on the Rhs and Cereal on lhs

Rulesinterest <- subset(rules, subset = rhs %pin% "Milk" & lhs %ain% "Cereal")

summary(Rulesinterest)

inspect(Rulesinterest)

```
> summary(Rulesinterest)
set of 24 rules

rule length distribution (lhs + rhs):sizes
 2  3  4  5
 1  7 15  1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   3.000   4.000   3.667   4.000   5.000

summary of quality measures:
    support           confidence          lift             count
 Min.   :0.01003   Min.   :0.2190   Min.   :2.451   Min.   : 650.0
 1st Qu.:0.01141   1st Qu.:0.5485   1st Qu.:6.139   1st Qu.: 739.8
 Median :0.01152   Median :0.6173   Median :6.910   Median : 746.5
 Mean   :0.01234   Mean   :0.5839   Mean   :6.603   Mean   : 799.8
 3rd Qu.:0.01359   3rd Qu.:[illegible]   3rd Qu.:7.576   3rd Qu.: 880.5
 Max.   :0.01961   Max.   :0.7088   Max.   :7.934   Max.   :1271.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(Rulesinterest)
     lhs                                      rhs    support    confidence lift     count
[1]  {Cereal}                              => {Milk} 0.01961147 0.2189669  2.451178 1271
[2]  {Cereal,Cottage Cheese}               => {Milk} 0.01365551 [illegible] 6.118616  885
[3]  {Cereal,Jam}                          => {Milk} [illegible] [illegible] [illegible]  904
[4]  {Cereal,Jelly}                        => {Milk} 0.01371723 0.5746006  6.432320  889
[5]  {Cereal,Waffles}                      => {Milk} 0.01356293 0.5380578  6.010311  879
[6]  {Cereal,Sliced Bread}                 => {Milk} 0.01407212 0.4532803  5.073686  912
[7]  {Cereal,Juice}                        => {Milk} 0.01396411 0.4408183  4.934196  905
[8]  {Cereal,Fresh Vegetables}             => {Milk} [illegible] [illegible] [illegible]  738
[9]  {Cereal,Cottage Cheese,Jam}           => {Milk} 0.01146446 0.6956929  7.787074  743
[10] {Cereal,Cottage Cheese,Jelly}         => {Milk} 0.01126387 0.6932574  7.759823  730
[11] {Cereal,Cottage Cheese,Waffles}       => {Milk} 0.01120215 0.6836158  7.651803  726
[12] {Cereal,Cottage Cheese,Sliced Bread}  => {Milk} [illegible] [illegible] [illegible]  749
[13] {Cereal,Cottage Cheese,Juice}         => {Milk} [illegible] [illegible] [illegible]  751
[14] {Cereal,Jam,Jelly}                    => {Milk} 0.01144903 0.6745455  7.550366  742
[15] {Cereal,Jam,Waffles}                  => {Milk} 0.01147989 0.6901670  7.725221  744
[16] {Cereal,Jam,Sliced Bread}             => {Milk} 0.01177306 0.6758193  7.564624  763
[17] {Cereal,Jam,Juice}                    => {Milk} [illegible] [illegible] [illegible]  766
[18] {Cereal,Jelly,Waffles}                => {Milk} [illegible] [illegible] [illegible]  722
[19] {Cereal,Jelly,Sliced Bread}           => {Milk} 0.01141817 0.6654676  7.448735  740
[20] {Cereal,Jelly,Juice}                  => {Milk} 0.01147989 0.6549296  7.330800  744
[21] {Cereal,Sliced Bread,Waffles}         => {Milk} 0.01140274 0.6158333  6.893185  739
[22] {Cereal,Juice,Waffles}                => {Milk} [illegible] [illegible] [illegible]  742
[23] {Cereal,Juice,Sliced Bread}           => {Milk} [illegible] [illegible] 6.145797  761
[24] {Cereal,Jam,Juice,Sliced Bread}       => {Milk} 0.01002947 0.7088332  7.934137  650
```

```
> inspect(Rulesinterest)
    lhs                             rhs    support    confidence lift     count
[1] {Eggs}                       => {Milk} 0.02167909 0.2392711  2.678225 1405
[2] {Eggs,Paper Wipes}           => {Milk} 0.01002947 0.4898267  5.482759  650
[3] {Eggs,Sliced Bread}          => {Milk} 0.01376352 0.5114679  5.724995  892
[4] {Eggs,Juice}                 => {Milk} 0.01404126 0.5194064  5.813853  910
[5] {Eggs,Juice,Sliced Bread} => {Milk} 0.01132559 0.6157718  6.892497  734
```

