Lab 6 - Association Rule Mining Assignments

Vũ Đặng Quỳnh Giang - 17133015

2020/12/06

Các câu hỏi dưới đây liên quan đến file `"browsing.txt"`. Đây là dữ liệu về các trang mà khách hàng truy cập trong mỗi phiên giao dịch của một cửa hàng bán lẻ online. Để đơn giản hóa cho bài tập này, dữ liệu đã được tiền xử lý để mỗi dòng là một phiên giao dịch gồm id của các trang (cách nhau bởi khoảng trắng).

Câu 1.

Load file `"browsing.txt"` lưu vào biến `browsing`. Cho biết các thông tin tổng quan về dữ liệu: dữ liệu có bao nhiêu phiên giao dịch (dòng), bao nhiêu trang (cột)? Liệt kê 5 trang được truy cập nhiều nhất. Bao nhiêu phiên giao dịch truy cập ít trang nhất, số lượng ít nhất là bao nhiêu? Bao nhiêu phiên giao dịch truy cập nhiều trang nhất, số lượng nhiều nhất là bao nhiêu? Trung bình có bao nhiêu trang được truy cập trên mỗi phiên giao dịch?

```
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

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

```
browsing <- read.transactions("browsing.txt", sep = " ", rm.duplicates = TRUE, header = TRUE)
```

```
## distribution of transactions with duplicates:
## 1
## 3
```

```
summary(browsing)
```

```
## transactions as itemMatrix in sparse format with
##  31100 rows (elements/itemsets/transactions) and
##  12592 columns (items) and a density of 0.0009724339
##
## most frequent items:
## DAI62779 FRO40251 ELE17451 GRO73461 SNA80324  (Other)
##     6667     3881     3874     3602     3044   359748
##
## element (itemset/transaction) length distribution:
## sizes
##    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
##   13  235  550  839 1554 2258 2536 2611 2428 2466 2282 2139 1925 1751 1492 1246
##   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32   33
## 1084  876  666  574  412  321  258  182  129  107   57   47   22   14    2    8
##   34   35   36   37
```

```
##    5    4    5    2
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    8.00   12.00   12.24   15.00   37.00
##
## includes extended item information - examples:
##     labels
## 1 DAI11153
## 2 DAI11223
## 3 DAI11238
```

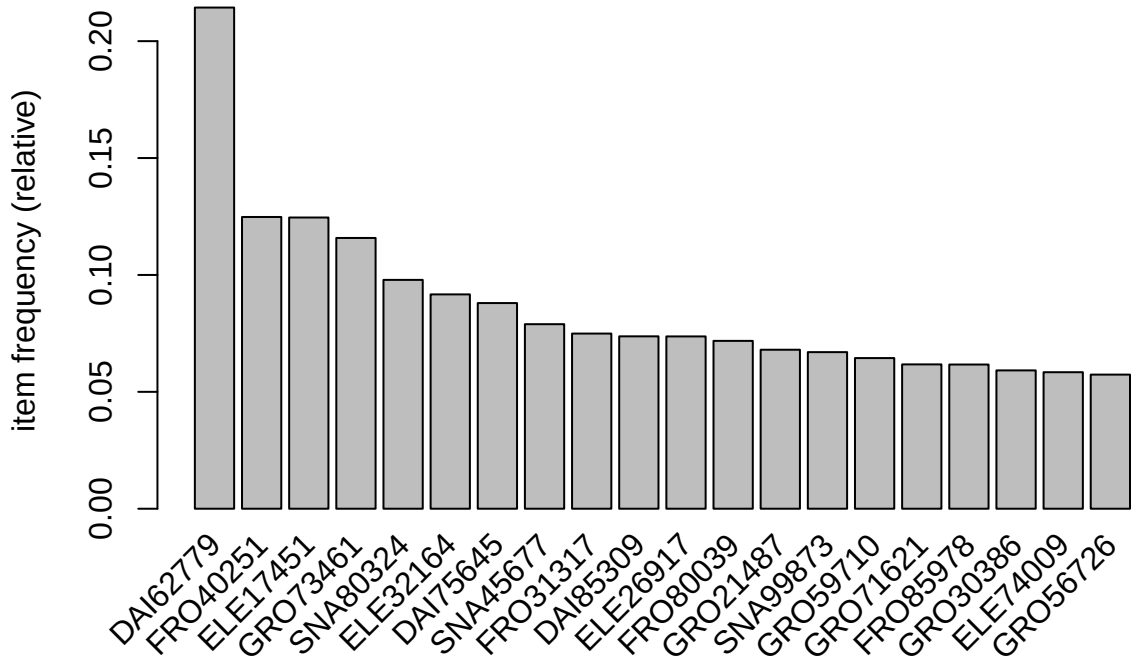
Câu 2.

Chọn một giá trị support mà bạn cho là phù hợp với dữ liệu. Giải thích ngắn gọn lý do bạn chọn giá trị đó. Vẽ biểu đồ tần số của các trang trong các phiên giao dịch với support bạn chọn. Vẽ biểu đồ tần số của top 20 trang được truy cập nhiều nhất trong các phiên giao dịch.

