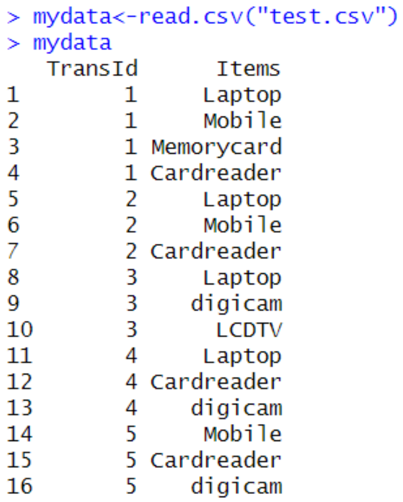
Name: Shaikh Inamul Hasan
Roll No: 100

Lab 6: **Association Apriori**

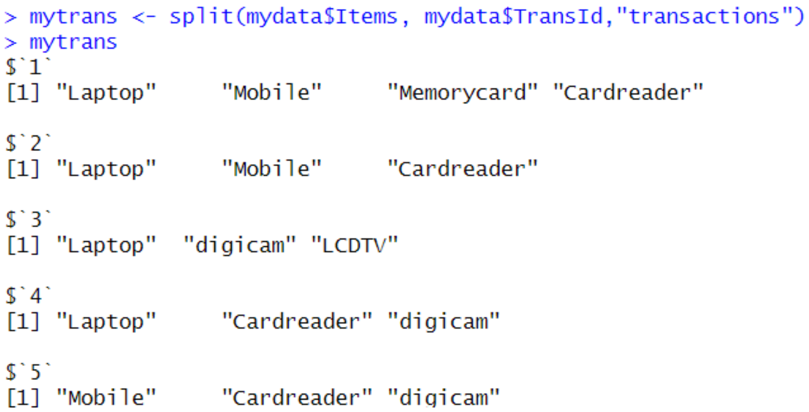
1. Find association rule using apriori algorithm with 50% support and 75% confidence for the following data. Find rules for maximum three frequent Itemset.

TransId	Items
1	Laptop, Mobile, Memory card, Card reader
2	Laptop, Mobile, Card reader
3	Laptop, digi cam, LCD TV
4	Laptop, Card reader, digi cam
5	Mobile, Card reader, digi cam

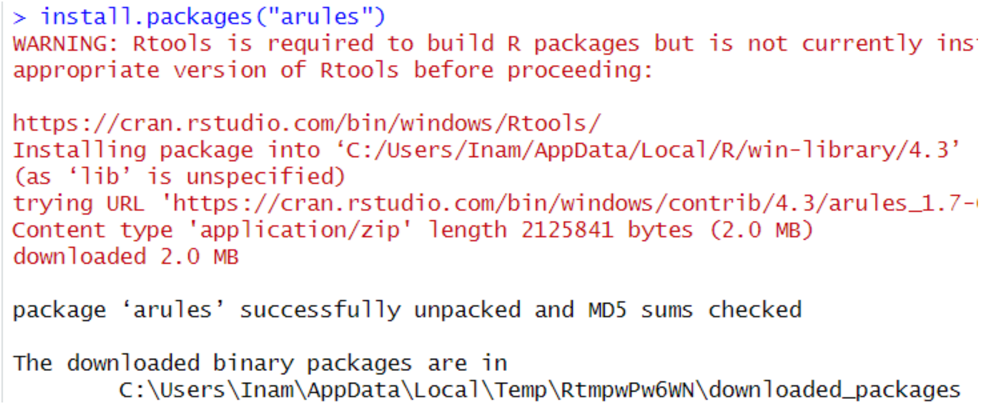
Code & Output:



