Frequent Pattern Mining - Association Rules

author: IE 2064 Data Science date: April 2016

- packages used
- arules
- arulesViz
- pmml

Introduction

type: section

Why use Frequent Pattern Mining?

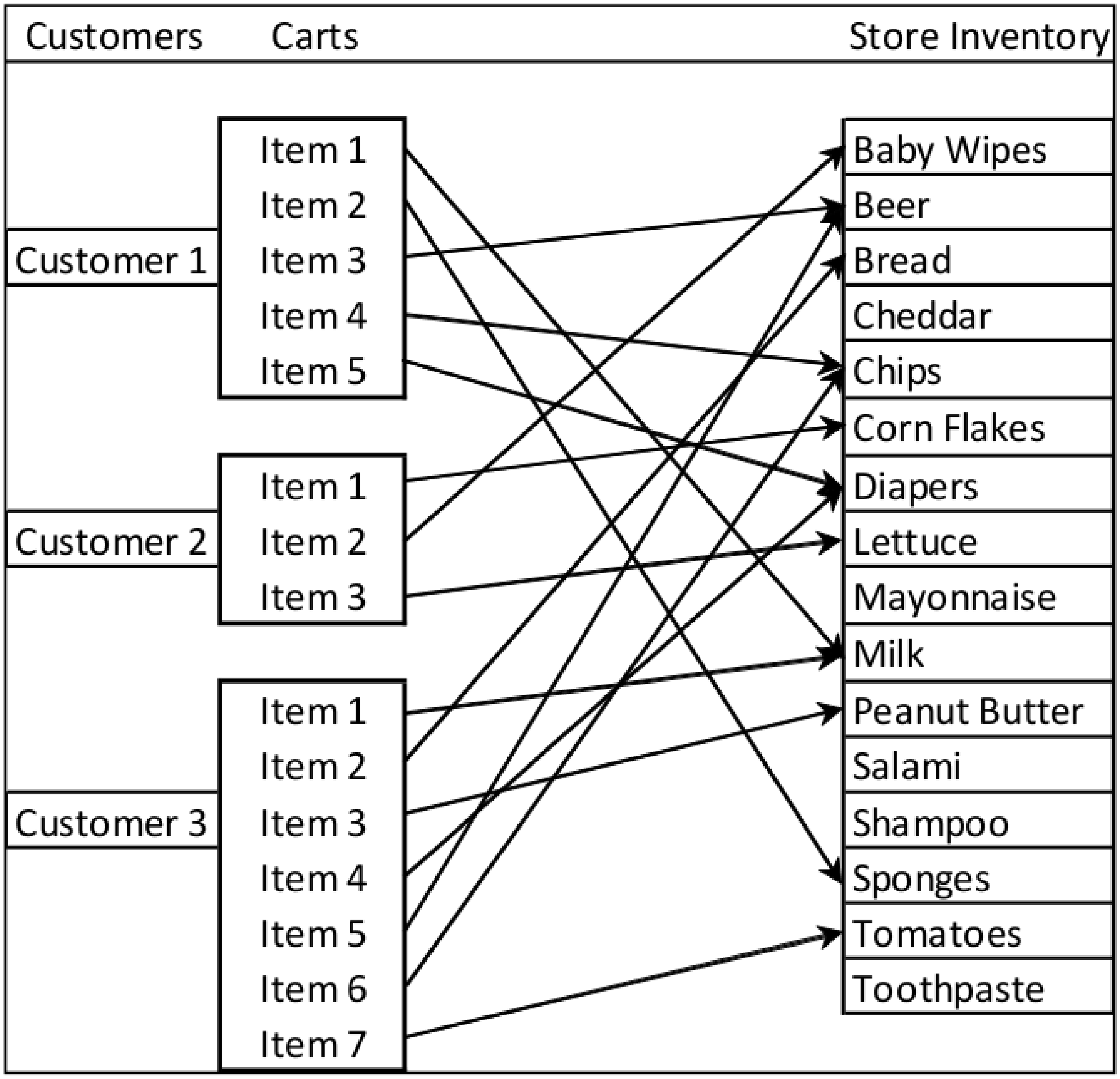
- One common purpose of data mining is to discover novel patterns in the data.
- How can we determine if elements in the data are related?
- *Association Rules* are one example of an *Unsupervised learning* method.
- You have not provided the procedure with examples of what *correct* answers look like, so the method needs to search for candidate correct answers.

Association Rules

type: section

Association Rules Illustrated

left: 40% - Grocery customers and shopping baskets. - What items are commonly bought together? - If you were told an answer, what would you want to know about it?



Affinity analysis

- Consider many possible propositions of combinations (rules).
- Evaluate the database of transactions to evaluate a list of rules for *support*.
- *support* - portion of cases that particular pair appears.
- *confidence* - of the cases where one member appears, the portion of the time where the second member of a pair appears.