```
itemFrequency(browsing[, 1:20])
```

```
##     DAI11153     DAI11223     DAI11238     DAI11257     DAI11261     DAI11273
## 2.572347e-04 4.983923e-03 9.646302e-05 3.215434e-05 1.929260e-04 3.215434e-05
##     DAI11290     DAI11299     DAI11375     DAI11462     DAI11541     DAI11552
## 1.607717e-04 6.430868e-05 3.215434e-05 2.572347e-04 1.607717e-04 2.572347e-04
##     DAI11555     DAI11582     DAI11613     DAI11695     DAI11707     DAI11778
## 8.038585e-04 3.215434e-05 6.430868e-05 1.607717e-04 3.215434e-05 3.762058e-03
##     DAI11927     DAI11946
## 2.347267e-03 3.215434e-05
```

```
itemFrequencyPlot(browsing,topN  = 20)
```



Chọn giá trị support mininum là 0.05.

Có bao nhiêu frequent itemset? Bao nhiêu frequent itemset có số item (trang) ít nhất? Bao nhiêu frequent itemset có số item là nhiều nhất? Bao nhiêu frequent itemset có ít nhất k item trở lên (thử với k = 2, 3, ...)? Xem top 10 itemset sắp xếp theo support. Xem top 10 k-itemset sắp xếp theo support (thử với k = 1, 2, 3, ...). Vẽ biểu đồ tần số theo bậc (số lượng item) của các frequent itemset.

```
is.freq <- apriori(browsing, parameter = list(target = "frequent itemsets", support = 0.05))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen            target  ext
##      10 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1555
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12592 item(s), 31100 transaction(s)] done [0.15s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 done [0.00s].
## sorting transactions ... done [0.01s].
## writing ... [29 set(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
is.freq
```

```
## set of 29 itemsets
```

```
is.freq2 <- apriori(browsing, parameter = list(target = "frequent itemsets", support = 0.02, minlen =2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.02      2
##  maxlen            target  ext
##      10 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 622
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12592 item(s), 31100 transaction(s)] done [0.14s].
## sorting and recoding items ... [114 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## sorting transactions ... done [0.01s].
## writing ... [22 set(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
is.freq2
```

```
## set of 22 itemsets
```

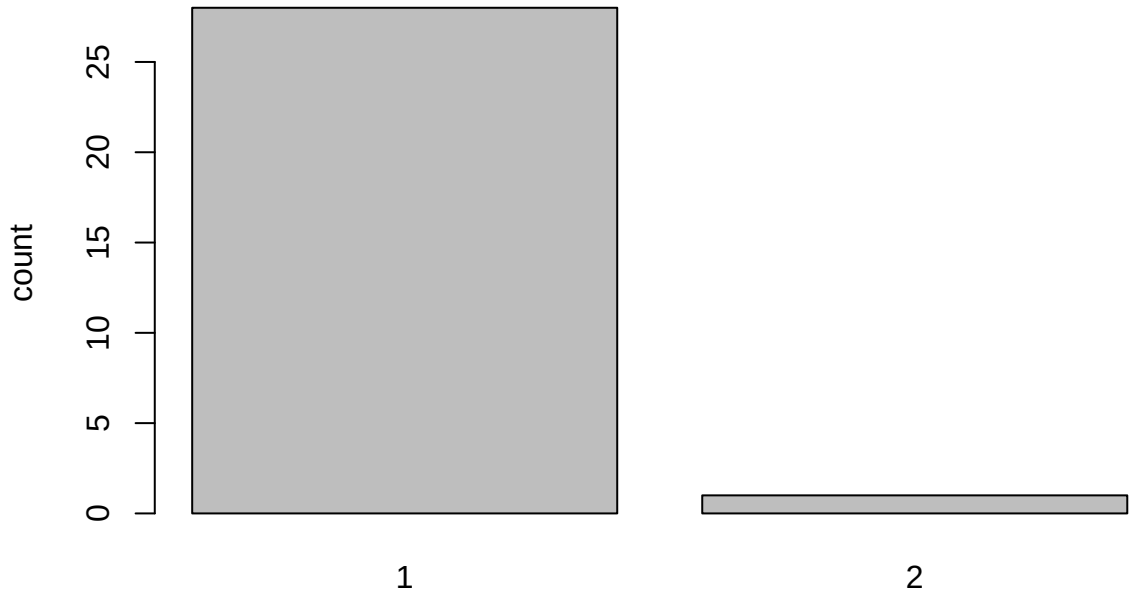
```
is.freq <- sort(is.freq, by = "support")
inspect(head(is.freq, n = 10))
```

