AssociationRuleMining.R

hamze

2020-01-21

```r
####################################################
#Installing the packages
#install.packages("Matrix")
#install.packages("arules")
#install.packages("qlcMatrix")
#install.packages("data.table")
#install.packages("ggplot2")
#install.packages("ggpubr")
#install.packages("gridExtra")
#install.packages("stringr")
#install.packages("RColorBrewer")
####################################################
#set workspace to this folder
setwd("D:/Work/IJGIS/R-scripts")
####################################################
#importing the packages
library(arules)
```

```
## Warning: package 'arules' was built under R version 3.5.3
```

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

```
## Warning: package 'Matrix' was built under R version 3.5.3
```

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

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

```r
library(qlcMatrix)
```

```
## Warning: package 'qlcMatrix' was built under R version 3.5.3
```

```
## Loading required package: slam
```

```
## Warning: package 'slam' was built under R version 3.5.3
## Loading required package: sparsesvd
```

```
## Warning: package 'sparsesvd' was built under R version 3.5.3
```

```r
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.5.3
```

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:slam':
##
##     rollup
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```r
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 3.5.3
```

```
## Loading required package: magrittr
```

```
## Warning: package 'magrittr' was built under R version 3.5.3
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.5.3
```

```r
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.3
```

```r
library(RColorBrewer)
####################################################

#######################################
#######################################
###############TYPE##############
#######################################
#######################################

type_raw <- read.table(file = "../sequences/type-nf-all.txt", sep = ",")
transpose_type_raw <- t(type_raw)[1, ]
pw <- pwMatrix(transpose_type_raw, sep = " ")
tt <- ttMatrix(pw$rownames)
```

```r
distr <- (tt$M*1) %*% (pw$M*1)
distr_ngCMat = as (distr, "ngCMatrix")
ttRows <- tt$rownames
td <- as.data.frame(ttRows)


setnames(td, "ttRows", "labels")
trans1 <- new("transactions", data = distr_ngCMat, itemInfo = td)

type_rules <- apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2],
                                               confidence = 0.5, maxlen = 3, maxtime = 15))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.01638002      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[6484 item(s), 6105 transaction(s)] done [0.03s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
```

```
##  done [0.00s].
## writing ... [40 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
type_rules_d <- inspect(head(sort(type_rules, by = "lift"), 100))
```

```
##      lhs                   rhs      support    confidence lift     count
## [1]  {ADM2,Q-PPLA2}     => {Q-ADM1} 0.01785422 0.6942675  1.796737  109
## [2]  {Q-ADM1,Q-PPL}     => {ADM2}   0.08468468 0.8503289  1.681107  517
## [3]  {Q-,Q-ADM1}        => {ADM2}   0.03652744 0.8446970  1.669972  223
## [4]  {PCLI,Q-}          => {ADM2}   0.03439803 0.8235294  1.628124  210
## [5]  {Q-ADM1,Q-PPLA2} => {ADM2}   0.01785422 0.8195489  1.620254  109
## [6]  {PCLI,Q-PPL}       => {ADM2}   0.07567568 0.8105263  1.602417  462
## [7]  {Q-PPLA2}          => {Q-ADM1} 0.02178542 0.6186047  1.600925  133
## [8]  {PCLI,Q-PPLA2}     => {ADM2}   0.01670762 0.8095238  1.600435  102
## [9]  {ADM2,Q-PPL}       => {Q-ADM1} 0.08468468 0.6103896  1.579665  517
## [10] {ADM2,PCLI}        => {Q-ADM1} 0.15298935 0.5979513  1.547475  934
## [11] {PCLI,Q-ADM1}    => {ADM2}   0.15298935 0.7796327  1.541340  934
## [12] {ADM2,Q-PPLA2}   => {PCLI}   0.01670762 0.6496815  1.520240  102
## [13] {ADM2,Q-}        => {Q-ADM1} 0.03652744 0.5807292  1.502904  223
## [14] {Q-ADM1}         => {ADM2}   0.28992629 0.7503179  1.483384 1770
## [15] {ADM2}           => {Q-ADM1} 0.28992629 0.5731865  1.483384 1770
## [16] {PCLI,Q-PPL}     => {Q-ADM1} 0.05339885 0.5719298  1.480132  326
## [17] {Q-PPLA2}        => {ADM2}   0.02571663 0.7302326  1.443675  157
## [18] {Q-}             => {ADM2}   0.06289926 0.7286528  1.440552  384
## [19] {PCLI,Q-}        => {Q-ADM1} 0.02260442 0.5411765  1.400544  138
## [20] {Q-PPLA2}        => {PCLI}   0.02063882 0.5860465  1.371335  126
## [21] {Q-ADM1,Q-ADM2}  => {ADM2}   0.03783784 0.6834320  1.351150  231
## [22] {Q-PPL}          => {ADM2}   0.13873874 0.6653574  1.315417  847
## [23] {Q-AREA}         => {ADM2}   0.02538903 0.6595745  1.303984  155
## [24] {Q-}             => {Q-ADM1} 0.04324324 0.5009488  1.296436  264
## [25] {ADM2,Q-}        => {PCLI}   0.03439803 0.5468750  1.279675  210
## [26] {ADM2,Q-PPL}     => {PCLI}   0.07567568 0.5454545  1.276351  462
## [27] {ADM2,Q-ADM2}    => {PCLI}   0.05307125 0.5391015  1.261485  324
## [28] {Q-ADM1,Q-PPL}   => {PCLI}   0.05339885 0.5361842  1.254659  326
## [29] {Q-ADM1,Q-ADM2}  => {PCLI}   0.02964783 0.5355030  1.253065  181
## [30] {ADM2,Q-ADM1}    => {PCLI}   0.15298935 0.5276836  1.234768  934
## [31] {Q-,Q-ADM1}      => {PCLI}   0.02260442 0.5227273  1.223170  138
## [32] {Q-ADM1}         => {PCLI}   0.19623260 0.5078423  1.188339 1198
## [33] {PCLI,Q-ADM2}    => {ADM2}   0.05307125 0.6000000  1.186205  324
## [34] {ADM1,ADM2}      => {PCLI}   0.04029484 0.5061728  1.184433  246
## [35] {ADM2}           => {PCLI}   0.25585586 0.5058290  1.183628 1562
## [36] {PCLI}           => {ADM2}   0.25585586 0.5986968  1.183628 1562
## [37] {Q-ADM2}         => {PCLI}   0.08845209 0.5037313  1.178720  540
## [38] {Q-ADM2}         => {ADM2}   0.09844390 0.5606343  1.108378  601
## [39] {ADM1,PCLI}      => {ADM2}   0.04029484 0.5061728  1.000708  246
## [40] {}               => {ADM2}   0.50581491 0.5058149  1.000000 3088
```

```
#plot(type_rules, jitter = 0)
write.csv(x = type_rules_d, file = "result/type_rules_qa.csv")


########################################
########################################
###############SCALE###############
########################################
########################################

scale_raw <- read.table("../sequences/scale-nf-all.txt", sep = ",")
transpose_scale_raw <- t(scale_raw)[1, ]
pw_scale <- pwMatrix(transpose_scale_raw, sep = " ")
tt_scale <- ttMatrix(pw_scale$rownames)
distr_scale <- (tt_scale$M*1) %*% (pw_scale$M*1)
distr_ngCMat_scale = as (distr_scale, "ngCMatrix")
ttRows_scale <- tt_scale$rownames
td_scale <- as.data.frame(ttRows_scale)

setnames(td_scale, "ttRows_scale", "labels")
trans_scale <- new("transactions", data = distr_ngCMat_scale, itemInfo = td_scale)

scale_rules <- apriori(trans_scale, parameter = list(support = 100/trans_scale@data@Dim[2],
                                                     confidence = 0.5, maxlen = 3, maxtime = 15))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.02654632      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3783 item(s), 3767 transaction(s)] done [0.01s].
## sorting and recoding items ... [13 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans_scale, parameter = list(support = 100/
## trans_scale@data@Dim[2], : Mining stopped (maxlen reached). Only patterns up to
## a length of 3 returned!
```

```
##  done [0.00s].
## writing ... [43 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(scale_rules, by = "support"), 100))
```