Groceries example

Let's look at some data.

```r
library(arules)
data(Groceries)
```

Grocery summary

```r
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
```

Data structure

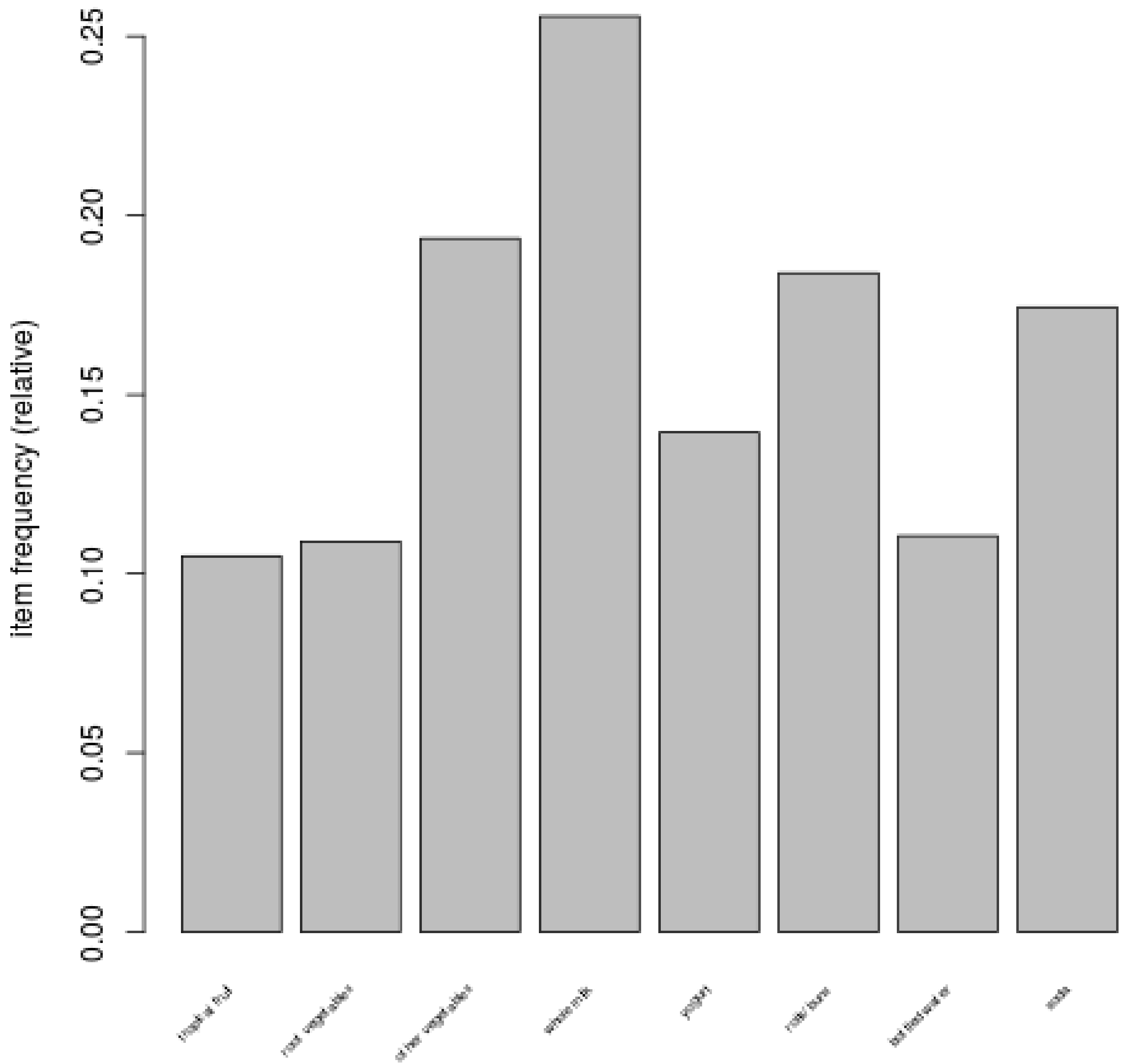
- Sparse matrix
- Which items are most frequent?
- How many items in a cart?