```
##      items      support    transIdenticalToItemsets count
## [1]  {DAI62779} 0.21437299 0.025144695               6667
## [2]  {FRO40251} 0.12479100 0.013440514               3881
## [3]  {ELE17451} 0.12456592 0.012861736               3874
## [4]  {GRO73461} 0.11581994 0.006302251               3602
## [5]  {SNA80324} 0.09787781 0.009807074               3044
## [6]  {ELE32164} 0.09167203 0.012604502               2851
## [7]  {DAI75645} 0.08797428 0.004180064               2736
## [8]  {SNA45677} 0.07893891 0.009678457               2455
## [9]  {FRO31317} 0.07491961 0.011511254               2330
## [10] {DAI85309} 0.07372990 0.006141479               2293
```

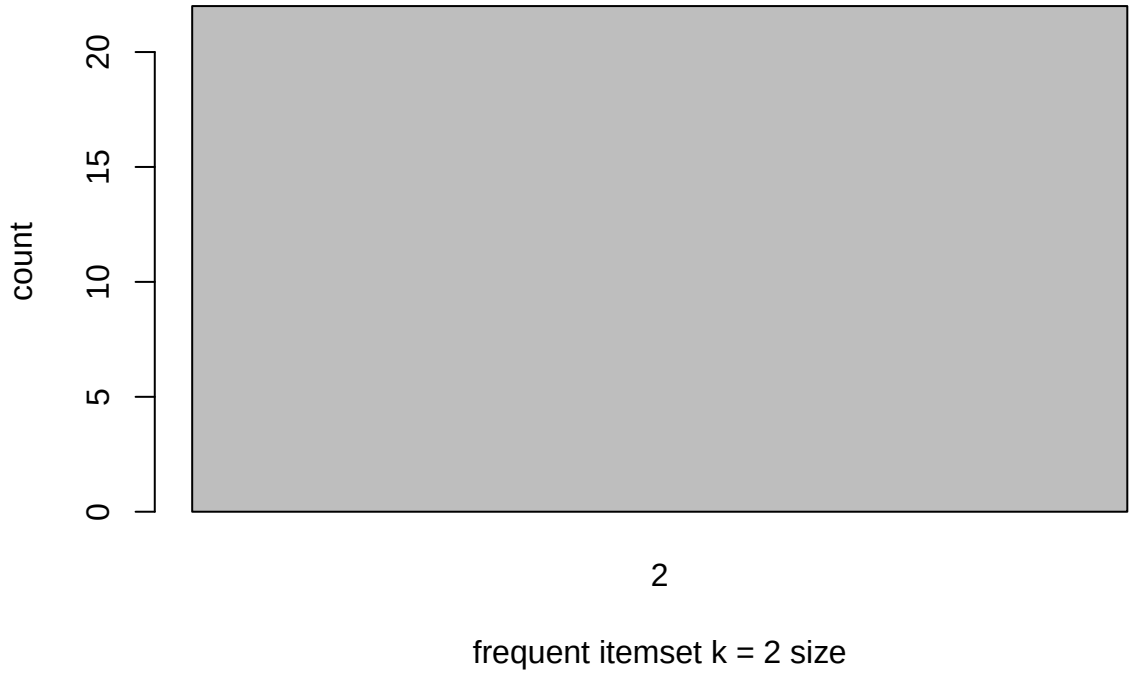
```
is.freq2 <- sort(is.freq2, by = "support")
inspect(head(is.freq2, n = 10))
```

```
##      items               support    transIdenticalToItemsets count
## [1]  {DAI62779,ELE17451} 0.05118971 4.180064e-04             1592
## [2]  {FRO40251,SNA80324} 0.04540193 4.180064e-04             1412
## [3]  {DAI75645,FRO40251} 0.04032154 2.893891e-04             1254
## [4]  {FRO40251,GRO85051} 0.03900322 6.109325e-04             1213
## [5]  {DAI62779,GRO73461} 0.03662379 2.893891e-04             1139
## [6]  {DAI75645,SNA80324} 0.03633441 1.929260e-04             1130
## [7]  {DAI62779,FRO40251} 0.03440514 9.646302e-05             1070
## [8]  {DAI62779,SNA80324} 0.02967846 1.607717e-04              923
## [9]  {DAI62779,DAI85309} 0.02951768 3.215434e-05              918
## [10] {ELE32164,GRO59710} 0.02929260 5.787781e-04              911
```

```
barplot(table(size(is.freq)), xlab = "frequent itemset size", ylab = "count")
```




```
barplot(table(size(is.freq2)), xlab = "frequent itemset k = 2 size", ylab = "count")
```



Câu 4.

Có bao nhiêu closed intemset? Bao nhiêu closed itemset có số item là ít nhất? Bao nhiêu closed itemset có số item là nhiều nhất? Bao nhiêu closed itemset có ít nhất k item trở lên (thử với k = 2, 3, …)? Xem top 10 closed k-itemset sắp xếp theo support (thử với k = 1, 2, 3, …). Vẽ biểu đồ tần số theo bậc (số lượng item) của các closed itemset.

```
is.closed <- is.freq[is.closed(is.freq)]
is.closed
```

```
## set of 29 itemsets
```

```
inspect(head(sort(is.closed, by = "support"), n = 10))
```

```
##      items      support    transIdenticalToItemsets count
## [1]  {DAI62779} 0.21437299 0.025144695              6667
## [2]  {FRO40251} 0.12479100 0.013440514              3881
## [3]  {ELE17451} 0.12456592 0.012861736              3874
## [4]  {GRO73461} 0.11581994 0.006302251              3602
## [5]  {SNA80324} 0.09787781 0.009807074              3044
## [6]  {ELE32164} 0.09167203 0.012604502              2851
## [7]  {DAI75645} 0.08797428 0.004180064              2736
## [8]  {SNA45677} 0.07893891 0.009678457              2455
## [9]  {FRO31317} 0.07491961 0.011511254              2330
## [10] {DAI85309} 0.07372990 0.006141479              2293
```

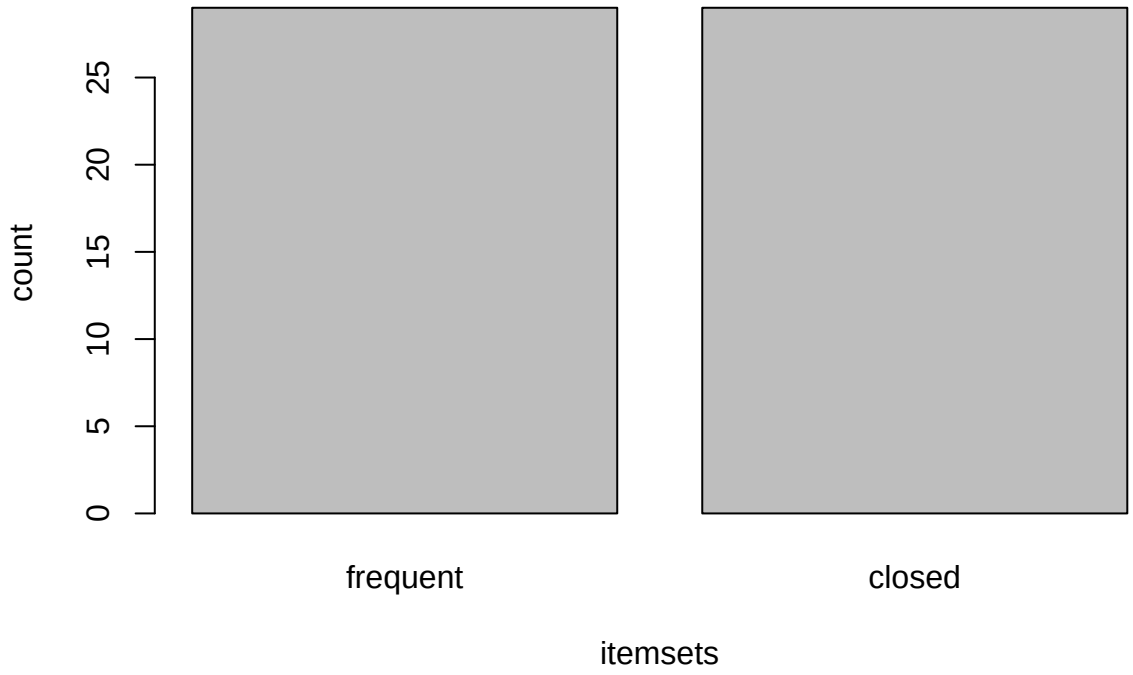
```
is.closed2 <- is.freq2[is.closed(is.freq2)]
is.closed2
```