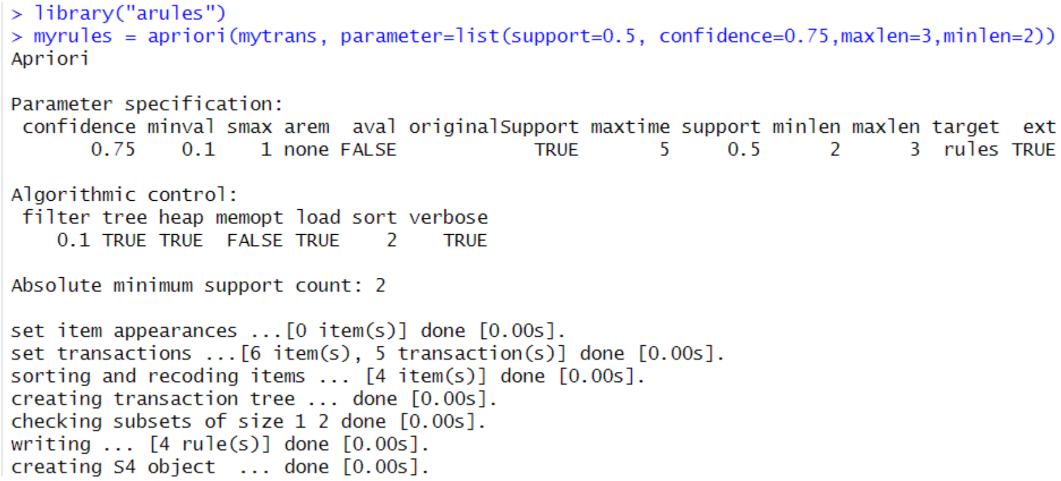
```
> mydata<-read.csv("test.csv")
> mydata
   TransId      Items
1        1     Laptop
2        1     Mobile
3        1 Memorycard
4        1 Cardreader
5        2     Laptop
6        2     Mobile
7        2 Cardreader
8        3     Laptop
9        3    digicam
10       3      LCDTV
11       4     Laptop
12       4 Cardreader
13       4    digicam
14       5     Mobile
15       5 Cardreader
16       5    digicam
```



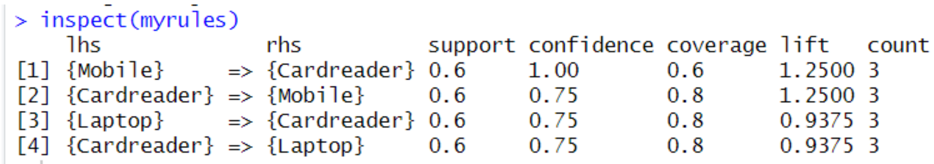
```
> mytrans <- split(mydata$Items, mydata$TransId,"transactions")
> mytrans
$`1`
[1] "Laptop"     "Mobile"     "Memorycard" "Cardreader"

$`2`
[1] "Laptop"     "Mobile"     "Cardreader"

$`3`
[1] "Laptop"  "digicam" "LCDTV"

$`4`
[1] "Laptop"     "Cardreader" "digicam"

$`5`
[1] "Mobile"     "Cardreader" "digicam"
```



```
> install.packages("arules")
WARNING: Rtools is required to build R packages but is not currently ins
appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Inam/AppData/Local/R/win-library/4.3’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/arules_1.7-
Content type 'application/zip' length 2125841 bytes (2.0 MB)
downloaded 2.0 MB

package ‘arules’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Inam\AppData\Local\Temp\RtmpwPw6WN\downloaded_packages
```



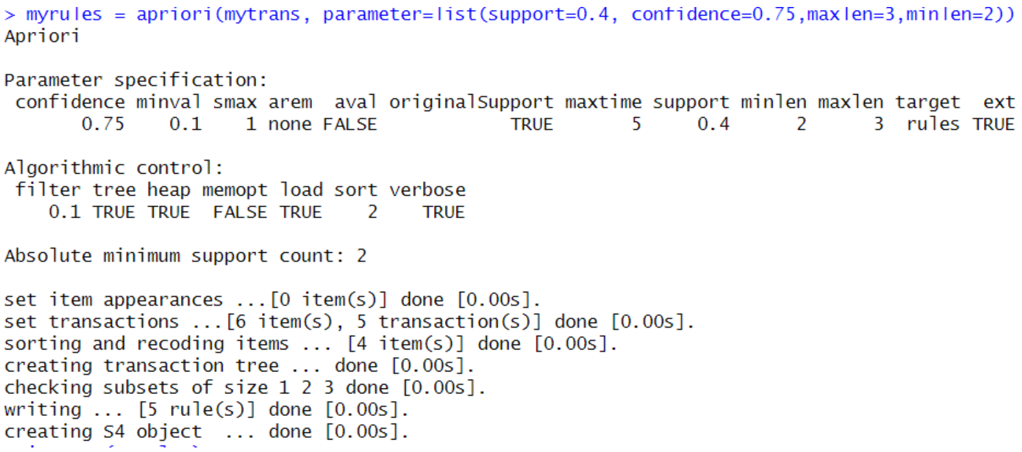
```
> library("arules")
> myrules = apriori(mytrans, parameter=list(support=0.5, confidence=0.75,maxlen=3,minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
       0.75    0.1    1 none FALSE            TRUE       5     0.5      2      3  rules TRUE

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

Absolute minimum support count: 2

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[6 item(s), 5 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [4 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```



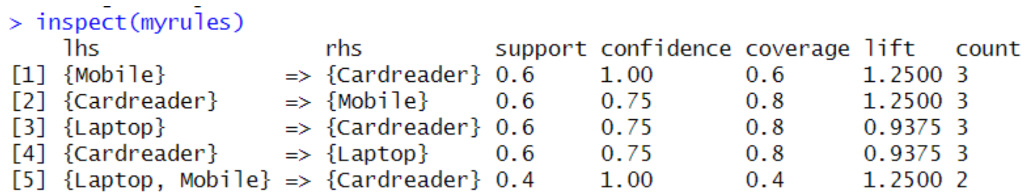
```
> inspect(myrules)
    lhs             rhs          support confidence coverage lift   count
[1] {Mobile}     => {Cardreader} 0.6     1.00       0.6      1.2500 3
[2] {Cardreader} => {Mobile}     0.6     0.75       0.8      1.2500 3
[3] {Laptop}     => {Cardreader} 0.6     0.75       0.8      0.9375 3
[4] {Cardreader} => {Laptop}     0.6     0.75       0.8      0.9375 3
```

2. For the above dataset find association rule using apriori algorithm with support =40% and confidence=75%.

Code & Output:



```
> myrules = apriori(mytrans, parameter=list(support=0.4, confidence=0.75,maxlen=3,minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
       0.75    0.1    1 none FALSE            TRUE       5     0.4      2      3  rules TRUE

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

Absolute minimum support count: 2

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[6 item(s), 5 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [5 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

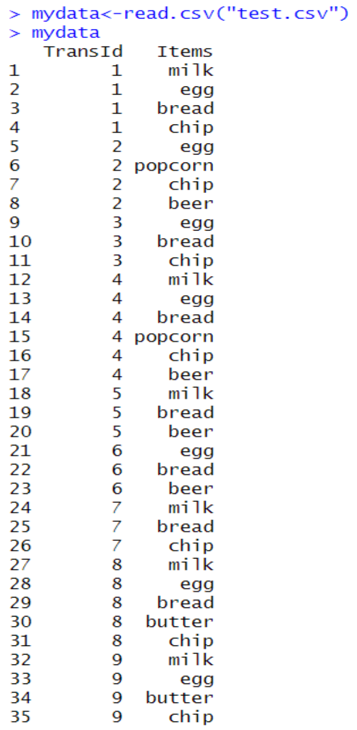


```
> inspect(myrules)
    lhs                 rhs          support confidence coverage lift   count
[1] {Mobile}         => {Cardreader} 0.6     1.00       0.6      1.2500 3    
[2] {Cardreader}     => {Mobile}     0.6     0.75       0.8      1.2500 3    
[3] {Laptop}         => {Cardreader} 0.6     0.75       0.8      0.9375 3    
[4] {Cardreader}     => {Laptop}     0.6     0.75       0.8      0.9375 3    
[5] {Laptop, Mobile} => {Cardreader} 0.4     1.00       0.4      1.2500 2    
```

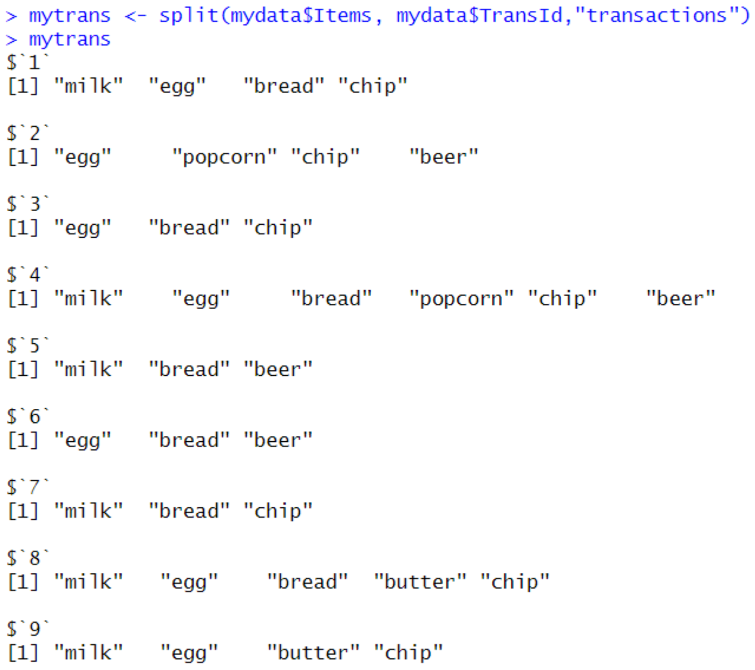
3. Find association rule with 30% support and 80% confidence for the following data. Find rules for maximum three frequent Itemset.