Itemset matrix example

		items			
		i_1	i_2	i_3	i_4
		milk	bread	butter	beer
itemsets	X_1	1	1	0	0
	X_2	0	1	0	1
	X_3	1	1	1	0
	X_4	0	0	1	0

Some common items

```
itemFrequencyPlot(Groceries,support=0.1,cex.names=0.5)
```



Association rules algorithms

- `apriori()`
- `eclat()`
- Parameter sets
- `parameter` changes the characteristics of the ruleset (e.g. *support, confidence, maxlen*)
- `control` influences the performance (e.g. sorting)
- `appearance` - Any restrictions
- Changing parameter values changes the results (size of subsets, number of rules generated tai)

apriori

```r
ruleset1 <-apriori(Groceries,parameter=list(support=0.005, confidence=0.5))
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen
        0.5    0.1    1 none FALSE            TRUE   0.005      1     10
 target   ext
  rules FALSE

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

Absolute minimum support count: 49

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [120 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [120 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Reduce the number of rules

```r
ruleset2 <- apriori(Groceries,parameter=list(support=0.01, confidence=0.5))
```

```
Apriori

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

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

Absolute minimum support count: 98

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [88 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
```

```
writing ... [15 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Look at the result

```r
summary(ruleset2)
```

```
set of 15 rules

rule length distribution (lhs + rhs):sizes
 3 
15 

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

summary of quality measures:
    support          confidence          lift      
 Min.   :0.01007   Min.   :0.5000   Min.   :1.984  
 1st Qu.:0.01174   1st Qu.:0.5151   1st Qu.:2.036  
 Median :0.01230   Median :0.5245   Median :2.203  
 Mean   :0.01316   Mean   :0.5411   Mean   :2.299  
 3rd Qu.:0.01403   3rd Qu.:0.5718   3rd Qu.:2.432  
 Max.   :0.02227   Max.   :0.5862   Max.   :3.030  

mining info:
      data ntransactions support confidence
 Groceries          9835    0.01        0.5
```

lift

- How do you determine how *interesting* a rule is?
- A measure of *support* for a rule.
- Gives increased weight where the Left Hand Side or Right Hand Side occur rarely, but when they do occur, occur together.
- Larger lift is more *interesting*

Take a closer look at the results

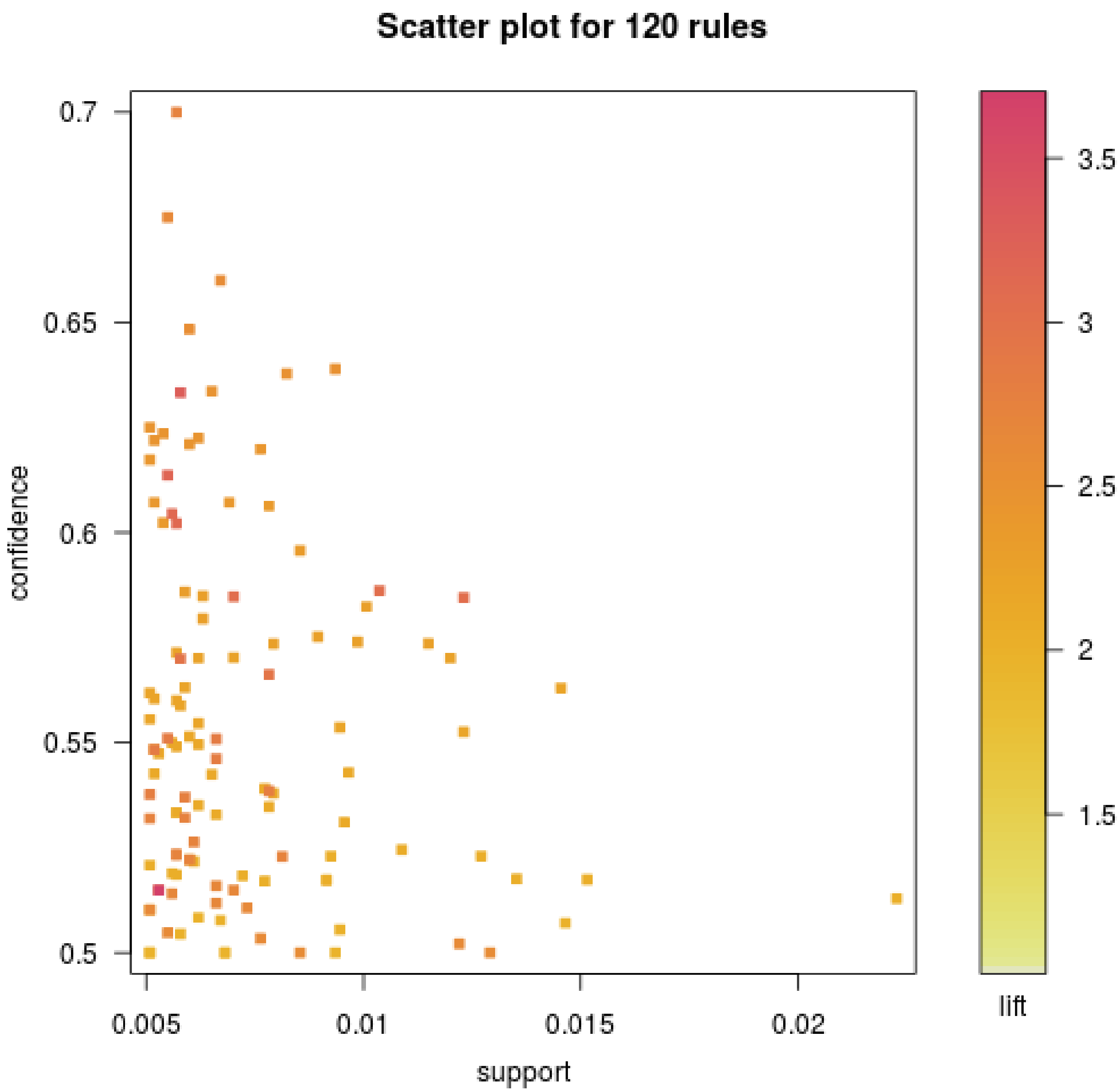
```r
inspect(ruleset2)
```

```
   lhs                     rhs                   support confidence     lift
1  {curd,                                                                   
    yogurt}             => {whole milk}       0.01006609  0.5823529 2.279125
2  {other vegetables,                                                       
    butter}             => {whole milk}       0.01148958  0.5736041 2.244885
```

```
3  {other vegetables,
    domestic eggs}      => {whole milk}       0.01230300  0.5525114 2.162336
4  {yogurt,
    whipped/sour cream} => {whole milk}       0.01087951  0.5245098 2.052747
5  {other vegetables,
    whipped/sour cream} => {whole milk}       0.01464159  0.5070423 1.984385
6  {pip fruit,
    other vegetables}   => {whole milk}       0.01352313  0.5175097 2.025351
7  {citrus fruit,
    root vegetables}    => {other vegetables} 0.01037112  0.5862069 3.029608
8  {tropical fruit,
    root vegetables}    => {other vegetables} 0.01230300  0.5845411 3.020999
9  {tropical fruit,
    root vegetables}    => {whole milk}       0.01199797  0.5700483 2.230969
10 {tropical fruit,
    yogurt}             => {whole milk}       0.01514997  0.5173611 2.024770
11 {root vegetables,
    yogurt}             => {other vegetables} 0.01291307  0.5000000 2.584078
12 {root vegetables,
    yogurt}             => {whole milk}       0.01453991  0.5629921 2.203354
13 {root vegetables,
    rolls/buns}         => {other vegetables} 0.01220132  0.5020921 2.594890
14 {root vegetables,
    rolls/buns}         => {whole milk}       0.01270971  0.5230126 2.046888
15 {other vegetables,
    yogurt}             => {whole milk}       0.02226741  0.5128806 2.007235
```