```
## set of 22 itemsets
```

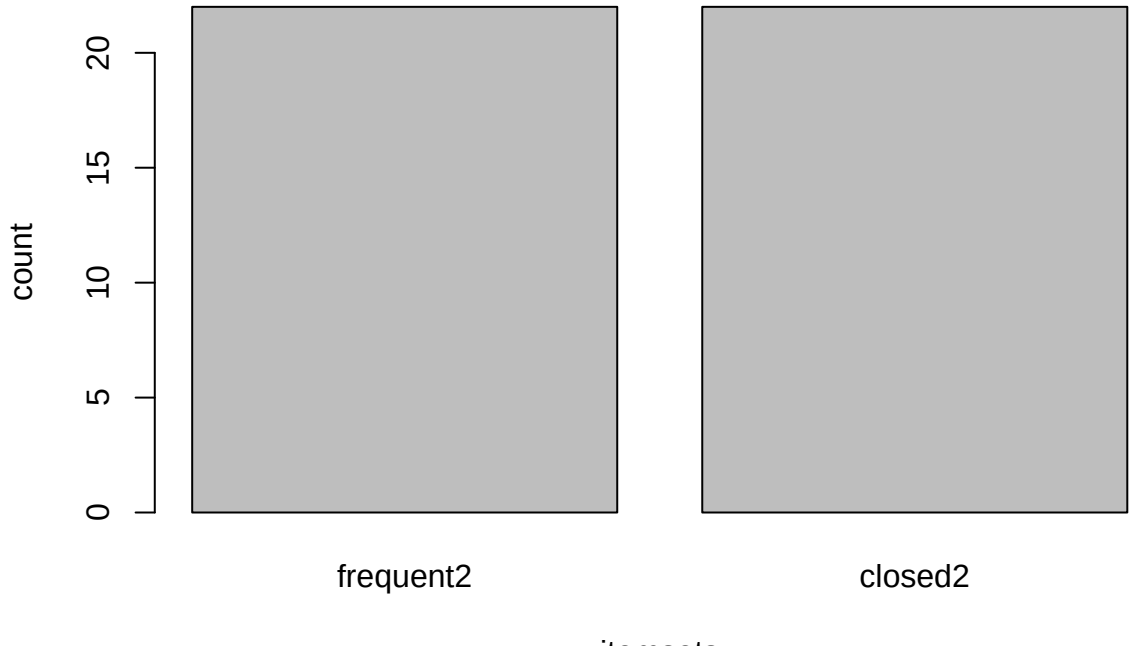
```
inspect(head(sort(is.closed2, by = "support"), n = 10))
```

```
##      items               support    transIdenticalToItemsets count
## [1]  {DAI62779,ELE17451} 0.05118971 4.180064e-04             1592
## [2]  {FRO40251,SNA80324} 0.04540193 4.180064e-04             1412
## [3]  {DAI75645,FRO40251} 0.04032154 2.893891e-04             1254
## [4]  {FRO40251,GRO85051} 0.03900322 6.109325e-04             1213
## [5]  {DAI62779,GRO73461} 0.03662379 2.893891e-04             1139
## [6]  {DAI75645,SNA80324} 0.03633441 1.929260e-04             1130
## [7]  {DAI62779,FRO40251} 0.03440514 9.646302e-05             1070
## [8]  {DAI62779,SNA80324} 0.02967846 1.607717e-04              923
## [9]  {DAI62779,DAI85309} 0.02951768 3.215434e-05              918
## [10] {ELE32164,GRO59710} 0.02929260 5.787781e-04              911
```

```
barplot(c(frequent = length(is.freq),
          closed = length(is.closed)),
        ylab="count", xlab="itemsets")
```




```
barplot(c(frequent2 = length(is.freq2),
          closed2 = length(is.closed2)),
        ylab="count", xlab="itemsets")
```



###

Câu 5. Có bao nhiêu maximal intemset? Bao nhiêu maximal itemset có số item ít nhất? Bao nhiêu maximal itemset có số item là nhiều nhất? Bao nhiêu maximal itemset có ít nhất k item trở lên (thử với k = 2, 3)? Xem top 10 maximal k-itemset sắp xếp theo support (thử với k = 1, 2, 3, …). Vẽ biểu đồ tần số theo bậc (số lượng item) của các maximal itemset.

