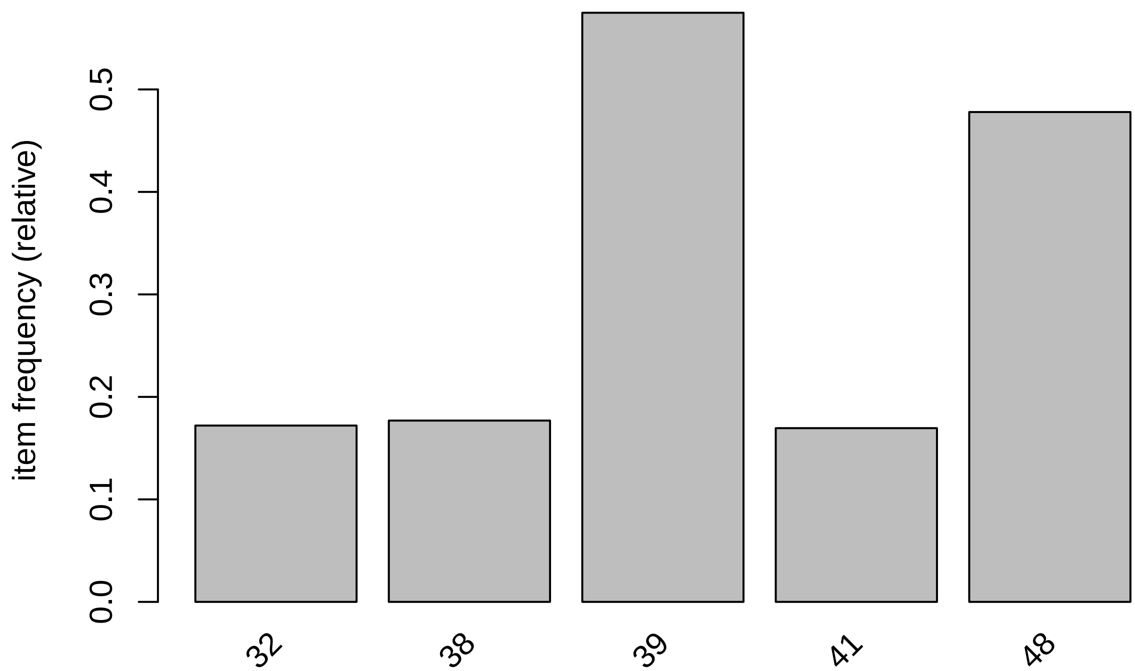
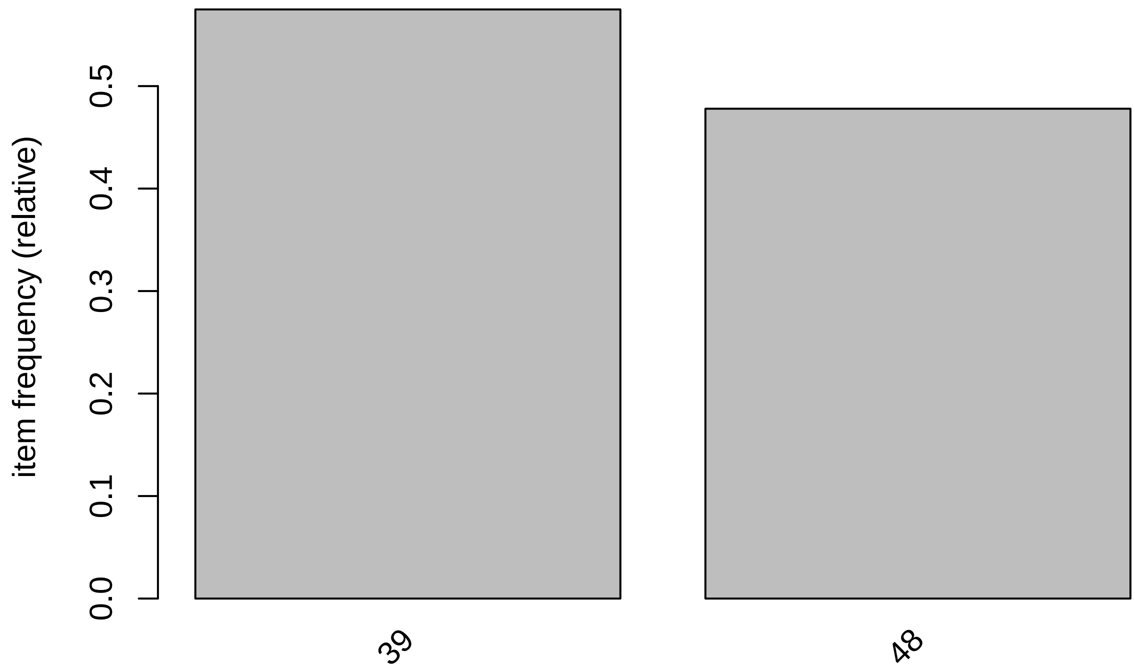
Assignment 4