Now for a visual inspection of results

```r
library(arulesViz)
plot(ruleset1)
```



See if there are any good rules from the larger set

```
goodrules <- ruleset1[quality(ruleset1)$lift > 3.0]
inspect(goodrules)
```

```
  lhs                        rhs                    support confidence     lift
1 {root vegetables,
   onions}                => {other vegetables} 0.005693950  0.6021505 3.112008
2 {tropical fruit,
   curd}                  => {yogurt}           0.005287239  0.5148515 3.690645
3 {pip fruit,
```

```
   whipped/sour cream} => {other vegetables} 0.005592272  0.6043956 3.123610
4 {citrus fruit,
   root vegetables}    => {other vegetables} 0.010371124  0.5862069 3.029608
5 {tropical fruit,
   root vegetables}    => {other vegetables} 0.012302999  0.5845411 3.020999
6 {pip fruit,
   root vegetables,
   whole milk}         => {other vegetables} 0.005490595  0.6136364 3.171368
7 {citrus fruit,
   root vegetables,
   whole milk}         => {other vegetables} 0.005795628  0.6333333 3.273165
8 {tropical fruit,
   root vegetables,
   whole milk}         => {other vegetables} 0.007015760  0.5847458 3.022057
```

Data preparation example

type:section

Epub downloads

Electronic book downloads from Vienna University of Economics

```r
data(Epub)
Epub
```

```
transactions in sparse format with
 15729 transactions (rows) and
 936 items (columns)
```

Get more information

```r
summary(Epub)
```

```
transactions as itemMatrix in sparse format with
 15729 rows (elements/itemsets/transactions) and
 936 columns (items) and a density of 0.001758755

most frequent items:
doc_11d doc_813 doc_4c6 doc_955 doc_698 (Other)
    356     329     288     282     245   24393

element (itemset/transaction) length distribution:
sizes
    1     2     3     4     5     6     7     8     9    10    11    12
11615  2189   854   409   198   121    93    50    42    34    26    12
   13    14    15    16    17    18    19    20    21    22    23    24
```

```
   10    10     6     8     6     5     8     2     2     3     2     3
   25    26    27    28    30    34    36    38    41    43    52    58
    4     5     1     1     1     2     1     2     1     1     1     1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.646   2.000  58.000

includes extended item information - examples:
   labels
1 doc_11d
2 doc_13d
3 doc_14c

includes extended transaction information - examples:
      transactionID           TimeStamp
10792  session_4795 2003-01-01 20:59:00
10793  session_4797 2003-01-02 07:46:01
10794  session_479a 2003-01-02 10:50:38
```