```
is.max <- is.freq[is.maximal(is.freq)]
is.max
```

```
## set of 27 itemsets
```

```
inspect(head(sort(is.max, by = "support"), n = 10))
```

```
##      items      support    transIdenticalToItemsets count
## [1]  {FRO40251} 0.12479100 0.013440514              3881
## [2]  {GRO73461} 0.11581994 0.006302251              3602
## [3]  {SNA80324} 0.09787781 0.009807074              3044
## [4]  {ELE32164} 0.09167203 0.012604502              2851
## [5]  {DAI75645} 0.08797428 0.004180064              2736
## [6]  {SNA45677} 0.07893891 0.009678457              2455
## [7]  {FRO31317} 0.07491961 0.011511254              2330
## [8]  {DAI85309} 0.07372990 0.006141479              2293
## [9]  {ELE26917} 0.07369775 0.010578778              2292
## [10] {FRO80039} 0.07180064 0.010771704              2233
```

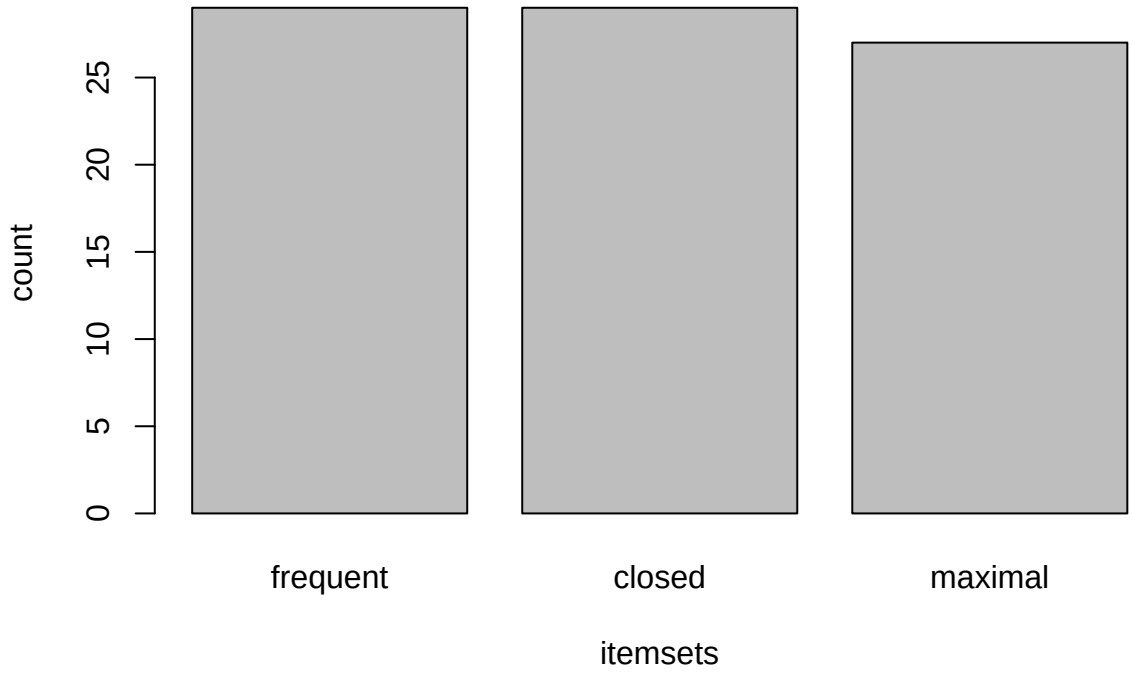
```
is.max2 <- is.freq2[is.maximal(is.freq2)]
is.max2
```

```
## set of 22 itemsets
```

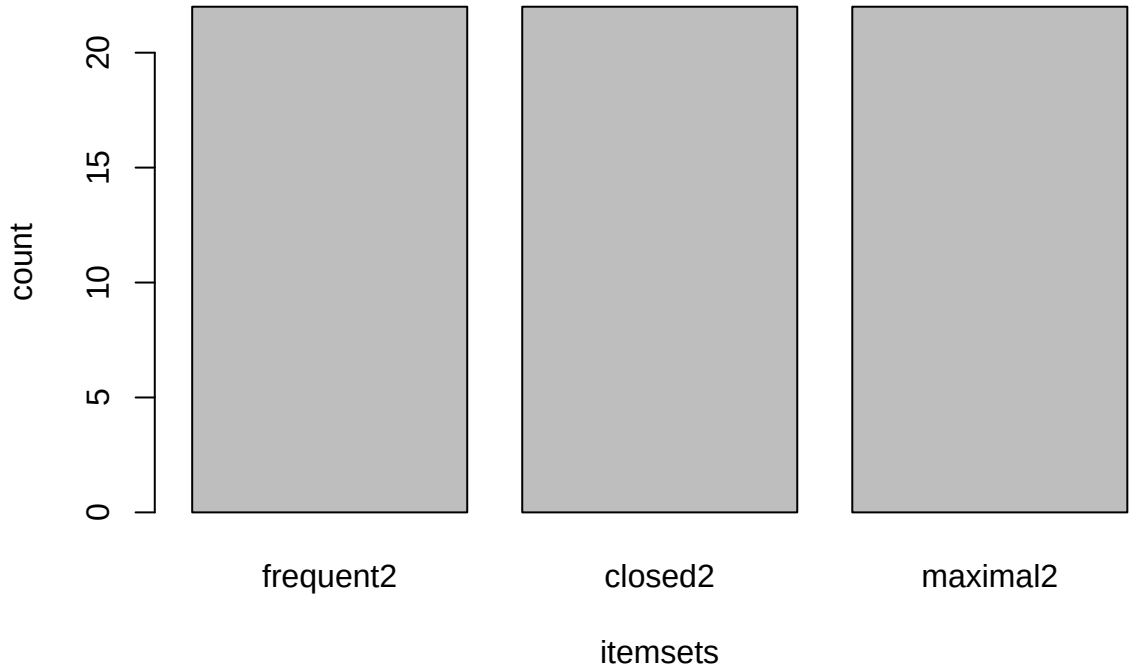
```
inspect(head(sort(is.max2, by = "support"), n = 10))
```

