# Homework4

*Ivan Ojiambo*

*October 12, 2017*

## Exe1

```
## Warning: package 'arules' was built under R version 3.4.2

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.4.2
```

### 1a Calculate the support and support count of patterns {D}, {D,F} and {D,F,G}

support({D}) = 11/15 support count {D} = 11

support( {D,F}) = 8/15 support count ({D,F}) = = 8

support({D,F,G}) = 6/15 support count ( {D,F,G}) = 6

### (1b) Report the row indices (identifiers) of transactions which include the pattern {D,F,G}

- 13, 10, 9, 8, 3, 2

### (1c) Explain what anti-monotonicity of support means, in the example of these patterns {D}, {D,F} and {D,F,G}

- {D} is subset of {D,F} and {D,F,G}
- since support({D}) > support({D,F}) > support({D,F,G})
- Therefore its anti-monotonic

```
## Eclat
##
## parameter specification:
##  tidLists   support minlen maxlen            target   ext
##     FALSE 0.3333333      1     10 frequent itemsets FALSE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 5
##
## create itemset ...
## set transactions ...[8 item(s), 15 transaction(s)] done [0.00s].
## sorting and recoding items ... [6 item(s)] done [0.00s].
## creating bit matrix ... [6 row(s), 15 column(s)] done [0.00s].
## writing  ... [17 set(s)] done [0.00s].
```

```
## Creating S4 object  ... done [0.00s].
```

### (1d) How many itemsets could be generated in total from 8 items?

```r
#number of itemsets = 2^d
  number_itemset <- 2^8
  number_itemset
```

```
## [1] 256
```

```r
##1e) What percentage of these itemsets have positive support (occur at least once in the data)?
#Use find_freq_itemsets(data,1) to find it out.

count <- nrow(find_freq_itemsets(data,1))
```

```
## Eclat
##
## parameter specification:
##  tidLists    support minlen maxlen            target   ext
##     FALSE 0.06666667      1     10 frequent itemsets FALSE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 1
```

```
## Warning in eclat(data, parameter = list(support = min_support)): You chose a very low absolute suppo
```

```
## create itemset ...
## set transactions ...[8 item(s), 15 transaction(s)] done [0.00s].
## sorting and recoding items ... [8 item(s)] done [0.00s].
## creating bit matrix ... [8 row(s), 15 column(s)] done [0.00s].
## writing  ... [96 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```r
percentage <- (count/number_itemset)*100
percentage
```

```
## [1] 37.5
```

```r
#(1f) Naive method would have to look through all possible subsets of size 3. How many subsets of size
choose(8,3)
```

```
## [1] 56
```

### 1g

```r
#{A,C,D}=4, {A,C,F}=3, {A,C,G}=2, {A,C,H}=2, {A,D,F}=4 , {A,D,G}=3, {A,D,H}=3,
#{A,F,G}=4, {A,F,H}=2 {C,D,F}=3, {C,D,G}=2 {C,D,H}=2, {C,F,G}=4, {C,F,H}=2 , {D,F,G}=6, {D,H,F}=4, {D,H
```

**h Study the 3-sets reported in (1g) and discard all the 3-sets for which some subset of size 2 is not frequent. Report the remaining candidate 3-sets.**

```
#{A,C,D}=4,{A,D,F}=4, {A,C,F}=3,  {C,D,F}=3,  {D,F,G}=6, {D,H,F}=4,
```

**i Instead of counting the frequencies of all candidate 3-sets just report all the frequent 3-sets from the output of find_freq_itemsets(data,5).**

```
# {D,F,G}
```

## Exe2

**2a Create and report all possible association rules where the union of the antecedent (left-hand-side) and the consequent (right-hand-side) is equal to the set {D,F,G}.**

```
#number of possible Association= (2^t)-2

#   {D}=> {F,G}
#   {F}=> {D,G}
#   {G}=> {D,F}
#   {F,G}=>D
#   {D,G}=> F
#   {D,F}=>G
```

**2bOrganise the rules from (2a) into a lattice (please see the lecture slides about this). No need to make a visualisation, just list the rules in each layer separately.**

```
#   layer 0
#   {D,F,G}  =>{}

#   layer1
#   {D,F} =>G
#   {D,G} =>F
#   {F,G} =>D

#layer2
#   {D}=> {F,G}
#   {F}=> {D,G}
#   {G}=> {D,F}
```

**2c Calculate the support, confidence and lift of all the rules from (2a), report by layers as in (2b).**

```r
#layer1

#support DFG
suppport_DFG <- 6/15

#support of DF
support_DF <- 8/15

#suport of DG
support_DG <- 6/15

#support of FG
support_FG <- 8/15

#layer2
#support of G
support_G <- 8/15
#support of F
support_F <- 12/15
#support of D
support_D <- 11/15

#confidence {D,F} =>G
suppport_DFG/support_DF
```

```
## [1] 0.75
```

```r
#lift of {D,F} =>G
suppport_DFG/(support_DF*support_G)
```

```
## [1] 1.40625
```

```r
#confidence of {D,G} =>F
suppport_DFG/support_DG
```