See how it changes over time

```r
year <- strftime(as.POSIXlt(transactionInfo(Epub)[["TimeStamp"]]), "%Y")
table(year)
```

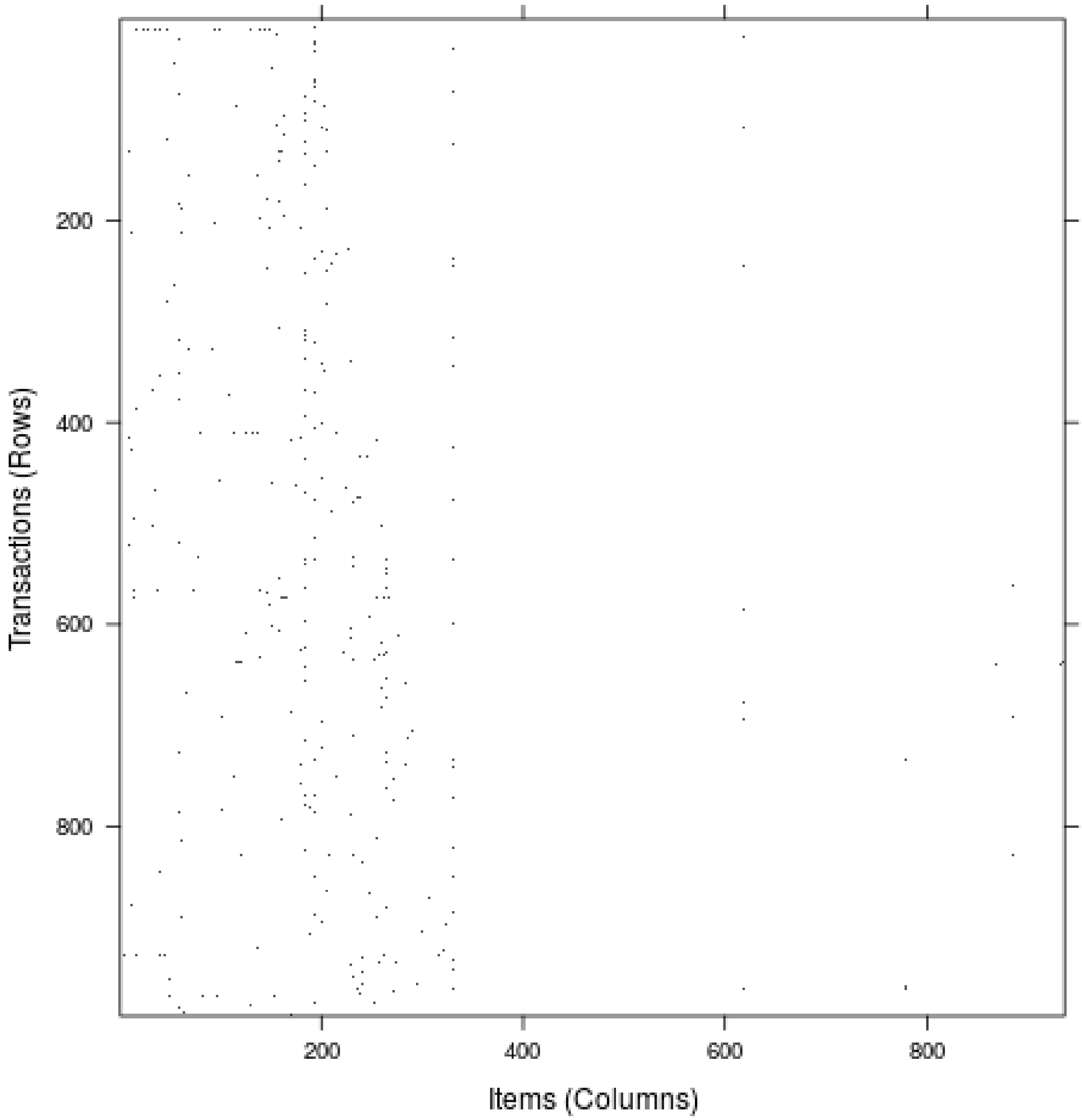
```
year
2003 2004 2005 2006 2007 2008
 987 1375 1611 3015 4050 4691
```

Look at one years worth of downloads

```r
epub2003 <- Epub[year=="2003"]
length(epub2003)
```

```
[1] 987
```

```r
image(epub2003)
```