TransId	Items
1	milk, egg, bread, chip
2	egg, popcorn, chip, beer
3	egg, bread, chip
4	milk, egg, bread, popcorn, chip, beer
5	milk, bread, beer
6	egg, bread, beer
7	milk, bread, chip
8	milk, egg, bread, butter, chip
9	milk, egg, butter, chip

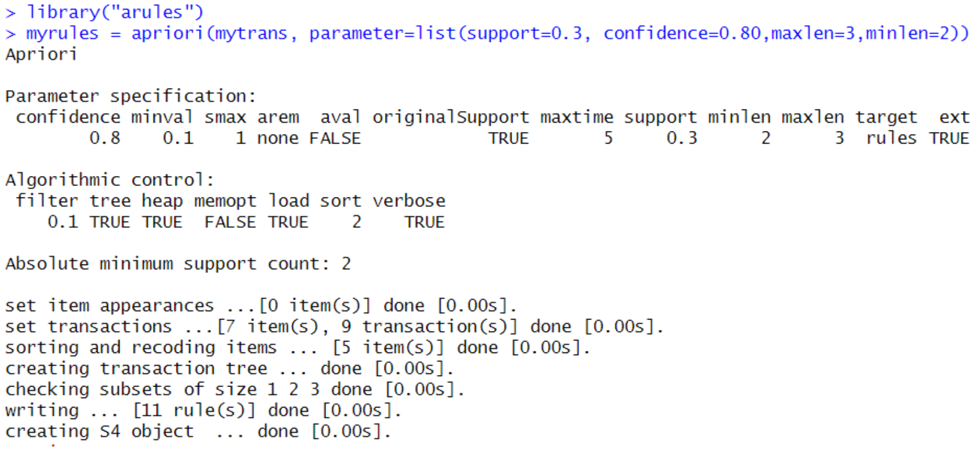
Code & Output:



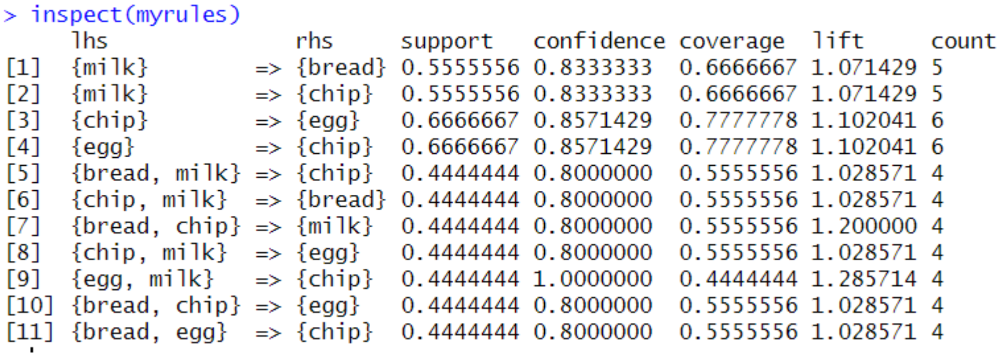
```
> mydata<-read.csv("test.csv")
> mydata
   TransId   Items
1        1    milk
2        1     egg
3        1   bread
4        1    chip
5        2     egg
6        2 popcorn
7        2    chip
8        2    beer
9        3     egg
10       3   bread
11       3    chip
12       4    milk
13       4     egg
14       4   bread
15       4 popcorn
16       4    chip
17       4    beer
18       5    milk
19       5   bread
20       5    beer
21       6     egg
22       6   bread
23       6    beer
24       7    milk
25       7   bread
26       7    chip
27       8    milk
28       8     egg
29       8   bread
30       8  butter
31       8    chip
32       9    milk
33       9     egg
34       9  butter
35       9    chip
```



```
> mytrans <- split(mydata$Items, mydata$TransId,"transactions")
> mytrans
$`1`
[1] "milk"  "egg"   "bread" "chip"

$`2`
[1] "egg"     "popcorn" "chip"    "beer"

$`3`
[1] "egg"   "bread" "chip"

$`4`
[1] "milk"    "egg"     "bread"   "popcorn" "chip"    "beer"

$`5`
[1] "milk"  "bread" "beer"

$`6`
[1] "egg"   "bread" "beer"

$`7`
[1] "milk"  "bread" "chip"

$`8`
[1] "milk"   "egg"    "bread"  "butter" "chip"

$`9`
[1] "milk"   "egg"    "butter" "chip"
```