```
##      items               support    transIdenticalToItemsets count
## [1]  {DAI62779,ELE17451} 0.05118971 4.180064e-04             1592
## [2]  {FRO40251,SNA80324} 0.04540193 4.180064e-04             1412
## [3]  {DAI75645,FRO40251} 0.04032154 2.893891e-04             1254
## [4]  {FRO40251,GRO85051} 0.03900322 6.109325e-04             1213
## [5]  {DAI62779,GRO73461} 0.03662379 2.893891e-04             1139
## [6]  {DAI75645,SNA80324} 0.03633441 1.929260e-04             1130
## [7]  {DAI62779,FRO40251} 0.03440514 9.646302e-05             1070
## [8]  {DAI62779,SNA80324} 0.02967846 1.607717e-04              923
## [9]  {DAI62779,DAI85309} 0.02951768 3.215434e-05              918
## [10] {ELE32164,GRO59710} 0.02929260 5.787781e-04              911
```

```
barplot(c(frequent = length(is.freq),
          closed = length(is.closed),
          maximal = length(is.max)),
        ylab="count", xlab="itemsets")
```




```
barplot(c(frequent2 = length(is.freq2),
          closed2 = length(is.closed2),
          maximal2 = length(is.max2)),
        ylab="count", xlab="itemsets")
```



###

Câu 6. Chọn một cặp giá trị support và confidence mà bạn cho là phù hợp với dữ liệu. Tìm tất cả các rule có tối thiểu 2 item ứng với cặp giá trị support và confidence mà bạn chọn. Cho biết thông tin tổng quan về các rule tìm được: Có bao nhiêu rule? Bậc thấp nhất và cao nhất của các rule là bao nhiêu? Support cao (thấp) nhất của các rule? Cho biết top 10 rule sắp xếp theo độ đo lift.

Giả sử mỗi trang được truy cập 12 lần mỗi ngày => tính theo tuần => 12x7/nrow(browsing)

```r
12*7/nrow(browsing)
```

```
## [1] 0.002700965
```

```r
rules <- apriori(browsing,
                 parameter = list(support = 0.003,
                                  confidence = 0.4,
                                  minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5   0.003      2
##  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: 93
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12592 item(s), 31100 transaction(s)] done [0.14s].
## sorting and recoding items ... [688 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 done [0.02s].
## writing ... [520 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```r
rules
```

```
## set of 520 rules
```

```r
summary(rules)
```

```
## set of 520 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4   5
##  76 327  97  20
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   3.000   3.117   3.000   5.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.003023   Min.   :0.4000   Min.   :0.003055   Min.   : 1.868
##  1st Qu.:0.003432   1st Qu.:0.4671   1st Qu.:0.006013   1st Qu.: 3.699
##  Median :0.004180   Median :0.5451   Median :0.007412   Median : 4.974
##  Mean   :0.005793   Mean   :0.6134   Mean   :0.010362   Mean   : 8.563
##  3rd Qu.:0.006053   3rd Qu.:0.7205   3rd Qu.:0.010547   3rd Qu.: 8.013
##  Max.   :0.051190   Max.   :1.0000   Max.   :0.124566   Max.   :67.638
##      count
##  Min.   :  94.0
```

```
##  1st Qu.: 106.8
##  Median : 130.0
##  Mean   : 180.2
##  3rd Qu.: 188.2
##  Max.   :1592.0
##
## mining info:
##      data ntransactions support confidence
##  browsing         31100   0.003        0.4
```

```
inspect(head(rules, n = 10))
```

```
##      lhs           rhs        support     confidence coverage    lift
## [1]  {ELE12951} => {FRO40251} 0.003376206 0.9905660  0.003408360  7.937801
## [2]  {GRO38636} => {FRO40251} 0.003408360 0.9906542  0.003440514  7.938507
## [3]  {ELE55848} => {GRO32086} 0.003376206 0.7094595  0.004758842 25.775922
## [4]  {GRO29598} => {DAI62779} 0.003279743 0.4766355  0.006881029  2.223394
## [5]  {SNA44451} => {DAI18527} 0.003279743 0.5828571  0.005627010 67.637527
## [6]  {DAI37288} => {ELE32164} 0.003762058 0.6464088  0.005819936  7.051321
## [7]  {DAI16732} => {FRO78087} 0.003408360 0.5668449  0.006012862 11.514616
## [8]  {SNA35625} => {DAI62779} 0.003054662 0.4773869  0.006398714  2.226899
## [9]  {DAI53152} => {FRO40251} 0.004501608 0.7179487  0.006270096  5.753209
## [10] {DAI93865} => {FRO40251} 0.006688103 1.0000000  0.006688103  8.013399
##      count
## [1]  105
## [2]  106
## [3]  105
## [4]  102
## [5]  102
## [6]  117
## [7]  106
## [8]   95
## [9]  140
## [10] 208
```