Let's look at only long transactions

```
transactionInfo(epub2003[size(epub2003) > 20])
```

```
      transactionID           TimeStamp
11092  session_56e2 2003-04-29 13:30:38
11371  session_6308 2003-08-17 18:16:12
```

Let's take a closer look

```r
inspect(epub2003[1:5])
```

```
      items                      transactionID TimeStamp
10792 {doc_154}                  session_4795  2003-01-01 20:59:00
10793 {doc_3d6}                  session_4797  2003-01-02 07:46:01
10794 {doc_16f}                  session_479a  2003-01-02 10:50:38
10795 {doc_11d,doc_1a7,doc_f4} session_47b7  2003-01-02 18:55:50
10796 {doc_83}                   session_47bb  2003-01-02 21:27:44
```

What if I want transactions per document

Coerce into a vertical layout with transaction ID list for each document.

```r
epubTidLists <- as(Epub, "tidLists")
as(epubTidLists[5], 'list')
```

```
$doc_150
 [1] "session_56e2"  "session_575d"  "session_7090"  "session_80ef"
 [5] "session_9b5a"  "session_bf41"  "session_112a9" "session_11e26"
 [9] "session_123bc" "session_12938" "session_12a5e" "session_14ae7"
[13] "session_15e17" "session_161ca" "session_177cf" "session_18649"
[17] "session_18a83" "session_190bf" "session_19152" "session_19c27"
[21] "session_1a264" "session_1c2e6" "session_1e935" "session_20955"
[25] "session_23fe8"
```

Questionnaire data example

- Source: 1994 U.S. Census
- 48842 records
- Filtered so that `AAGE>16` and `AGI>100`
- Adults with non-zero income
- Can we determine if

```r
data("AdultUCI")
dim(AdultUCI)
```

```
[1] 48842    15
```

Data summary

```r
summary(AdultUCI)
```

```
     age                    workclass        fnlwgt       
 Min.   :17.00   Private         :33906   Min.   :  12285  
 1st Qu.:28.00   Self-emp-not-inc: 3862   1st Qu.: 117550  
 Median :37.00   Local-gov       : 3136   Median : 178144  
 Mean   :38.64   State-gov       : 1981   Mean   : 189664  
 3rd Qu.:48.00   Self-emp-inc    : 1695   3rd Qu.: 237642  
 Max.   :90.00   (Other)         : 1463   Max.   :1490400  
                 NA's            : 2799                    
        education     education-num                 marital-status 
 HS-grad     :15784   Min.   : 1.00   Divorced             : 6633  
 Some-college:10878   1st Qu.: 9.00   Married-AF-spouse    :   37  
 Bachelors   : 8025   Median :10.00   Married-civ-spouse   :22379  
 Masters     : 2657   Mean   :10.08   Married-spouse-absent:  628  
 Assoc-voc   : 2061   3rd Qu.:12.00   Never-married        :16117  
 11th        : 1812   Max.   :16.00   Separated            : 1530  
 (Other)     : 7625                   Widowed              : 1518  
           occupation          relationship                   race      
 Prof-specialty : 6172   Husband       :19716   Amer-Indian-Eskimo:  470  
 Craft-repair   : 6112   Not-in-family :12583   Asian-Pac-Islander: 1519  
 Exec-managerial: 6086   Other-relative: 1506   Black             : 4685  
 Adm-clerical   : 5611   Own-child     : 7581   Other             :  406  
 Sales          : 5504   Unmarried     : 5125   White             :41762  
 (Other)        :16548   Wife          : 2331                             
 NA's           : 2809                                                    
     sex         capital-gain    capital-loss    hours-per-week 
 Female:16192   Min.   :    0   Min.   :   0.0   Min.   : 1.00  
 Male  :32650   1st Qu.:    0   1st Qu.:   0.0   1st Qu.:40.00  
                Median :    0   Median :   0.0   Median :40.00  
                Mean   : 1079   Mean   :  87.5   Mean   :40.42  
                3rd Qu.:    0   3rd Qu.:   0.0   3rd Qu.:45.00  
                Max.   :99999   Max.   :4356.0   Max.   :99.00  
                                                                
       native-country    income     
 United-States:43832   small:24720  
 Mexico       :  951   large: 7841  
 Philippines  :  295   NA's :16281  
 Germany      :  206                
 Puerto-Rico  :  184                
 (Other)      : 2517                
 NA's         :  857                
```

Take a closer look

```r
AdultUCI[1:2,]
```

```
  age        workclass fnlwgt education education-num     marital-status
1  39        State-gov  77516 Bachelors            13      Never-married
2  50 Self-emp-not-inc  83311 Bachelors            13 Married-civ-spouse
       occupation  relationship  race  sex capital-gain capital-loss
1    Adm-clerical Not-in-family White Male         2174            0
2 Exec-managerial       Husband White Male            0            0
```

```
  hours-per-week native-country income
1             40  United-States  small
2             13  United-States  small
```