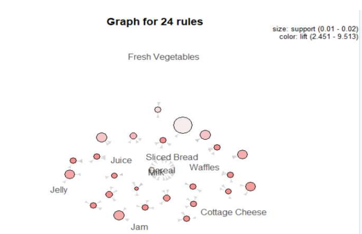
Milk on the lhs and Cereal on rhs

Rulesinterest1 <- subset(rules, subset = lhs %ain% "Milk" & rhs %ain% "Cereal")

summary(Rulesinterest1)

plot(Rulesinterest1, method="graph")

```
> summary(Rulesinterest1)
set of 24 rules

rule length distribution (lhs + rhs):sizes
 2  3  4  5
 1  7 15  1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   3.000   4.000   3.667   4.000   5.000

summary of quality measures:
    support           confidence          lift            count
 Min.   :0.01003   Min.   :0.2195   Min.   :2.451   Min.   : 650.0
 1st Qu.:0.01141   1st Qu.:0.6986   1st Qu.:7.801   1st Qu.: 739.8
 Median :0.01152   Median :0.8246   Median :9.208   Median : 746.5
 Mean   :0.01234   Mean   :0.7369   Mean   :8.228   Mean   : 799.8
 3rd Qu.:0.01359   3rd Qu.:0.8365   3rd Qu.:9.340   3rd Qu.: 880.5
 Max.   :0.01961   Max.   :0.8519   Max.   :9.513   Max.   :1271.0

mining info:
   data ntransactions support confidence
 mydata         64809    0.01        0.2
> inspect(Rulesinterest1)
     lhs                                    rhs      support    confidence lift     count
[1]  {Milk}                              => {Cereal} 0.01961147 0.2195164  2.451178 1271
[2]  {Cottage Cheese,Milk}               => {Cereal} 0.01365551 0.7577055  8.460740  885
[3]  {Jam,Milk}                          => {Cereal} 0.01394868 0.7062500  7.886174  904
[4]  {Jelly,Milk}                        => {Cereal} 0.01371723 0.7186742  8.024906  889
[5]  {Milk,Waffles}                      => {Cereal} 0.01356293 0.6756341  7.544309  879
[6]  {Milk,Sliced Bread}                 => {Cereal} 0.01407212 0.5595092  6.247628  912
[7]  {Juice,Milk}                        => {Cereal} 0.01396411 0.5441972  6.076650  905
[8]  {Fresh Vegetables,Milk}             => {Cereal} 0.01138731 0.3744292  4.180976  738
[9]  {Cottage Cheese,Jam,Milk}           => {Cereal} 0.01146446 0.8320269  9.290632  743
[10] {Cottage Cheese,Jelly,Milk}         => {Cereal} 0.01126387 0.8381171  9.358637  730
[11] {Cottage Cheese,Milk,Waffles}       => {Cereal} 0.01120215 0.8268793  9.233153  726
[12] {Cottage Cheese,Milk,Sliced Bread} => {Cereal} 0.01155704 0.8359375  9.334299  749
[13] {Cottage Cheese,Juice,Milk}         => {Cereal} 0.01158790 0.8400447  9.380162  751
[14] {Jam,Jelly,Milk}                    => {Cereal} 0.01144903 0.8318386  9.288530  742
[15] {Jam,Milk,Waffles}                  => {Cereal} 0.01147989 0.8175824  9.129342  744
[16] {Jam,Milk,Sliced Bread}             => {Cereal} 0.01177306 0.8293478  9.260717  763
[17] {Jam,Juice,Milk}                    => {Cereal} 0.01181935 0.8426843  9.409636  766
[18] {Jelly,Milk,Waffles}                => {Cereal} 0.01114043 0.8223235  9.182281  722
[19] {Jelly,Milk,Sliced Bread}           => {Cereal} 0.01141817 0.8390023  9.368521  740
[20] {Jelly,Juice,Milk}                  => {Cereal} 0.01147989 0.8406780  9.387233  744
[21] {Milk,Sliced Bread,Waffles}         => {Cereal} 0.01140274 0.8211111  9.168744  739
[22] {Juice,Milk,Waffles}                => {Cereal} 0.01144903 0.8281250  9.247063  742
[23] {Juice,Milk,Sliced Bread}           => {Cereal} 0.01174220 0.6320598  7.057747  761
[24] {Jam,Juice,Milk,Sliced Bread}       => {Cereal} 0.01002947 0.8519004  9.512545  650
```