```
##      lhs          rhs   support    confidence lift      count
## [1]  {}        => {A-7} 0.56225113 0.5622511  1.0000000 2118
## [2]  {}        => {A-9} 0.52428989 0.5242899  1.0000000 1975
## [3]  {}        => {Q-6} 0.51499867 0.5149987  1.0000000 1940
## [4]  {Q-8}     => {A-7} 0.38173613 0.7768774  1.3817266 1438
## [5]  {A-7}     => {Q-8} 0.38173613 0.6789424  1.3817266 1438
## [6]  {Q-6}     => {A-7} 0.33315636 0.6469072  1.1505663 1255
## [7]  {A-7}     => {Q-6} 0.33315636 0.5925401  1.1505663 1255
## [8]  {A-9}     => {A-7} 0.31510486 0.6010127  1.0689399 1187
## [9]  {A-7}     => {A-9} 0.31510486 0.5604344  1.0689399 1187
## [10] {Q-6}     => {A-9} 0.27847093 0.5407216  1.0313410 1049
## [11] {A-9}     => {Q-6} 0.27847093 0.5311392  1.0313410 1049
## [12] {Q-8}     => {A-9} 0.27581630 0.5613182  1.0706257 1039
## [13] {A-9}     => {Q-8} 0.27581630 0.5260759  1.0706257 1039
## [14] {Q-8}     => {Q-6} 0.25457924 0.5180983  1.0060188  959
## [15] {Q-6,Q-8} => {A-7} 0.21953809 0.8623566  1.5337570  827
## [16] {A-7,Q-8} => {Q-6} 0.21953809 0.5751043  1.1167103  827
## [17] {A-7,Q-6} => {Q-8} 0.21953809 0.6589641  1.3410686  827
## [18] {A-9,Q-8} => {A-7} 0.21343244 0.7738210  1.3762907  804
## [19] {A-7,Q-8} => {A-9} 0.21343244 0.5591099  1.0664136  804
## [20] {A-7,A-9} => {Q-8} 0.21343244 0.6773378  1.3784612  804
## [21] {A-9,Q-6} => {A-7} 0.19617733 0.7044805  1.2529641  739
## [22] {A-7,Q-6} => {A-9} 0.19617733 0.5888446  1.1231279  739
## [23] {A-7,A-9} => {Q-6} 0.19617733 0.6225779  1.2088923  739
## [24] {Q-6,Q-8} => {A-9} 0.14600478 0.5735141  1.0938874  550
## [25] {A-9,Q-8} => {Q-6} 0.14600478 0.5293551  1.0278767  550
## [26] {A-9,Q-6} => {Q-8} 0.14600478 0.5243089  1.0670294  550
## [27] {A-8}     => {Q-6} 0.12768782 0.5268346  1.0229825  481
## [28] {Q-7}     => {A-9} 0.11043271 0.5745856  1.0959312  416
## [29] {Q-7}     => {A-7} 0.10618529 0.5524862  0.9826324  400
## [30] {A-6}     => {Q-6} 0.09822140 0.5117566  0.9937046  370
## [31] {A-9,Q-7} => {A-7} 0.06264932 0.5673077  1.0089934  236
## [32] {A-7,Q-7} => {A-9} 0.06264932 0.5900000  1.1253316  236
## [33] {A-8,A-9} => {Q-6} 0.05972923 0.5984043  1.1619530  225
## [34] {Q-7,Q-8} => {A-7} 0.05680913 0.6666667  1.1857098  214
## [35] {A-7,Q-7} => {Q-8} 0.05680913 0.5350000  1.0887871  214
## [36] {A-7,A-8} => {Q-6} 0.05123440 0.6126984  1.1897087  193
## [37] {A-7,A-8} => {A-9} 0.04884523 0.5841270  1.1141298  184
## [38] {Q-7,Q-8} => {A-9} 0.04698699 0.5514019  1.0517118  177
## [39] {A-6,A-7} => {Q-8} 0.03424476 0.5633188  1.1464191  129
## [40] {Q-5,Q-8} => {A-7} 0.03371383 0.7937500  1.4117357  127
## [41] {A-7,Q-5} => {Q-8} 0.03371383 0.5746606  1.1695011  127
## [42] {A-6,A-7} => {Q-6} 0.03371383 0.5545852  1.0768671  127
## [43] {A-6,A-9} => {Q-6} 0.03026281 0.5700000  1.1067990  114
```

```
scale_rules_d <- inspect(head(sort(scale_rules, by = "support"), 100))
```

```
##      lhs          rhs   support    confidence lift      count
## [1]  {}        => {A-7} 0.56225113 0.5622511  1.0000000 2118
## [2]  {}        => {A-9} 0.52428989 0.5242899  1.0000000 1975
## [3]  {}        => {Q-6} 0.51499867 0.5149987  1.0000000 1940
## [4]  {Q-8}     => {A-7} 0.38173613 0.7768774  1.3817266 1438
## [5]  {A-7}     => {Q-8} 0.38173613 0.6789424  1.3817266 1438
## [6]  {Q-6}     => {A-7} 0.33315636 0.6469072  1.1505663 1255
## [7]  {A-7}     => {Q-6} 0.33315636 0.5925401  1.1505663 1255
## [8]  {A-9}     => {A-7} 0.31510486 0.6010127  1.0689399 1187
## [9]  {A-7}     => {A-9} 0.31510486 0.5604344  1.0689399 1187
## [10] {Q-6}     => {A-9} 0.27847093 0.5407216  1.0313410 1049
## [11] {A-9}     => {Q-6} 0.27847093 0.5311392  1.0313410 1049
## [12] {Q-8}     => {A-9} 0.27581630 0.5613182  1.0706257 1039
## [13] {A-9}     => {Q-8} 0.27581630 0.5260759  1.0706257 1039
## [14] {Q-8}     => {Q-6} 0.25457924 0.5180983  1.0060188  959
## [15] {Q-6,Q-8} => {A-7} 0.21953809 0.8623566  1.5337570  827
## [16] {A-7,Q-8} => {Q-6} 0.21953809 0.5751043  1.1167103  827
## [17] {A-7,Q-6} => {Q-8} 0.21953809 0.6589641  1.3410686  827
## [18] {A-9,Q-8} => {A-7} 0.21343244 0.7738210  1.3762907  804
## [19] {A-7,Q-8} => {A-9} 0.21343244 0.5591099  1.0664136  804
## [20] {A-7,A-9} => {Q-8} 0.21343244 0.6773378  1.3784612  804
## [21] {A-9,Q-6} => {A-7} 0.19617733 0.7044805  1.2529641  739
## [22] {A-7,Q-6} => {A-9} 0.19617733 0.5888446  1.1231279  739
## [23] {A-7,A-9} => {Q-6} 0.19617733 0.6225779  1.2088923  739
## [24] {Q-6,Q-8} => {A-9} 0.14600478 0.5735141  1.0938874  550
## [25] {A-9,Q-8} => {Q-6} 0.14600478 0.5293551  1.0278767  550
## [26] {A-9,Q-6} => {Q-8} 0.14600478 0.5243089  1.0670294  550
## [27] {A-8}     => {Q-6} 0.12768782 0.5268346  1.0229825  481
## [28] {Q-7}     => {A-9} 0.11043271 0.5745856  1.0959312  416
## [29] {Q-7}     => {A-7} 0.10618529 0.5524862  0.9826324  400
## [30] {A-6}     => {Q-6} 0.09822140 0.5117566  0.9937046  370
## [31] {A-9,Q-7} => {A-7} 0.06264932 0.5673077  1.0089934  236
## [32] {A-7,Q-7} => {A-9} 0.06264932 0.5900000  1.1253316  236
## [33] {A-8,A-9} => {Q-6} 0.05972923 0.5984043  1.1619530  225
## [34] {Q-7,Q-8} => {A-7} 0.05680913 0.6666667  1.1857098  214
## [35] {A-7,Q-7} => {Q-8} 0.05680913 0.5350000  1.0887871  214
## [36] {A-7,A-8} => {Q-6} 0.05123440 0.6126984  1.1897087  193
## [37] {A-7,A-8} => {A-9} 0.04884523 0.5841270  1.1141298  184
## [38] {Q-7,Q-8} => {A-9} 0.04698699 0.5514019  1.0517118  177
## [39] {A-6,A-7} => {Q-8} 0.03424476 0.5633188  1.1464191  129
## [40] {Q-5,Q-8} => {A-7} 0.03371383 0.7937500  1.4117357  127
## [41] {A-7,Q-5} => {Q-8} 0.03371383 0.5746606  1.1695011  127
## [42] {A-6,A-7} => {Q-6} 0.03371383 0.5545852  1.0768671  127
## [43] {A-6,A-9} => {Q-6} 0.03026281 0.5700000  1.1067990  114
```