Nelson Corrocher

December 4, 2016

Part 1





item frequency (relative)
0.0
0.1
0.2
0.3
0.4
0.5
39
48


```
## transactions as itemMatrix in sparse format with
##  88162 rows (elements/itemsets/transactions) and
##  16470 columns (items) and a density of 0.00063
##
## most frequent items:
##      39      48      38      32      41 (Other)
##   50675   42135   15596   15167   14945  770058
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 3016 5516 6919 7210 6814 6163 5746 5143 4660 4086 3751 3285 2866 2620 2310
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
## 2115 1874 1645 1469 1290 1205  981  887  819  684  586  582  472  480  355
##   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45
##  310  303  272  234  194  136  153  123  115  112   76   66   71   60   50
##   46   47   48   49   50   51   52   53   54   55   56   57   58   59   60
##   44   37   37   33   22   24   21   21   10   11   10    9   11    4    9
##   61   62   63   64   65   66   67   68   71   73   74   76
##    7    4    5    2    2    5    3    3    1    1    1    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       4       8      10      14      76
##
## includes extended item information - examples:
##   labels
```

```
## 1      0
## 2      1
## 3     10
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 4408
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16470 item(s), 88162 transaction(s)] done [0.09s].
## sorting and recoding items ... [6 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##      lhs        rhs  support confidence lift
## [1]  {41,48} => {39} 0.084   0.82       1.42
## [2]  {38,48} => {39} 0.069   0.77       1.34
## [3]  {41}    => {39} 0.129   0.76       1.33
## [4]  {48}    => {39} 0.331   0.69       1.20
## [5]  {32,48} => {39} 0.061   0.67       1.17
## [6]  {38}    => {39} 0.117   0.66       1.15
## [7]  {39,41} => {48} 0.084   0.65       1.35
## [8]  {32,39} => {48} 0.061   0.64       1.34
## [9]  {41}    => {48} 0.102   0.60       1.26
## [10] {38,39} => {48} 0.069   0.59       1.23
## [11] {39}    => {48} 0.331   0.58       1.20
## [12] {}      => {39} 0.575   0.57       1.00
## [13] {32}    => {39} 0.096   0.56       0.97
## [14] {32}    => {48} 0.091   0.53       1.11
## [15] {38}    => {48} 0.090   0.51       1.07
```

Part 2

Some notable rules are: the one with highest confidence is product 39 with basket of 41 and 48. The two-items combo that seems be most related is of products 39 and 48. One interesting results is that products 32 and 39 had a lift below 1.00, which could imply they are inversely associated (the appear together less than what would be expected if they were independent).

Part 3

The next steps from here would depend on the goal. For example, were I working for a big retailer trying to improve the product location, one step would be to use the support and confidence to better relocate SKUs (stock keeping units) to different places in in the store like putting associated SKUs in both ends of the same shelf. Knowing the customers would likely walk from one end to the other, products placed in the middle of the shelf would have more assured visibility, which could be used to place a third associated SKU with higher margin at that location.

In the case the retailer had different format stores, for example, a smaller one located in the train station, it could be an option to add virtual items using customers' information to basket to look for ways to improve the product assortment at the location (makes sense since the public who goes to the regular retail stores are not the same or don't have the same goals as the ones who shop quickly at a train station).

Using association analysis in conjunction with multivariate regressions, it could be possible to estimate the price of a SKU that would maximize profits (if the price is higher than the optimal, volume will decrease faster revenue increase; if price is lower, revenue would grow but the profit margin decreases faster).

Another possible use would be in healthcare. In many cases, patients getting one treatment will need complimentary treatments depending on their age group, ethnicity or gender. Adding these variables as virtual items could bring new information about services usage, reducing asymetric information between patient and the insurance company and thus, leading to more accurate charged insurance premiums.