Clean data

Remove a weighting calculation and a duplicate education factor

```
AdultUCI[["fnlwgt"]] <- NULL
AdultUCI[["education-num"]] <- NULL
```

Map some other values to categorical variables

```
AdultUCI[[ "age"]] <- ordered(cut(AdultUCI[[ "age"]],                         c(15,25,45,65,100)), labe
AdultUCI[[ "hours-per-week"]] <- ordered(cut(AdultUCI[["hours-per-week"]], c(0,25,40,60,168)),labels = c
AdultUCI[[ "capital-gain"]] <- ordered(cut(AdultUCI[[
"capital-gain"]],  c(-Inf,0,median(AdultUCI[[ "capital-gain"]]                    [AdultUCI[[ "capital-gai
AdultUCI[[ "capital-loss"]] <- ordered(cut(AdultUCI[["capital-loss"]],  c(-Inf,0, median(AdultUCI[[ "cap
```

Convert to a binary incidence matrix through coercion to transactions

```
Adult <- as(AdultUCI, "transactions")
Adult
```

```
transactions in sparse format with
 48842 transactions (rows) and
 115 items (columns)
```

See what we have

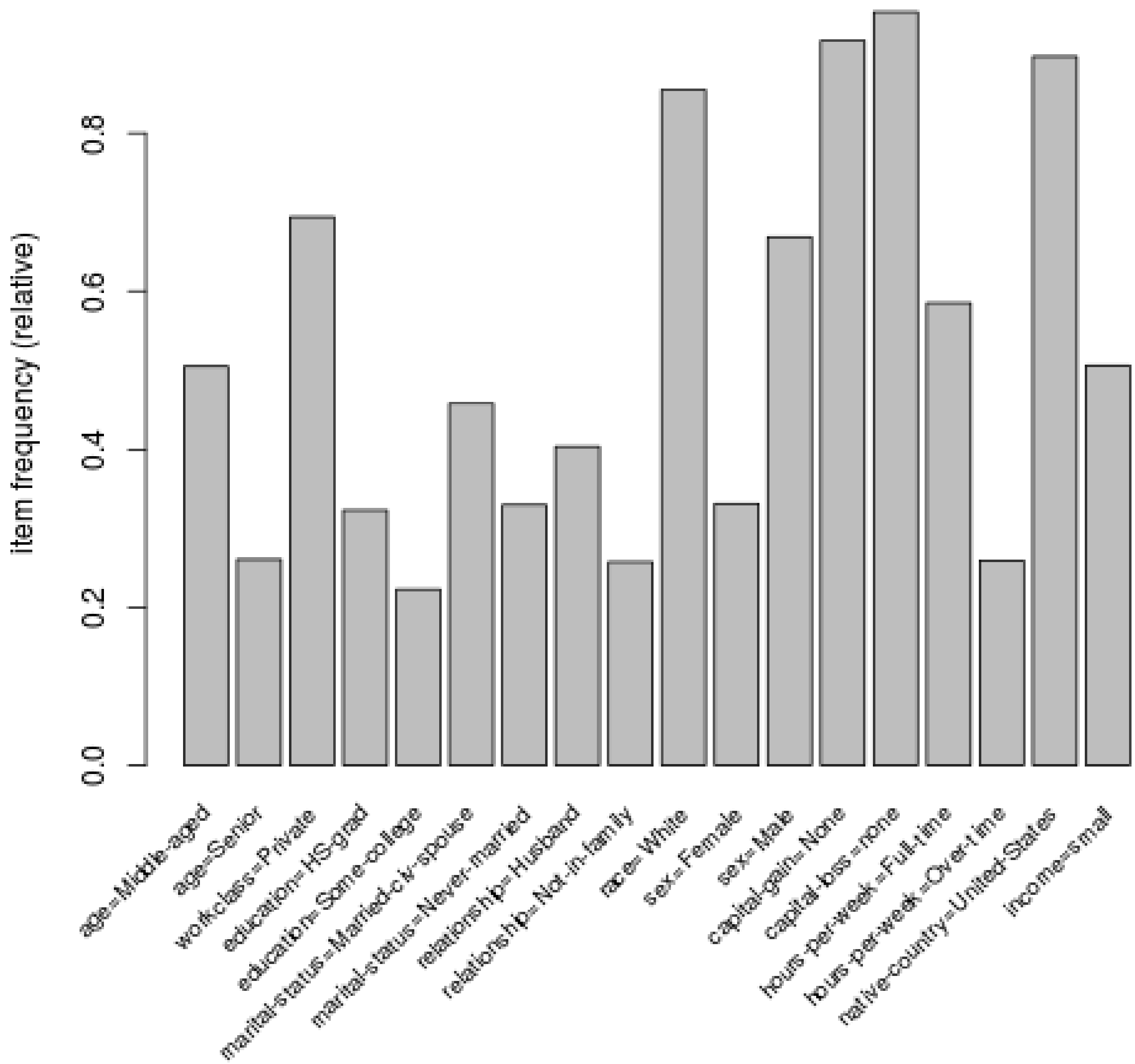
```
summary(Adult)
```

```
transactions as itemMatrix in sparse format with
 48842 rows (elements/itemsets/transactions) and
 115 columns (items) and a density of 0.1089939

most frequent items:
           capital-loss=none           capital-gain=None
                       46560                       44807
native-country=United-States                  race=White
                       43832                       41762
```

```
           workclass=Private                      (Other)
                       33906                       401333

element (itemset/transaction) length distribution:
sizes
    9    10    11    12    13
   19   971  2067 15623 30162

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   9.00   12.00   13.00   12.53   13.00   13.00

includes extended item information - examples:
           labels variables      levels
1       age=Young       age       Young
2 age=Middle-aged       age Middle-aged
3      age=Senior       age      Senior

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

Now plot the Item Frequency Plot