```r
#plot(scale_rules, jitter = 0)
write.csv(x = scale_rules_d, file = "result/scale_rules_qa.csv")

#######################################
#######################################
###############PROMINENCE##############
#######################################
#######################################

imp_raw <- read.table("../sequences/prominence-nf-all.txt", sep = ",")
transpose_imp_raw <- t(imp_raw)[1, ]
pw_imp <- pwMatrix(transpose_imp_raw, sep = " ")
tt_imp <- ttMatrix(pw_imp$rownames)
distr_imp <- (tt_imp$M*1) %*% (pw_imp$M*1)
distr_ngCMat_imp = as (distr_imp, "ngCMatrix")
ttRows_imp <- tt_imp$rownames
td_imp <- as.data.frame(ttRows_imp)

setnames(td_imp, "ttRows_imp", "labels")
trans_imp <- new("transactions", data = distr_ngCMat_imp, itemInfo = td_imp)

imp_rules <- apriori(trans_imp, parameter = list(support = 100/trans_imp@data@Dim[2],
                                                 confidence = 0.5, maxlen = 3, maxtime = 15))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.01696065      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
```

```
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[5910 item(s), 5896 transaction(s)] done [0.02s].
## sorting and recoding items ... [14 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans_imp, parameter = list(support = 100/
## trans_imp@data@Dim[2], : Mining stopped (maxlen reached). Only patterns up to a
## length of 3 returned!
```

```
##  done [0.00s].
## writing ... [21 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(imp_rules, by = "support"), 100))
```

```
##      lhs          rhs   support    confidence lift     count
## [1]  {A-4}     => {A-7} 0.19063772 0.5316935  1.107335 1124
## [2]  {Q-6}     => {A-4} 0.17944369 0.5163494  1.440112 1058
## [3]  {A-4}     => {Q-6} 0.17944369 0.5004730  1.440112 1058
## [4]  {Q-6}     => {A-7} 0.17808684 0.5124451  1.067247 1050
## [5]  {Q-3}     => {A-7} 0.16570556 0.5005123  1.042395  977
## [6]  {A-4,Q-6} => {A-7} 0.09616689 0.5359168  1.116131  567
## [7]  {A-7,Q-6} => {A-4} 0.09616689 0.5400000  1.506074  567
## [8]  {A-4,A-7} => {Q-6} 0.09616689 0.5044484  1.451551  567
## [9]  {Q-4}     => {A-7} 0.08344640 0.5162644  1.075201  492
## [10] {A-4,Q-3} => {A-7} 0.07547490 0.5380895  1.120655  445
## [11] {Q-3,Q-6} => {A-4} 0.06580733 0.5914634  1.649607  388
## [12] {A-4,Q-2} => {A-7} 0.06512890 0.5638767  1.174361  384
## [13] {Q-2,Q-6} => {A-4} 0.05885346 0.6087719  1.697880  347
## [14] {A-4,Q-2} => {Q-6} 0.05885346 0.5095448  1.466216  347
## [15] {Q-2,Q-6} => {A-7} 0.05088195 0.5263158  1.096135  300
## [16] {A-4,A-6} => {A-7} 0.03476934 0.5099502  1.062051  205
## [17] {A-3,Q-6} => {Q-3} 0.03171642 0.5040431  1.522458  187
## [18] {A-4,Q-4} => {A-7} 0.02917232 0.5530547  1.151823  172
## [19] {A-3,A-4} => {Q-3} 0.02662822 0.5508772  1.663920  157
## [20] {Q-4,Q-6} => {A-7} 0.02306649 0.5714286  1.190089  136
## [21] {Q-4,Q-6} => {A-4} 0.02069199 0.5126050  1.429669  122
```

```
imp_rules_d <- inspect(head(sort(imp_rules, by = "support"), 100))
```

```
##      lhs          rhs   support    confidence lift     count
## [1]  {A-4}     => {A-7} 0.19063772 0.5316935  1.107335 1124
## [2]  {Q-6}     => {A-4} 0.17944369 0.5163494  1.440112 1058
## [3]  {A-4}     => {Q-6} 0.17944369 0.5004730  1.440112 1058
## [4]  {Q-6}     => {A-7} 0.17808684 0.5124451  1.067247 1050
## [5]  {Q-3}     => {A-7} 0.16570556 0.5005123  1.042395  977
## [6]  {A-4,Q-6} => {A-7} 0.09616689 0.5359168  1.116131  567
## [7]  {A-7,Q-6} => {A-4} 0.09616689 0.5400000  1.506074  567
## [8]  {A-4,A-7} => {Q-6} 0.09616689 0.5044484  1.451551  567
## [9]  {Q-4}     => {A-7} 0.08344640 0.5162644  1.075201  492
## [10] {A-4,Q-3} => {A-7} 0.07547490 0.5380895  1.120655  445
## [11] {Q-3,Q-6} => {A-4} 0.06580733 0.5914634  1.649607  388
## [12] {A-4,Q-2} => {A-7} 0.06512890 0.5638767  1.174361  384
## [13] {Q-2,Q-6} => {A-4} 0.05885346 0.6087719  1.697880  347
## [14] {A-4,Q-2} => {Q-6} 0.05885346 0.5095448  1.466216  347
## [15] {Q-2,Q-6} => {A-7} 0.05088195 0.5263158  1.096135  300
## [16] {A-4,A-6} => {A-7} 0.03476934 0.5099502  1.062051  205
## [17] {A-3,Q-6} => {Q-3} 0.03171642 0.5040431  1.522458  187
## [18] {A-4,Q-4} => {A-7} 0.02917232 0.5530547  1.151823  172
## [19] {A-3,A-4} => {Q-3} 0.02662822 0.5508772  1.663920  157
## [20] {Q-4,Q-6} => {A-7} 0.02306649 0.5714286  1.190089  136
## [21] {Q-4,Q-6} => {A-4} 0.02069199 0.5126050  1.429669  122
```

```r
write.csv(x = imp_rules_d, file = "result/prominence_rules_qa.csv")


#######################################################################################################
#######################################################################################################


##############################################FUNCTIONS################################################
fun.extract.ncomplex.ids = function(questions, n) {
  validIds = c()
  counter = 0
  for (i in 1:length(questions[,1])) {
    qVals = questions[i, 2:length(questions)]
    if (length(qVals[qVals!=""]) == n) {
      counter= counter + 1
      validIds[counter] = questions[i, 1]
    }
  }
  return (validIds)
}

fun.write.simple.complex = function(all, questions, fileAddressSWQ, fileAddressDWQ) {
  swq_ids <- fun.extract.ncomplex.ids(questions, 1)
  swq <- all[all$V1 %in% swq_ids, 2:length(all)]
  dwq <- all[!all$V1 %in% swq_ids, 2:length(all)]
  write.table(row.names = FALSE, file = fileAddressSWQ, x = swq, col.names = FALSE)
  write.table(row.names = FALSE, file = fileAddressDWQ, x = dwq, col.names = FALSE)
  return (TRUE)
}

fun.extract.rules = function(inputAddress, outputAddress, minconf, minsup, window) {
  raw <- read.table(inputAddress, sep = ",")
  transpose_raw <- t(raw)[1, ]
  pw <- pwMatrix(transpose_raw, sep = " ")
  tt <- ttMatrix(pw$rownames)
  distr <- (tt$M*1) %*% (pw$M*1)
  distr_ngCMat = as (distr, "ngCMatrix")
  ttRows <- tt$rownames
  td <- as.data.frame(ttRows)
  setnames(td, "ttRows", "labels")
  trans1 <- new("transactions", data = distr_ngCMat, itemInfo = td)