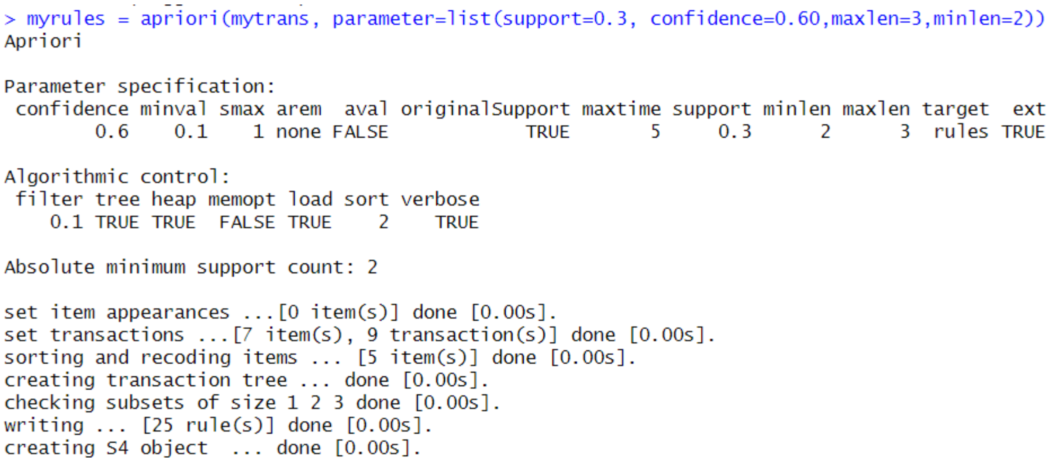
```
> library("arules")
> myrules = apriori(mytrans, parameter=list(support=0.3, confidence=0.80,maxlen=3,minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.3      2      3  rules TRUE

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

Absolute minimum support count: 2

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[7 item(s), 9 transaction(s)] done [0.00s].
sorting and recoding items ... [5 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [11 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```



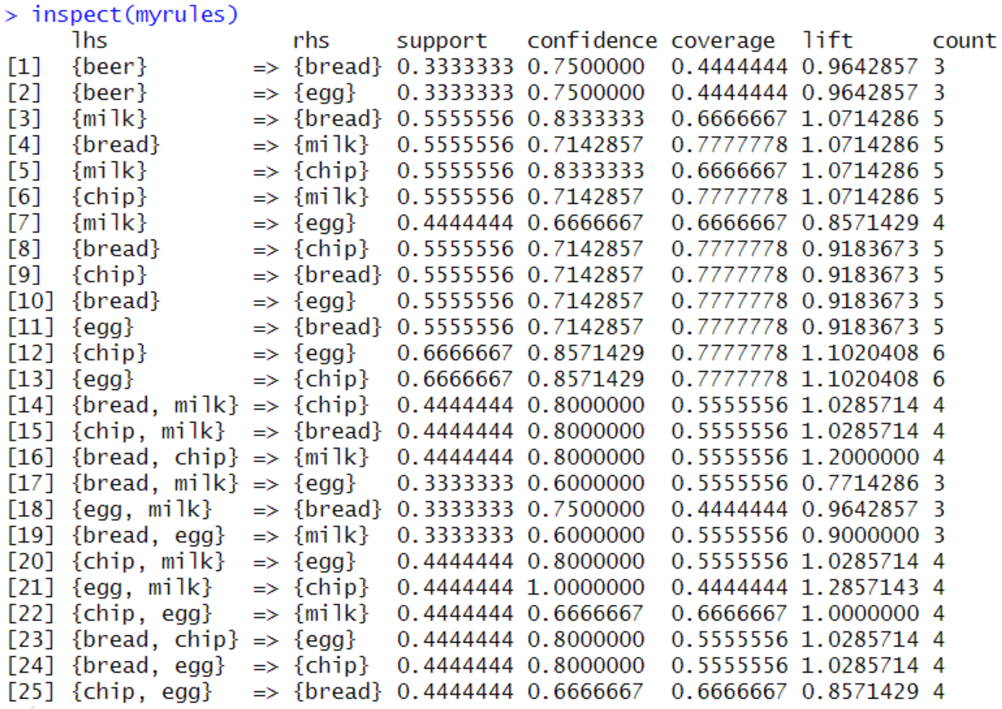
```
> inspect(myrules)
     lhs              rhs     support   confidence coverage  lift     count
[1]  {milk}        => {bread} 0.5555556 0.8333333  0.6666667 1.071429 5
[2]  {milk}        => {chip}  0.5555556 0.8333333  0.6666667 1.071429 5
[3]  {chip}        => {egg}   0.6666667 0.8571429  0.7777778 1.102041 6
[4]  {egg}         => {chip}  0.6666667 0.8571429  0.7777778 1.102041 6
[5]  {bread, milk} => {chip}  0.4444444 0.8000000  0.5555556 1.028571 4
[6]  {chip, milk}  => {bread} 0.4444444 0.8000000  0.5555556 1.028571 4
[7]  {bread, chip} => {milk}  0.4444444 0.8000000  0.5555556 1.200000 4
[8]  {chip, milk}  => {egg}   0.4444444 0.8000000  0.5555556 1.028571 4
[9]  {egg, milk}   => {chip}  0.4444444 1.0000000  0.4444444 1.285714 4
[10] {bread, chip} => {egg}   0.4444444 0.8000000  0.5555556 1.028571 4
[11] {bread, egg}  => {chip}  0.4444444 0.8000000  0.5555556 1.028571 4
```

4. For the above dataset find association rule with support =30% and confidence=60%.

Code & Output:



```
> myrules = apriori(mytrans, parameter=list(support=0.3, confidence=0.60,maxlen=3,minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.6    0.1    1 none FALSE            TRUE       5     0.3      2      3  rules TRUE

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

Absolute minimum support count: 2

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[7 item(s), 9 transaction(s)] done [0.00s].
sorting and recoding items ... [5 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [25 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```



```
> inspect(myrules)
     lhs              rhs     support   confidence coverage  lift      count
[1]  {beer}        => {bread} 0.3333333 0.7500000  0.4444444 0.9642857 3
[2]  {beer}        => {egg}   0.3333333 0.7500000  0.4444444 0.9642857 3
[3]  {milk}        => {bread} 0.5555556 0.8333333  0.6666667 1.0714286 5
[4]  {bread}       => {milk}  0.5555556 0.7142857  0.7777778 1.0714286 5
[5]  {milk}        => {chip}  0.5555556 0.8333333  0.6666667 1.0714286 5
[6]  {chip}        => {milk}  0.5555556 0.7142857  0.7777778 1.0714286 5
[7]  {milk}        => {egg}   0.4444444 0.6666667  0.6666667 0.8571429 4
[8]  {bread}       => {chip}  0.5555556 0.7142857  0.7777778 0.9183673 5
[9]  {chip}        => {bread} 0.5555556 0.7142857  0.7777778 0.9183673 5
[10] {bread}       => {egg}   0.5555556 0.7142857  0.7777778 0.9183673 5
[11] {egg}         => {bread} 0.5555556 0.7142857  0.7777778 0.9183673 5
[12] {chip}        => {egg}   0.6666667 0.8571429  0.7777778 1.1020408 6
[13] {egg}         => {chip}  0.6666667 0.8571429  0.7777778 1.1020408 6
[14] {bread, milk} => {chip}  0.4444444 0.8000000  0.5555556 1.0285714 4
[15] {chip, milk}  => {bread} 0.4444444 0.8000000  0.5555556 1.0285714 4
[16] {bread, chip} => {milk}  0.4444444 0.8000000  0.5555556 1.2000000 4
[17] {bread, milk} => {egg}   0.3333333 0.6000000  0.5555556 0.7714286 3
[18] {egg, milk}   => {bread} 0.3333333 0.7500000  0.4444444 0.9642857 3
[19] {bread, egg}  => {milk}  0.3333333 0.6000000  0.5555556 0.9000000 3
[20] {chip, milk}  => {egg}   0.4444444 0.8000000  0.5555556 1.0285714 4
[21] {egg, milk}   => {chip}  0.4444444 1.0000000  0.4444444 1.2857143 4
[22] {chip, egg}   => {milk}  0.4444444 0.6666667  0.6666667 1.0000000 4
[23] {bread, chip} => {egg}   0.4444444 0.8000000  0.5555556 1.0285714 4
[24] {bread, egg}  => {chip}  0.4444444 0.8000000  0.5555556 1.0285714 4
[25] {chip, egg}   => {bread} 0.4444444 0.6666667  0.6666667 0.8571429 4
```