Extra Question:

1. Find high utility itemsets (10 points)

Mining High Utility Itemsets from a transaction database is to find item sets that have utility beyond a given threshold. However, mining high utility itemsets presents a greater challenge than frequent itemset mining, since high utility itemsets lack the *anti-monotone* property of frequent itemsets. I used count function to mine the most top 6 utility transactions. This finding indicates that fresh vegetable, fresh fruits, cheese, juice, and dried fruit are high utility items, which are purchased most and also bought together. So, they are most profitable items at FDMart Grocery.

```
> inspect(head(sort(rules, by ="count")))
    lhs                   rhs                support    confidence lift     count
[1] {}                 => {Fresh Vegetables} 0.30861454 0.3086145  1.000000 20001
[2] {Fresh Fruit}      => {Fresh Vegetables} 0.07353917 0.3770271  1.221676  4766
[3] {Fresh Vegetables} => {Fresh Fruit}      0.07353917 0.2382881  1.221676  4766
[4] {Cheese}           => {Fresh Vegetables} 0.04953016 0.3422175  1.108883  3210
[5] {Juice}            => {Fresh Vegetables} 0.04246324 0.3887555  1.259680  2752
[6] {Dried Fruit}      => {Fresh Vegetables} 0.04212378 0.3353808  1.086730  2730
>
```

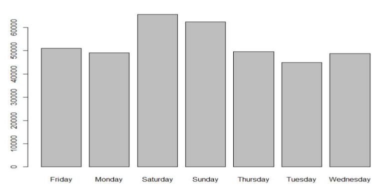
2. Weekdays vs Weekends – how do purchase patterns differ? (10 points)

I used count and aggregate function to analyze the purchased patterns of weekdays and weekends. The results indicate that the highest transaction frequency is on Saturday, following by Sunday and Friday. Fresh vegetable is the most frequent item purchased every day. This means FDMart Grocery should prepare sufficient items on weekends, and also have enough fresh vegetables for every day.

```
mydata1 <- read.csv("C:\\Users\\ludai\\Desktop\\BAN620\\Assignment2\\Time.csv")
mydata2 <- read.csv("C:\\Users\\ludai\\Desktop\\BAN620\\Assignment2\\TransactionListTime.csv",header = TRUE)
names(mydata1)[1]<-"time_id"
names(mydata2)[1]<-"transaction_id"
mydata3=merge(x=mydata1,y=mydata2,by='time_id',all.y=T)
summary(mydata3)
count(mydata3, vars = "the_day")
barplot(table(mydata3$the_day))
aggregate(mydata3[,4], list(the_day = mydata3$the_day),
      function(x) names(which.max(table(x))))
```

```
> count(mydata3, vars = "the_day")
    the_day  freq
1    Friday 50906
2    Monday 49136
3  Saturday 65517
4    Sunday 62347
5  Thursday 49608
6   Tuesday 44972
7 Wednesday 48724
```

```
> summary(mydata3)
    time_id          the_day       transaction_id           product.name
 Min.   : 367.0   Friday   :50906   Min.   :    1   Fresh Vegetables: 20001
 1st Qu.: 611.0   Monday   :49136   1st Qu.:22515   Fresh Fruit     : 12641
 Median : 804.0   Saturday :65517   Median :42660   Cheese          :  9380
 Mean   : 779.3   Sunday   :62347   Mean   :39847   Soup            :  8209
 3rd Qu.: 956.0   Thursday :49608   3rd Qu.:60716   Dried Fruit     :  8140
 Max.   :1095.0   Tuesday  :44972   Max.   :65308   Cookies         :  7254
                  Wednesday:48724                   (Other)         :305585
>
```




```
> aggregate(mydata3[,4], list(the_day = mydata3$the_day),
+           function(x) names(which.max(table(x))))
    the_day                x
1    Friday Fresh Vegetables
2    Monday Fresh Vegetables
3  Saturday Fresh Vegetables
4    Sunday Fresh Vegetables
5  Thursday Fresh Vegetables
6   Tuesday Fresh Vegetables
7 Wednesday Fresh Vegetables
```