  rules <- apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2],
                                            confidence = 0.5, maxlen = window, maxtime = 15))
  rules_d <- inspect(head(sort(rules, by = "support"), 100))

  write.csv(x = rules_d, file = outputAddress)
  return (rules)
}

################################ReadFiles##################################
prominence_questions <- read.table("../sequences/prominence-nf-Q.txt",
                                   header = FALSE, sep = " ", col.names = paste0("V",seq_len(5)),
                                   fill = TRUE)
prominence_all <- read.table("../sequences/prominence-nf-all.txt",
                             header = FALSE, sep = " ", col.names = paste0("V",seq_len(20)),
                             fill = TRUE)
prominence_answers <- read.table("../sequences/prominence-nf-A.txt",
                                 header = FALSE, sep = " ", col.names = paste0("V",seq_len(15)),
                                 fill = TRUE)

scale_questions <- read.table("../sequences/scale-nf-Q.txt",
                              header = FALSE, sep = " ", col.names = paste0("V",seq_len(4)),
                              fill = TRUE)
scale_all <- read.table("../sequences/scale-nf-all.txt",
                        header = FALSE, sep = " ", col.names = paste0("V",seq_len(20)),
                        fill = TRUE)
scale_answers <- read.table("../sequences/scale-nf-A.txt",
                            header = FALSE, sep = " ", col.names = paste0("V",seq_len(15)),
                            fill = TRUE)

type_questions <- read.table("../sequences/type-nf-Q.txt",
                             header = FALSE, sep = " ", col.names = paste0("V",seq_len(5)),
                             fill = TRUE)
type_all <- read.table("../sequences/type-nf-all.txt",
                       header = FALSE, sep = " ", col.names = paste0("V",seq_len(20)),
                       fill = TRUE)
type_answers <- read.table("../sequences/type-nf-A.txt",
                           header = FALSE, sep = " ", col.names = paste0("V",seq_len(15)),
                           fill = TRUE)


################################Differentiating SWQ and DWQ##################################
fun.write.simple.complex(all= prominence_all, questions = prominence_questions,
                         fileAddressSWQ = "../sequences/prominence-nf-all-SWQ.txt",
                         fileAddressDWQ = "../sequences/prominence-nf-all-DWQ.txt")
```

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

```r
fun.write.simple.complex(all= scale_all, questions = scale_questions,
                         fileAddressSWQ = "../sequences/scale-nf-all-SWQ.txt",
                         fileAddressDWQ = "../sequences/scale-nf-all-DWQ.txt")
```

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

```
fun.write.simple.complex(all= type_all, questions = type_questions,
                         fileAddressSWQ = "../sequences/type-nf-all-SWQ.txt",
                         fileAddressDWQ = "../sequences/type-nf-all-DWQ.txt")
```

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

```
fun.write.simple.complex(all= type_questions, questions = type_questions,
                         fileAddressSWQ = "../sequences/type-nf-Q-SWQ.txt",
                         fileAddressDWQ = "../sequences/type-nf-Q-DWQ.txt")
```

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

```
fun.write.simple.complex(all= type_answers, questions = type_questions,
                         fileAddressSWQ = "../sequences/type-nf-A-SWQ.txt",
                         fileAddressDWQ = "../sequences/type-nf-A-DWQ.txt")
```

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

```
#############################################################################################