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

```r
#lift of  {D,G} =>F
suppport_DFG/(support_DG*support_F)
```

```
## [1] 1.25
```

```r
#confidence of {F,G} =>D
suppport_DFG/support_FG
```

```
## [1] 0.75
```

```r
#lift of {F,G} =>D
suppport_DFG/(support_FG*support_D)
```

```
## [1] 1.022727
```

```r
#layer2
#confidence of {D}=> {F,G}
suppport_DFG/support_D
```

```
## [1] 0.5454545
```

```r
#lift of {D}=> {F,G}
suppport_DFG/(support_FG*support_D)
```

```
## [1] 1.022727
```

```r
#confidence of {F}=> {D,G}
suppport_DFG/support_F
```

```
## [1] 0.5
```

```r
#lift of {F}=> {D,G}
suppport_DFG/(support_DG*support_F)
```

```
## [1] 1.25
```

```r
#confidence of {G}=> {D,F}
suppport_DFG/support_DF
```

```
## [1] 0.75
```

```r
#lift of {G}=> {D,F}
suppport_DFG/(support_DF*support_G)
```

```
## [1] 1.40625
```

### 2d Find and report all rules from (2a) that have confidence at least 0.5.

```r
#{D,F} =>G,  {D,G} =>F, {D,G} =>F,  {D}=> {F,G}, {F}=> {D,G}, {G}=> {D,F}
```

### 2e Is this result in agreement with what you obtained in (2d)?

- its not in agreement

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime   support
##         0.5    0.1    1 none FALSE            TRUE       5 0.3333333
##  minlen maxlen target   ext
##       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: 5
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[8 item(s), 15 transaction(s)] done [0.00s].
## sorting and recoding items ... [6 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [26 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

# Exe3

## 3a

```
#number of rules = 27
```

## 3b

The most intresting rule is + {Class=Crew,Sex=Male,Survived=No}

Its intresting because confidence of an itemset implies the same confidence for all the subsets of that particular item.

## 3c

- {Class=Crew,Age=Adult,Survived=No} => {Sex=Male}
- {Class=Crew,Survived=No} => {Sex=Male}
- I think they have the same lift value because one item set is a subset of another

## 3d What is the most interesting rule in these results, other than the ones discussed in (3b) and (3c)?

```
# {Sex=Male} => {Age=Adult}   0.7573830 0.9630272  1.0132040
#the other intresting rule is {Sex=Male} => {Age=Adult} , its intresting to learn  that  majority of th
```

# Exe4

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##       1e-06    0.1    1 none FALSE            TRUE       5   1e-06      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: 0
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[10 item(s), 2201 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [326 rule(s)] done [0.00s].
```

```
## creating S4 object  ... done [0.00s].
```

### (4a) Discuss what you can learn from the 3 rules with the highest lift.

- This rule {Class=2nd,Sex=Male,Survived=Yes} => {Age=Child} tells us there is a moderate correlation between the young male who sat in the second class that survived ie confidence =0.44
- Rule {Class=2nd,Survived=Yes} => {Age=Child} tell us that majority of people who survived in the second class were children. Though this rule is less likely to occur because of the low confidence (0.20203)
- Rule {Class=2nd,Age=Adult,Survived=Yes} => {Sex=Female}, tell us that majority of the adults who survived in the second class were females and this was true based on the high value of confidence

### (4b) Calculate the support count of the antecedent (left-hand-side) in the rules of (4a) by dividing the count (last column) by confidence (3rd column). Which of these rules do you find the most interesting?

```r
#support of {Class=2nd,Sex=Male,Survived=Yes}
11/0.440
```

```
## [1] 25
```

```r
#support of {Class=2nd,Survived=Yes}
24/0.2033898
```

```
## [1] 118
```

```r
#support of {Class=2nd,Age=Adult,Survived=Yes}
80/0.8510638
```

```
## [1] 94
```

```r
#I find rule  {Class=2nd, Survived=Yes}  I find it intresting because it has a high  support and given
#item of all the  all the other antecedant
```

### (4c) Sort all rules by confidence. What can you learn from the 9 rules with confidence 1.0 and lift greater than 3?

```r
rules = rules %>% arrange(-confidence)
#head(rules, n = 10) # remember you can show as many rows as you want by changing n
```

- From the rules we learn that no child survived in the third class
- from the rules were also learn that children who survived were only in the first and second class

### (4d) Sort all rules by support. What can you learn from the 4 rules with support greater than 0.7?

```r
rules = rules %>% arrange(-support)
#head(rules) # remember you can show as many rows as you want by changing n
```

- The majority of the male people on the ship were adults

- we also learn that majority of adults were males