```r
itemFrequencyPlot(Adult, support = 0.2, cex.names=0.8)
```



Generate some rules

```
rules <- apriori(Adult,
                 parameter =
                   list(support = 0.01, confidence = 0.6))
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen
        0.6    0.1    1 none FALSE            TRUE    0.01      1     10
```

```
 target   ext
  rules FALSE

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

Absolute minimum support count: 488

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[115 item(s), 48842 transaction(s)] done [0.04s].
sorting and recoding items ... [67 item(s)] done [0.01s].
creating transaction tree ... done [0.04s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [1.20s].
writing ... [276443 rule(s)] done [0.04s].
creating S4 object  ... done [0.22s].
```

Summarize the rules

```
summary(rules)
```

```
set of 276443 rules

rule length distribution (lhs + rhs):sizes
    1     2     3     4     5     6     7     8     9    10
    6   432  4981 22127 52669 75104 67198 38094 13244  2588

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   5.000   6.000   6.289   7.000  10.000

summary of quality measures:
    support           confidence          lift
 Min.   :0.01001   Min.   :0.6000   Min.   : 0.7171
 1st Qu.:0.01253   1st Qu.:0.7691   1st Qu.: 1.0100
 Median :0.01701   Median :0.9051   Median : 1.0554
 Mean   :0.02679   Mean   :0.8600   Mean   : 1.3109
 3rd Qu.:0.02741   3rd Qu.:0.9542   3rd Qu.: 1.2980
 Max.   :0.95328   Max.   :1.0000   Max.   :20.6826

mining info:
  data ntransactions support confidence
 Adult         48842    0.01        0.6
```

Break data into subset, and limit the number of rules

- Create rules for both ‘income-small’ and ‘income-large’
- Limit the number of rules by specifying a minimum lift.

```r
rulesIncomeSmall <- subset(rules, subset =
            rhs %in% "income=small" & lift > 1.2)
rulesIncomeLarge <- subset(rules, subset =
            rhs %in% "income=large" & lift > 1.2)
```

Inspect the best rules

```r
inspect(head(sort(rulesIncomeSmall, by = "confidence"),
             n = 3))
```

```
  lhs                                 rhs              support  confidence     lift
1 {workclass=Private,
   marital-status=Never-married,
   relationship=Own-child,
   sex=Male,
   hours-per-week=Part-time,
   native-country=United-States} => {income=small} 0.01074895  0.7104195 1.403653
2 {workclass=Private,
   marital-status=Never-married,
   relationship=Own-child,
   sex=Male,
   hours-per-week=Part-time}     => {income=small} 0.01144507  0.7102922 1.403402
3 {workclass=Private,
   marital-status=Never-married,
   relationship=Own-child,
   sex=Male,
   capital-gain=None,
   hours-per-week=Part-time,
   native-country=United-States} => {income=small} 0.01046231  0.7097222 1.402276
```

Inspect when income large

```r
inspect(head(sort(rulesIncomeLarge, by = "confidence"),
             n = 3))
```

```
  lhs                                      rhs              support  confidence     lift
1 {marital-status=Married-civ-spouse,
   capital-gain=High,
   native-country=United-States}      => {income=large} 0.01562180  0.6849192 4.266398
2 {marital-status=Married-civ-spouse,
   capital-gain=High,
   capital-loss=none,
   native-country=United-States}      => {income=large} 0.01562180  0.6849192 4.266398
3 {relationship=Husband,
   race=White,
   capital-gain=High,
   native-country=United-States}      => {income=large} 0.01302158  0.6846071 4.264454
```

Save the rules

Save these rules using PMML for use in other systems.

```
library(pmml)
write.PMML(rulesIncomeSmall, file = "incomerulessmall.xml")
```