scale_rules_swq = fun.extract.rules (inputAddress = "../sequences/scale-nf-all-SWQ.txt",
                                     outputAddress = "result/scale_swq_rules_qa.csv",
                                     minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.05263158      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[18 item(s), 1900 transaction(s)] done [0.00s].
## sorting and recoding items ... [15 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
##  done [0.00s].
## writing ... [125 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##       lhs          rhs   support    confidence lift     count
## [1]   {}        => {}    1.00000000 1.0000000  1.000000 1900
## [2]   {}        => {NA}  1.00000000 1.0000000  1.000000 1900
## [3]   {}        => {NA}  1.00000000 1.0000000  1.000000 1900
## [4]   {NA}      => {}    1.00000000 1.0000000  1.000000 1900
## [5]   {}        => {A-9} 0.53315789 0.5331579  1.000000 1013
## [6]   {A-9}     => {}    0.53315789 1.0000000  1.000000 1013
## [7]   {}        => {A-9} 0.53315789 0.5331579  1.000000 1013
## [8]   {A-9}     => {NA}  0.53315789 1.0000000  1.000000 1013
## [9]   {NA}      => {A-9} 0.53315789 0.5331579  1.000000 1013
## [10]  {,A-9}    => {NA}  0.53315789 1.0000000  1.000000 1013
## [11]  {A-9,NA}  => {}    0.53315789 1.0000000  1.000000 1013
## [12]  {,NA}     => {A-9} 0.53315789 0.5331579  1.000000 1013
## [13]  {A-8}     => {}    0.40052632 1.0000000  1.000000  761
## [14]  {A-8}     => {NA}  0.40052632 1.0000000  1.000000  761
## [15]  {,A-8}    => {NA}  0.40052632 1.0000000  1.000000  761
## [16]  {A-8,NA}  => {}    0.40052632 1.0000000  1.000000  761
## [17]  {A-7}     => {}    0.38894737 1.0000000  1.000000  739
## [18]  {A-7}     => {NA}  0.38894737 1.0000000  1.000000  739
## [19]  {,A-7}    => {NA}  0.38894737 1.0000000  1.000000  739
## [20]  {A-7,NA}  => {}    0.38894737 1.0000000  1.000000  739
## [21]  {Q-6}     => {}    0.38684211 1.0000000  1.000000  735
## [22]  {Q-6}     => {NA}  0.38684211 1.0000000  1.000000  735
## [23]  {,Q-6}    => {NA}  0.38684211 1.0000000  1.000000  735
## [24]  {NA,Q-6}  => {}    0.38684211 1.0000000  1.000000  735
## [25]  {A-7}     => {A-9} 0.22210526 0.5710419  1.071056  422
## [26]  {A-7,A-9} => {}    0.22210526 1.0000000  1.000000  422
## [27]  {,A-7}    => {A-9} 0.22210526 0.5710419  1.071056  422
## [28]  {A-7,A-9} => {NA}  0.22210526 1.0000000  1.000000  422
## [29]  {A-7,NA}  => {A-9} 0.22210526 0.5710419  1.071056  422
## [30]  {Q-6}     => {A-9} 0.21842105 0.5646259  1.059022  415
## [31]  {A-9,Q-6} => {}    0.21842105 1.0000000  1.000000  415
## [32]  {,Q-6}    => {A-9} 0.21842105 0.5646259  1.059022  415
## [33]  {A-9,Q-6} => {NA}  0.21842105 1.0000000  1.000000  415
## [34]  {NA,Q-6}  => {A-9} 0.21842105 0.5646259  1.059022  415
## [35]  {A-6}     => {}    0.21631579 1.0000000  1.000000  411
## [36]  {A-6}     => {NA}  0.21631579 1.0000000  1.000000  411
## [37]  {,A-6}    => {NA}  0.21631579 1.0000000  1.000000  411
## [38]  {A-6,NA}  => {}    0.21631579 1.0000000  1.000000  411
## [39]  {Q-6}     => {A-8} 0.20210526 0.5224490  1.304406  384
## [40]  {A-8}     => {Q-6} 0.20210526 0.5045992  1.304406  384
## [41]  {A-8,Q-6} => {}    0.20210526 1.0000000  1.000000  384
## [42]  {,Q-6}    => {A-8} 0.20210526 0.5224490  1.304406  384
## [43]  {,A-8}    => {Q-6} 0.20210526 0.5045992  1.304406  384
## [44]  {A-8,Q-6} => {NA}  0.20210526 1.0000000  1.000000  384
## [45]  {NA,Q-6}  => {A-8} 0.20210526 0.5224490  1.304406  384
## [46]  {A-8,NA}  => {Q-6} 0.20210526 0.5045992  1.304406  384
## [47]  {Q-8}     => {}    0.18842105 1.0000000  1.000000  358
## [48]  {Q-8}     => {NA}  0.18842105 1.0000000  1.000000  358
## [49]  {,Q-8}    => {NA}  0.18842105 1.0000000  1.000000  358
## [50]  {NA,Q-8}  => {}    0.18842105 1.0000000  1.000000  358
## [51]  {A-8,A-9} => {}    0.18421053 1.0000000  1.000000  350
## [52]  {A-8,A-9} => {NA}  0.18421053 1.0000000  1.000000  350
## [53]  {Q-5}     => {}    0.15894737 1.0000000  1.000000  302
## [54]  {Q-5}     => {NA}  0.15894737 1.0000000  1.000000  302
## [55]  {,Q-5}    => {NA}  0.15894737 1.0000000  1.000000  302
## [56]  {NA,Q-5}  => {}    0.15894737 1.0000000  1.000000  302
## [57]  {A-7,Q-6} => {}    0.15894737 1.0000000  1.000000  302
## [58]  {A-7,Q-6} => {NA}  0.15894737 1.0000000  1.000000  302
## [59]  {A-7,A-8} => {}    0.14368421 1.0000000  1.000000  273
## [60]  {A-7,A-8} => {NA}  0.14368421 1.0000000  1.000000  273
## [61]  {Q-7}     => {}    0.12736842 1.0000000  1.000000  242
## [62]  {Q-7}     => {NA}  0.12736842 1.0000000  1.000000  242
## [63]  {,Q-7}    => {NA}  0.12736842 1.0000000  1.000000  242
## [64]  {NA,Q-7}  => {}    0.12736842 1.0000000  1.000000  242
## [65]  {Q-8}     => {A-7} 0.12631579 0.6703911  1.723604  240
## [66]  {A-7,Q-8} => {}    0.12631579 1.0000000  1.000000  240
## [67]  {,Q-8}    => {A-7} 0.12631579 0.6703911  1.723604  240
## [68]  {A-7,Q-8} => {NA}  0.12631579 1.0000000  1.000000  240
## [69]  {NA,Q-8}  => {A-7} 0.12631579 0.6703911  1.723604  240
## [70]  {Q-8}     => {A-9} 0.11578947 0.6145251  1.152614  220
## [71]  {A-9,Q-8} => {}    0.11578947 1.0000000  1.000000  220
## [72]  {,Q-8}    => {A-9} 0.11578947 0.6145251  1.152614  220
## [73]  {A-9,Q-8} => {NA}  0.11578947 1.0000000  1.000000  220
## [74]  {NA,Q-8}  => {A-9} 0.11578947 0.6145251  1.152614  220
## [75]  {A-8,Q-6} => {A-9} 0.11052632 0.5468750  1.025728  210
## [76]  {A-9,Q-6} => {A-8} 0.11052632 0.5060241  1.263398  210
## [77]  {A-8,A-9} => {Q-6} 0.11052632 0.6000000  1.551020  210
## [78]  {A-7,Q-6} => {A-9} 0.09894737 0.6225166  1.167603  188
## [79]  {A-6,Q-6} => {}    0.09368421 1.0000000  1.000000  178
## [80]  {A-6,Q-6} => {NA}  0.09368421 1.0000000  1.000000  178
## [81]  {A-7,Q-6} => {A-8} 0.09000000 0.5662252  1.413703  171
## [82]  {A-7,A-8} => {Q-6} 0.09000000 0.6263736  1.619197  171
## [83]  {A-7,A-8} => {A-9} 0.08842105 0.6153846  1.154226  168
## [84]  {A-6,A-8} => {}    0.07842105 1.0000000  1.000000  149
## [85]  {A-6,A-8} => {NA}  0.07842105 1.0000000  1.000000  149
## [86]  {A-8,Q-5} => {}    0.07736842 1.0000000  1.000000  147
## [87]  {A-8,Q-5} => {NA}  0.07736842 1.0000000  1.000000  147
## [88]  {A-9,Q-5} => {}    0.07631579 1.0000000  1.000000  145
## [89]  {A-9,Q-5} => {NA}  0.07631579 1.0000000  1.000000  145
## [90]  {A-7,Q-8} => {A-9} 0.07263158 0.5750000  1.078480  138
## [91]  {A-9,Q-8} => {A-7} 0.07263158 0.6272727  1.612744  138
## [92]  {Q-4}     => {}    0.07157895 1.0000000  1.000000  136
## [93]  {Q-4}     => {NA}  0.07157895 1.0000000  1.000000  136
## [94]  {,Q-4}    => {NA}  0.07157895 1.0000000  1.000000  136
## [95]  {NA,Q-4}  => {}    0.07157895 1.0000000  1.000000  136
## [96]  {Q-7}     => {A-9} 0.07000000 0.5495868  1.030814  133
## [97]  {A-9,Q-7} => {}    0.07000000 1.0000000  1.000000  133
## [98]  {,Q-7}    => {A-9} 0.07000000 0.5495868  1.030814  133
## [99]  {A-9,Q-7} => {NA}  0.07000000 1.0000000  1.000000  133
## [100] {NA,Q-7}  => {A-9} 0.07000000 0.5495868  1.030814  133
```

```
scale_rules_dwq = fun.extract.rules (inputAddress = "../sequences/scale-nf-all-DWQ.txt",
                                     outputAddress = "result/scale_dwq_rules_qa.csv",
                                     minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.05356186      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[18 item(s), 1867 transaction(s)] done [0.00s].
## sorting and recoding items ... [11 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
```