```
rules <- sort(rules, by = "lift")
inspect(head(rules, n = 10))
```

```
##      lhs                             rhs        support     confidence
## [1]  {SNA44451}                   => {DAI18527} 0.003279743 0.5828571
## [2]  {DAI92600,SNA59903}          => {DAI42083} 0.003151125 0.8672566
## [3]  {DAI43868}                   => {SNA82528} 0.009260450 0.9729730
## [4]  {SNA82528}                   => {DAI43868} 0.009260450 0.4848485
## [5]  {FRO17734}                   => {ELE28189} 0.005273312 0.5815603
## [6]  {ELE28189}                   => {FRO17734} 0.005273312 0.4542936
## [7]  {GRO30912}                   => {ELE88583} 0.003376206 0.5000000
## [8]  {DAI62779,FRO19221,SNA93860} => {SNA53220} 0.003504823 0.8720000
## [9]  {GRO89004}                   => {ELE25077} 0.006913183 0.6980519
## [10] {ELE25077}                   => {GRO89004} 0.006913183 0.4497908
##      coverage    lift     count
## [1]  0.005627010 67.63753 102
## [2]  0.003633441 51.66989  98
## [3]  0.009517685 50.94185 288
## [4]  0.019099678 50.94185 288
## [5]  0.009067524 50.10118 164
```

```
## [6]  0.011607717 50.10118 164
## [7]  0.006752412 47.40854 105
## [8]  0.004019293 46.75724 109
## [9]  0.009903537 45.41719 215
## [10] 0.015369775 45.41719 215
```

Câu 7.

Cho biết có bao nhiêu rule mà vế trái có ít nhất k items (thử với k = 2, 3, …)? Vẽ biểu đồ các rule dựa trên số bậc. Vẽ biểu đồ dạng graph của top 50 rule theo độ đo lift.

```
library(arulesViz)
```

```
## Loading required package: grid
```

```
rules2 <- apriori(browsing,
                  parameter = list(support = 0.003,
                                   confidence = 0.4,
                                   minlen = 2))
```

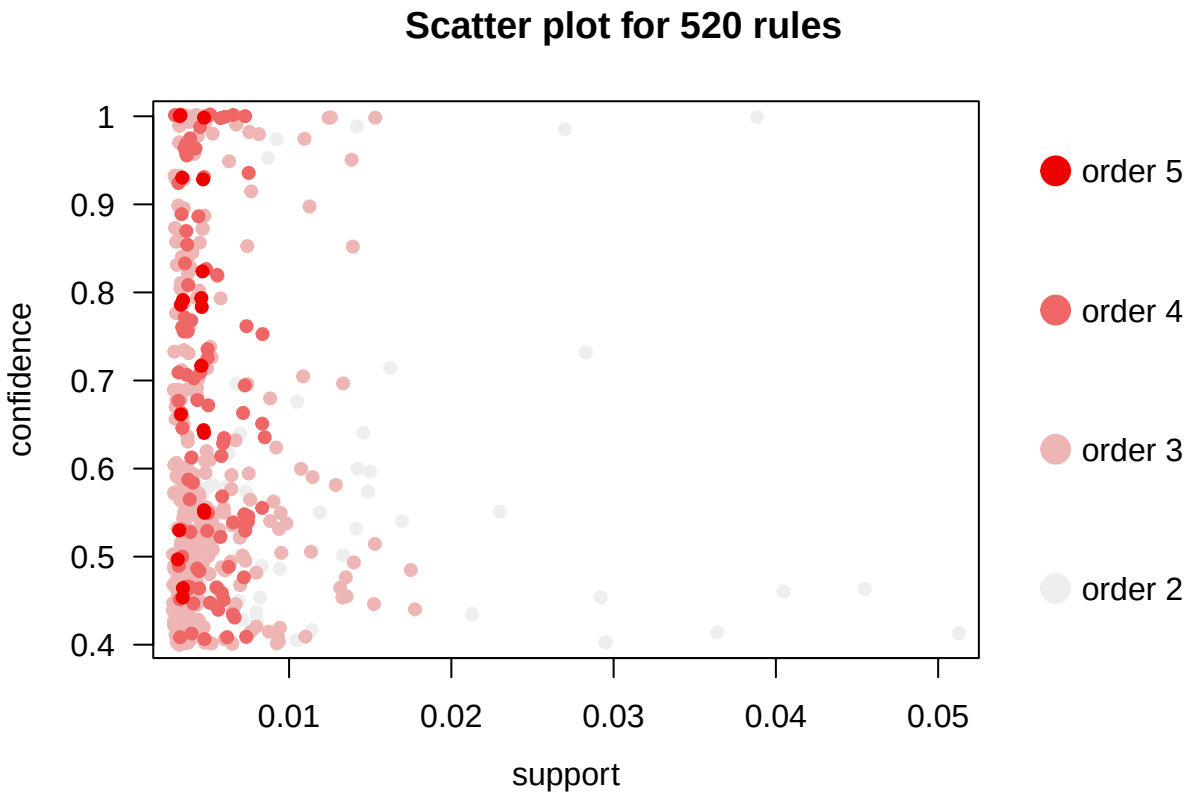
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5   0.003      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 93
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12592 item(s), 31100 transaction(s)] done [0.14s].
## sorting and recoding items ... [688 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 done [0.02s].
## writing ... [520 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
rules2
```