```
##  done [0.00s].
## writing ... [164 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##       lhs          rhs    support   confidence lift      count
## [1]   {}        => {NA}  1.0000000 1.0000000  1.0000000 1867
## [2]   {}        => {}    0.9994644 0.9994644  1.0000000 1866
## [3]   {}        => {NA}  0.9994644 1.0000000  1.0000000 1866
## [4]   {NA}      => {}    0.9994644 0.9994644  1.0000000 1866
## [5]   {}        => {Q-8} 0.7996786 0.7996786  1.0000000 1493
## [6]   {Q-8}     => {}    0.7996786 1.0000000  1.0005359 1493
## [7]   {}        => {Q-8} 0.7996786 0.8001072  1.0005359 1493
## [8]   {Q-8}     => {NA}  0.7996786 1.0000000  1.0000000 1493
## [9]   {NA}      => {Q-8} 0.7996786 0.7996786  1.0000000 1493
## [10]  {,Q-8}    => {NA}  0.7996786 1.0000000  1.0000000 1493
## [11]  {NA,Q-8}  => {}    0.7996786 1.0000000  1.0005359 1493
## [12]  {,NA}     => {Q-8} 0.7996786 0.8001072  1.0005359 1493
## [13]  {}        => {A-7} 0.7386181 0.7386181  1.0000000 1379
## [14]  {A-7}     => {NA}  0.7386181 1.0000000  1.0000000 1379
## [15]  {NA}      => {A-7} 0.7386181 0.7386181  1.0000000 1379
## [16]  {A-7}     => {}    0.7380825 0.9992748  0.9998104 1378
## [17]  {}        => {A-7} 0.7380825 0.7384780  0.9998104 1378
## [18]  {,A-7}    => {NA}  0.7380825 1.0000000  1.0000000 1378
## [19]  {A-7,NA}  => {}    0.7380825 0.9992748  0.9998104 1378
## [20]  {,NA}     => {A-7} 0.7380825 0.7384780  0.9998104 1378
## [21]  {}        => {Q-6} 0.6454205 0.6454205  1.0000000 1205
## [22]  {Q-6}     => {NA}  0.6454205 1.0000000  1.0000000 1205
## [23]  {NA}      => {Q-6} 0.6454205 0.6454205  1.0000000 1205
## [24]  {Q-6}     => {}    0.6448848 0.9991701  0.9997056 1204
## [25]  {}        => {Q-6} 0.6448848 0.6452304  0.9997056 1204
## [26]  {,Q-6}    => {NA}  0.6448848 1.0000000  1.0000000 1204
## [27]  {NA,Q-6}  => {}    0.6448848 0.9991701  0.9997056 1204
## [28] {,NA}      => {Q-6} 0.6448848 0.6452304  0.9997056 1204
## [29] {A-7}      => {Q-8} 0.6416711 0.8687455  1.0863682 1198
## [30] {Q-8}      => {A-7} 0.6416711 0.8024113  1.0863682 1198
## [31] {A-7,Q-8}  => {}    0.6416711 1.0000000  1.0005359 1198
## [32] {,A-7}     => {Q-8} 0.6416711 0.8693759  1.0871566 1198
## [33] {,Q-8}     => {A-7} 0.6416711 0.8024113  1.0863682 1198
## [34] {A-7,Q-8}  => {NA}  0.6416711 1.0000000  1.0000000 1198
## [35] {A-7,NA}   => {Q-8} 0.6416711 0.8687455  1.0863682 1198
## [36] {NA,Q-8}   => {A-7} 0.6416711 0.8024113  1.0863682 1198
## [37] {}         => {A-9} 0.5152651 0.5152651  1.0000000  962
## [38] {A-9}      => {}    0.5152651 1.0000000  1.0005359  962
## [39] {}         => {A-9} 0.5152651 0.5155413  1.0005359  962
## [40] {A-9}      => {NA}  0.5152651 1.0000000  1.0000000  962
## [41] {NA}       => {A-9} 0.5152651 0.5152651  1.0000000  962
## [42] {,A-9}     => {NA}  0.5152651 1.0000000  1.0000000  962
## [43] {A-9,NA}   => {}    0.5152651 1.0000000  1.0005359  962
## [44] {,NA}      => {A-9} 0.5152651 0.5155413  1.0005359  962
## [45] {Q-6}      => {Q-8} 0.5136583 0.7958506  0.9952131  959
## [46] {Q-8}      => {Q-6} 0.5136583 0.6423309  0.9952131  959
## [47] {Q-6,Q-8}  => {}    0.5136583 1.0000000  1.0005359  959
## [48] {,Q-6}     => {Q-8} 0.5136583 0.7965116  0.9960397  959
## [49] {,Q-8}     => {Q-6} 0.5136583 0.6423309  0.9952131  959
## [50] {Q-6,Q-8}  => {NA}  0.5136583 1.0000000  1.0000000  959
## [51] {NA,Q-6}   => {Q-8} 0.5136583 0.7958506  0.9952131  959
## [52] {NA,Q-8}   => {Q-6} 0.5136583 0.6423309  0.9952131  959
## [53] {Q-6}      => {A-7} 0.5104446 0.7908714  1.0707446  953
## [54] {A-7}      => {Q-6} 0.5104446 0.6910805  1.0707446  953
## [55] {A-7,Q-6}  => {NA}  0.5104446 1.0000000  1.0000000  953
## [56] {NA,Q-6}   => {A-7} 0.5104446 0.7908714  1.0707446  953
## [57] {A-7,NA}   => {Q-6} 0.5104446 0.6910805  1.0707446  953
## [58] {A-7,Q-6}  => {}    0.5099089 0.9989507  0.9994860  952
## [59] {,Q-6}     => {A-7} 0.5099089 0.7906977  1.0705095  952
## [60] {,A-7}     => {Q-6} 0.5099089 0.6908563  1.0703973  952
## [61] {A-7,Q-6}  => {Q-8} 0.4429566 0.8677859  1.0851684  827
## [62] {Q-6,Q-8}  => {A-7} 0.4429566 0.8623566  1.1675271  827
## [63] {A-7,Q-8}  => {Q-6} 0.4429566 0.6903172  1.0695620  827
## [64] {A-9}      => {Q-8} 0.4386717 0.8513514  1.0646169  819
## [65] {Q-8}      => {A-9} 0.4386717 0.5485599  1.0646169  819
## [66] {A-9,Q-8}  => {}    0.4386717 1.0000000  1.0005359  819
## [67] {,A-9}     => {Q-8} 0.4386717 0.8513514  1.0646169  819
## [68] {,Q-8}     => {A-9} 0.4386717 0.5485599  1.0646169  819
## [69] {A-9,Q-8}  => {NA}  0.4386717 1.0000000  1.0000000  819
## [70] {A-9,NA}   => {Q-8} 0.4386717 0.8513514  1.0646169  819
## [71] {NA,Q-8}   => {A-9} 0.4386717 0.5485599  1.0646169  819
## [72] {A-9}      => {A-7} 0.4097483 0.7952183  1.0766298  765
## [73] {A-7}      => {A-9} 0.4097483 0.5547498  1.0766298  765
## [74] {A-7,A-9}  => {}    0.4097483 1.0000000  1.0005359  765
## [75] {,A-9}     => {A-7} 0.4097483 0.7952183  1.0766298  765
## [76] {,A-7}     => {A-9} 0.4097483 0.5551524  1.0774111  765
## [77] {A-7,A-9}  => {NA}  0.4097483 1.0000000  1.0000000  765
## [78] {A-9,NA}   => {A-7} 0.4097483 0.7952183  1.0766298  765
## [79] {A-7,NA}   => {A-9} 0.4097483 0.5547498  1.0766298  765
## [80] {A-7,A-9}  => {Q-8} 0.3567220 0.8705882  1.0886726  666
## [81] {A-9,Q-8}  => {A-7} 0.3567220 0.8131868  1.1009571  666
## [82]  {A-7,Q-8} => {A-9} 0.3567220 0.5559265  1.0789136  666
## [83]  {A-9}     => {Q-6} 0.3395822 0.6590437  1.0211075  634
## [84]  {Q-6}     => {A-9} 0.3395822 0.5261411  1.0211075  634
## [85]  {A-9,Q-6} => {}    0.3395822 1.0000000  1.0005359  634
## [86]  {,A-9}    => {Q-6} 0.3395822 0.6590437  1.0211075  634
## [87]  {,Q-6}    => {A-9} 0.3395822 0.5265781  1.0219556  634
## [88]  {A-9,Q-6} => {NA}  0.3395822 1.0000000  1.0000000  634
## [89]  {A-9,NA}  => {Q-6} 0.3395822 0.6590437  1.0211075  634
## [90]  {NA,Q-6}  => {A-9} 0.3395822 0.5261411  1.0211075  634
## [91]  {A-9,Q-6} => {A-7} 0.2951259 0.8690852  1.1766367  551
## [92]  {A-7,A-9} => {Q-6} 0.2951259 0.7202614  1.1159569  551
## [93]  {A-7,Q-6} => {A-9} 0.2951259 0.5781742  1.1220907  551
## [94]  {A-9,Q-6} => {Q-8} 0.2945903 0.8675079  1.0848206  550
## [95]  {A-9,Q-8} => {Q-6} 0.2945903 0.6715507  1.0404856  550
## [96]  {Q-6,Q-8} => {A-9} 0.2945903 0.5735141  1.1130466  550
## [97]  {Q-7}     => {}    0.2581682 1.0000000  1.0005359  482
## [98]  {Q-7}     => {NA}  0.2581682 1.0000000  1.0000000  482
## [99]  {,Q-7}    => {NA}  0.2581682 1.0000000  1.0000000  482
## [100] {NA,Q-7}  => {}    0.2581682 1.0000000  1.0005359  482
```

```
scale_rules_all = fun.extract.rules (inputAddress = "../sequences/scale-nf-all.txt",
                                     outputAddress = "result/scale_all_rules_qa.csv",
                                     minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.02654632      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3783 item(s), 3767 transaction(s)] done [0.01s].
## sorting and recoding items ... [13 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3

## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!

##  done [0.00s].
## writing ... [43 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##      lhs          rhs   support    confidence lift      count
## [1]  {}        => {A-7} 0.56225113 0.5622511  1.0000000 2118
## [2]  {}        => {A-9} 0.52428989 0.5242899  1.0000000 1975
## [3]  {}        => {Q-6} 0.51499867 0.5149987  1.0000000 1940
## [4]  {Q-8}     => {A-7} 0.38173613 0.7768774  1.3817266 1438
## [5]  {A-7}     => {Q-8} 0.38173613 0.6789424  1.3817266 1438
## [6]  {Q-6}     => {A-7} 0.33315636 0.6469072  1.1505663 1255
## [7]  {A-7}     => {Q-6} 0.33315636 0.5925401  1.1505663 1255
## [8]  {A-9}     => {A-7} 0.31510486 0.6010127  1.0689399 1187
## [9]  {A-7}     => {A-9} 0.31510486 0.5604344  1.0689399 1187
## [10] {Q-6}     => {A-9} 0.27847093 0.5407216  1.0313410 1049
## [11] {A-9}     => {Q-6} 0.27847093 0.5311392  1.0313410 1049
## [12] {Q-8}     => {A-9} 0.27581630 0.5613182  1.0706257 1039
## [13] {A-9}     => {Q-8} 0.27581630 0.5260759  1.0706257 1039
## [14] {Q-8}     => {Q-6} 0.25457924 0.5180983  1.0060188  959
## [15] {Q-6,Q-8} => {A-7} 0.21953809 0.8623566  1.5337570  827
## [16] {A-7,Q-8} => {Q-6} 0.21953809 0.5751043  1.1167103  827
## [17] {A-7,Q-6} => {Q-8} 0.21953809 0.6589641  1.3410686  827
## [18] {A-9,Q-8} => {A-7} 0.21343244 0.7738210  1.3762907  804
## [19] {A-7,Q-8} => {A-9} 0.21343244 0.5591099  1.0664136  804
## [20] {A-7,A-9} => {Q-8} 0.21343244 0.6773378  1.3784612  804
## [21] {A-9,Q-6} => {A-7} 0.19617733 0.7044805  1.2529641  739
## [22] {A-7,Q-6} => {A-9} 0.19617733 0.5888446  1.1231279  739
## [23] {A-7,A-9} => {Q-6} 0.19617733 0.6225779  1.2088923  739
## [24] {Q-6,Q-8} => {A-9} 0.14600478 0.5735141  1.0938874  550
## [25] {A-9,Q-8} => {Q-6} 0.14600478 0.5293551  1.0278767  550
## [26] {A-9,Q-6} => {Q-8} 0.14600478 0.5243089  1.0670294  550
## [27] {A-8}     => {Q-6} 0.12768782 0.5268346  1.0229825  481
## [28] {Q-7}     => {A-9} 0.11043271 0.5745856  1.0959312  416
## [29] {Q-7}     => {A-7} 0.10618529 0.5524862  0.9826324  400
## [30] {A-6}     => {Q-6} 0.09822140 0.5117566  0.9937046  370
## [31] {A-9,Q-7} => {A-7} 0.06264932 0.5673077  1.0089934  236
## [32] {A-7,Q-7} => {A-9} 0.06264932 0.5900000  1.1253316  236
## [33] {A-8,A-9} => {Q-6} 0.05972923 0.5984043  1.1619530  225
## [34] {Q-7,Q-8} => {A-7} 0.05680913 0.6666667  1.1857098  214
## [35] {A-7,Q-7} => {Q-8} 0.05680913 0.5350000  1.0887871  214
## [36] {A-7,A-8} => {Q-6} 0.05123440 0.6126984  1.1897087  193
## [37] {A-7,A-8} => {A-9} 0.04884523 0.5841270  1.1141298  184
## [38] {Q-7,Q-8} => {A-9} 0.04698699 0.5514019  1.0517118  177
## [39] {A-6,A-7} => {Q-8} 0.03424476 0.5633188  1.1464191  129
## [40] {Q-5,Q-8} => {A-7} 0.03371383 0.7937500  1.4117357  127
## [41] {A-7,Q-5} => {Q-8} 0.03371383 0.5746606  1.1695011  127
## [42] {A-6,A-7} => {Q-6} 0.03371383 0.5545852  1.0768671  127
## [43] {A-6,A-9} => {Q-6} 0.03026281 0.5700000  1.1067990  114
```

```
prom_rules_swq = fun.extract.rules (inputAddress = "../sequences/prominence-nf-all-SWQ.txt",
                                    outputAddress = "result/prominence_swq_rules_qa.csv",
                                    minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.03310162      1
##  maxlen target   ext
##       3  rules FALSE
##
```

```
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16 item(s), 3021 transaction(s)] done [0.00s].
## sorting and recoding items ... [16 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3

## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!