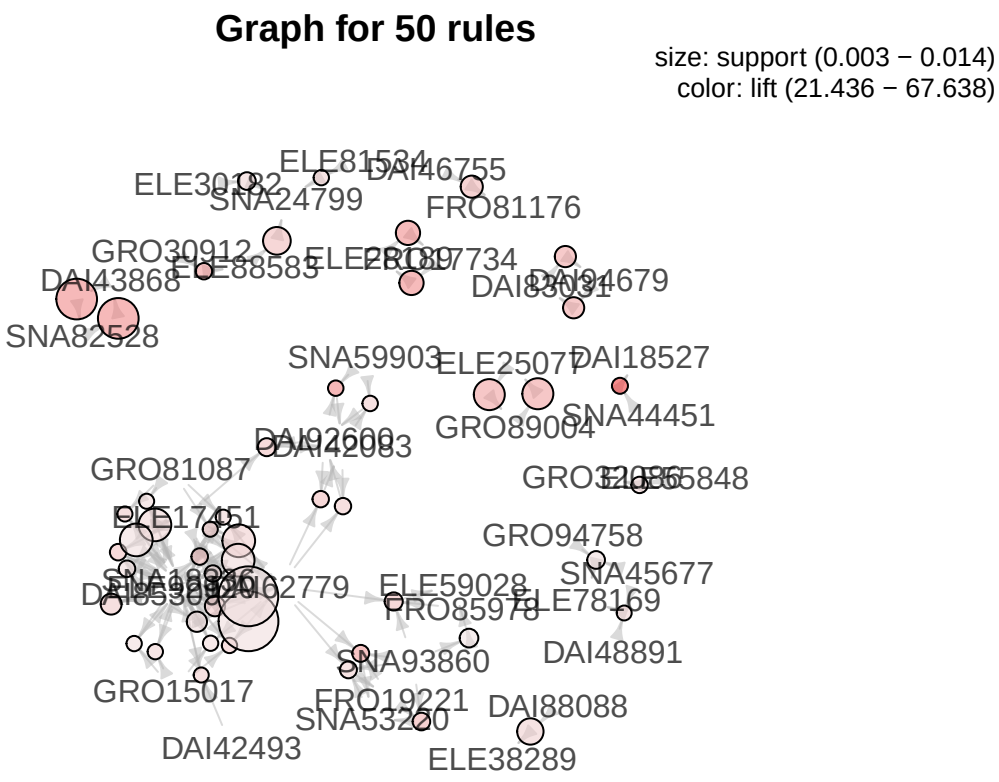
```
## set of 520 rules
```

```
plot(rules2, shading="order")
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```




```
plot(head(sort(rules2, by="lift"), n = 50), method="graph")
```




```
rules3 <- apriori(browsing,
                  parameter = list(support = 0.003,
```

```
                                    confidence = 0.4,
                                    minlen = 3))
```

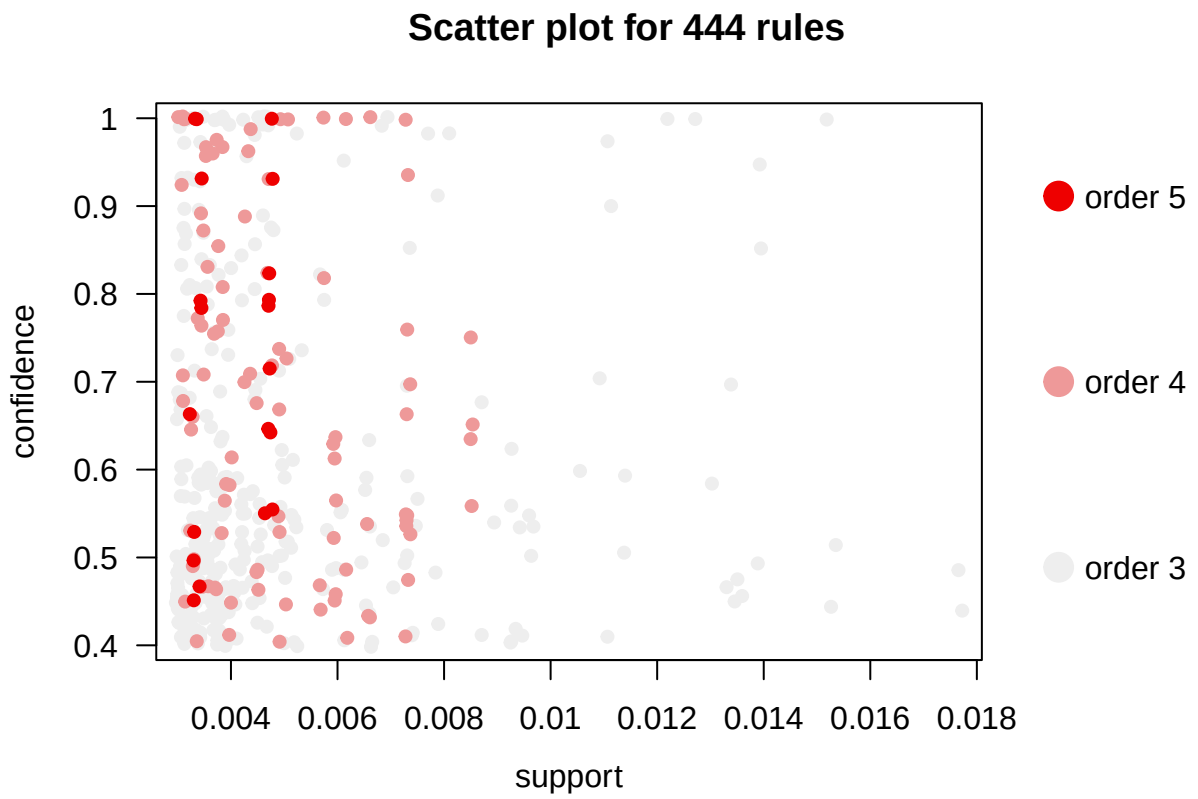
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5   0.003      3
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 93
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12592 item(s), 31100 transaction(s)] done [0.14s].
## sorting and recoding items ... [688 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 done [0.02s].
## writing ... [444 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
rules3
```

```
## set of 444 rules
```

```
plot(rules3, shading="order")
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```




```
plot(head(sort(rules3, by="lift"), n = 50), method="graph")
```