##  done [0.00s].
## writing ... [141 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##       lhs          rhs   support    confidence lift     count
## [1]   {}        => {NA}  1.00000000 1.0000000  1.000000 3021
## [2]   {}        => {}    1.00000000 1.0000000  1.000000 3021
## [3]   {NA}      => {}    1.00000000 1.0000000  1.000000 3021
## [4]   {}        => {NA}  1.00000000 1.0000000  1.000000 3021
## [5]   {A-7}     => {NA}  0.48063555 1.0000000  1.000000 1452
## [6]   {A-7}     => {}    0.48063555 1.0000000  1.000000 1452
## [7]   {A-7,NA}  => {}    0.48063555 1.0000000  1.000000 1452
## [8]   {,A-7}    => {NA}  0.48063555 1.0000000  1.000000 1452
## [9]   {A-6}     => {NA}  0.37073817 1.0000000  1.000000 1120
## [10]  {A-6}     => {}    0.37073817 1.0000000  1.000000 1120
## [11]  {A-6,NA}  => {}    0.37073817 1.0000000  1.000000 1120
## [12]  {,A-6}    => {NA}  0.37073817 1.0000000  1.000000 1120
## [13]  {A-5}     => {NA}  0.26580602 1.0000000  1.000000  803
## [14]  {A-5}     => {}    0.26580602 1.0000000  1.000000  803
## [15]  {A-5,NA}  => {}    0.26580602 1.0000000  1.000000  803
## [16]  {,A-5}    => {NA}  0.26580602 1.0000000  1.000000  803
## [17]  {A-4}     => {NA}  0.25521351 1.0000000  1.000000  771
## [18]  {A-4}     => {}    0.25521351 1.0000000  1.000000  771
## [19]  {A-4,NA}  => {}    0.25521351 1.0000000  1.000000  771
## [20]  {,A-4}    => {NA}  0.25521351 1.0000000  1.000000  771
## [21]  {Q-2}     => {NA}  0.23601456 1.0000000  1.000000  713
## [22]  {Q-2}     => {}    0.23601456 1.0000000  1.000000  713
## [23]  {NA,Q-2}  => {}    0.23601456 1.0000000  1.000000  713
## [24]  {,Q-2}    => {NA}  0.23601456 1.0000000  1.000000  713
## [25]  {Q-3}     => {NA}  0.23171135 1.0000000  1.000000  700
## [26]  {Q-3}     => {}    0.23171135 1.0000000  1.000000  700
## [27]  {NA,Q-3}  => {}    0.23171135 1.0000000  1.000000  700
## [28]  {,Q-3}    => {NA}  0.23171135 1.0000000  1.000000  700
## [29]  {A-3}     => {NA}  0.19629262 1.0000000  1.000000  593
## [30]  {A-3}     => {}    0.19629262 1.0000000  1.000000  593
## [31]  {A-3,NA}  => {}    0.19629262 1.0000000  1.000000  593
## [32]  {,A-3}    => {NA}  0.19629262 1.0000000  1.000000  593
## [33]  {A-2}     => {NA}  0.15590864 1.0000000  1.000000  471
## [34]  {A-2}     => {}    0.15590864 1.0000000  1.000000  471
## [35]  {A-2,NA}  => {}    0.15590864 1.0000000  1.000000  471
## [36]  {,A-2}     => {NA}  0.15590864 1.0000000  1.000000  471
## [37]  {Q-4}      => {NA}  0.15193644 1.0000000  1.000000  459
## [38]  {Q-4}      => {}    0.15193644 1.0000000  1.000000  459
## [39]  {NA,Q-4}   => {}    0.15193644 1.0000000  1.000000  459
## [40]  {,Q-4}     => {NA}  0.15193644 1.0000000  1.000000  459
## [41]  {Q-6}      => {NA}  0.13604767 1.0000000  1.000000  411
## [42]  {Q-6}      => {}    0.13604767 1.0000000  1.000000  411
## [43]  {NA,Q-6}   => {}    0.13604767 1.0000000  1.000000  411
## [44]  {,Q-6}     => {NA}  0.13604767 1.0000000  1.000000  411
## [45]  {A-4}      => {A-7} 0.13439259 0.5265888  1.095609  406
## [46]  {A-4,A-7}  => {NA}  0.13439259 1.0000000  1.000000  406
## [47]  {A-4,NA}   => {A-7} 0.13439259 0.5265888  1.095609  406
## [48]  {A-4,A-7}  => {}    0.13439259 1.0000000  1.000000  406
## [49]  {,A-4}     => {A-7} 0.13439259 0.5265888  1.095609  406
## [50]  {A-5,A-7}  => {NA}  0.13108242 1.0000000  1.000000  396
## [51]  {A-5,A-7}  => {}    0.13108242 1.0000000  1.000000  396
## [52]  {A-6,A-7}  => {NA}  0.13008937 1.0000000  1.000000  393
## [53]  {A-6,A-7}  => {}    0.13008937 1.0000000  1.000000  393
## [54]  {Q-3}      => {A-7} 0.11817279 0.5100000  1.061095  357
## [55]  {A-7,Q-3}  => {NA}  0.11817279 1.0000000  1.000000  357
## [56]  {NA,Q-3}   => {A-7} 0.11817279 0.5100000  1.061095  357
## [57]  {A-7,Q-3}  => {}    0.11817279 1.0000000  1.000000  357
## [58]  {,Q-3}     => {A-7} 0.11817279 0.5100000  1.061095  357
## [59]  {Q-5}      => {NA}  0.11022840 1.0000000  1.000000  333
## [60]  {Q-5}      => {}    0.11022840 1.0000000  1.000000  333
## [61]  {NA,Q-5}   => {}    0.11022840 1.0000000  1.000000  333
## [62]  {,Q-5}     => {NA}  0.11022840 1.0000000  1.000000  333
## [63]  {A-7,Q-2}  => {NA}  0.10559417 1.0000000  1.000000  319
## [64]  {A-7,Q-2}  => {}    0.10559417 1.0000000  1.000000  319
## [65]  {Q-1}      => {NA}  0.09433962 1.0000000  1.000000  285
## [66]  {Q-1}      => {}    0.09433962 1.0000000  1.000000  285
## [67]  {NA,Q-1}   => {}    0.09433962 1.0000000  1.000000  285
## [68]  {,Q-1}     => {NA}  0.09433962 1.0000000  1.000000  285
## [69]  {A-6,Q-3}  => {NA}  0.09433962 1.0000000  1.000000  285
## [70]  {A-6,Q-3}  => {}    0.09433962 1.0000000  1.000000  285
## [71]  {A-6,Q-2}  => {NA}  0.09433962 1.0000000  1.000000  285
## [72]  {A-6,Q-2}  => {}    0.09433962 1.0000000  1.000000  285
## [73]  {A-4,A-6}  => {NA}  0.08341609 1.0000000  1.000000  252
## [74]  {A-4,A-6}  => {}    0.08341609 1.0000000  1.000000  252
## [75]  {A-1}      => {NA}  0.08043694 1.0000000  1.000000  243
## [76]  {A-1}      => {}    0.08043694 1.0000000  1.000000  243
## [77]  {A-1,NA}   => {}    0.08043694 1.0000000  1.000000  243
## [78]  {,A-1}     => {NA}  0.08043694 1.0000000  1.000000  243
## [79]  {Q-4}      => {A-7} 0.07745780 0.5098039  1.060687  234
## [80]  {A-7,Q-4}  => {NA}  0.07745780 1.0000000  1.000000  234
## [81]  {NA,Q-4}   => {A-7} 0.07745780 0.5098039  1.060687  234
## [82]  {A-7,Q-4}  => {}    0.07745780 1.0000000  1.000000  234
## [83]  {,Q-4}     => {A-7} 0.07745780 0.5098039  1.060687  234
## [84]  {Q-6}      => {A-7} 0.07679576 0.5644769  1.174438  232
## [85]  {A-7,Q-6}  => {NA}  0.07679576 1.0000000  1.000000  232
## [86]  {NA,Q-6}   => {A-7} 0.07679576 0.5644769  1.174438  232
## [87]  {A-7,Q-6}  => {}    0.07679576 1.0000000  1.000000  232
## [88]  {,Q-6}     => {A-7} 0.07679576 0.5644769  1.174438  232
## [89]  {A-4,Q-3}  => {NA}  0.07216154 1.0000000  1.000000  218
## [90]  {A-4,Q-3} => {}    0.07216154 1.0000000  1.000000  218
## [91]  {A-5,A-6} => {NA}  0.07116849 1.0000000  1.000000  215
## [92]  {A-5,A-6} => {}    0.07116849 1.0000000  1.000000  215
## [93]  {A-3,A-7} => {NA}  0.06885137 1.0000000  1.000000  208
## [94]  {A-3,A-7} => {}    0.06885137 1.0000000  1.000000  208
## [95]  {A-6,Q-4} => {NA}  0.06454816 1.0000000  1.000000  195
## [96]  {A-6,Q-4} => {}    0.06454816 1.0000000  1.000000  195
## [97]  {A-3,A-6} => {NA}  0.06322410 1.0000000  1.000000  191
## [98]  {A-3,A-6} => {}    0.06322410 1.0000000  1.000000  191
## [99]  {A-3,Q-2} => {NA}  0.06190003 1.0000000  1.000000  187
## [100] {A-3,Q-2} => {}    0.06190003 1.0000000  1.000000  187
```

```
prom_rules_dwq = fun.extract.rules (inputAddress = "../sequences/prominence-nf-all-DWQ.txt",
                                    outputAddress = "result/prominence_dwq_rules_qa.csv",
                                    minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.03478261      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16 item(s), 2875 transaction(s)] done [0.00s].
## sorting and recoding items ... [16 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
```

```
##  done [0.00s].
## writing ... [231 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##       lhs          rhs   support   confidence lift      count
## [1]   {}        => {NA}  1.0000000 1.0000000  1.0000000 2875
## [2]   {}        => {}    0.9996522 0.9996522  1.0000000 2874
## [3]   {}        => {NA}  0.9996522 1.0000000  1.0000000 2874
## [4]   {NA}      => {}    0.9996522 0.9996522  1.0000000 2874
## [5]   {}        => {Q-6} 0.5697391 0.5697391  1.0000000 1638
## [6]   {Q-6}     => {}    0.5697391 1.0000000  1.0003479 1638
## [7]   {}        => {Q-6} 0.5697391 0.5699374  1.0003479 1638
## [8]   {Q-6}     => {NA}  0.5697391 1.0000000  1.0000000 1638
## [9]   {NA}      => {Q-6} 0.5697391 0.5697391  1.0000000 1638
## [10]  {,Q-6}    => {NA}  0.5697391 1.0000000  1.0000000 1638
## [11]  {NA,Q-6}  => {}     0.5697391 1.0000000  1.0003479 1638
## [12]  {,NA}     => {Q-6}  0.5697391 0.5699374  1.0003479 1638
## [13]  {A-7}     => {}     0.4796522 1.0000000  1.0003479 1379
## [14]  {A-7}     => {NA}   0.4796522 1.0000000  1.0000000 1379
## [15]  {,A-7}    => {NA}   0.4796522 1.0000000  1.0000000 1379
## [16]  {A-7,NA}  => {}     0.4796522 1.0000000  1.0003479 1379
## [17]  {A-4}     => {NA}   0.4671304 1.0000000  1.0000000 1343
## [18]  {A-4}     => {}     0.4667826 0.9992554  0.9996031 1342
## [19]  {,A-4}    => {NA}   0.4667826 1.0000000  1.0000000 1342
## [20]  {A-4,NA}  => {}     0.4667826 0.9992554  0.9996031 1342
## [21]  {Q-3}     => {}     0.4354783 1.0000000  1.0003479 1252
## [22]  {Q-3}     => {NA}   0.4354783 1.0000000  1.0000000 1252
## [23]  {,Q-3}    => {NA}   0.4354783 1.0000000  1.0000000 1252
## [24]  {NA,Q-3}  => {}     0.4354783 1.0000000  1.0003479 1252
## [25]  {Q-2}     => {NA}   0.3593043 1.0000000  1.0000000 1033
## [26]  {Q-2}     => {}     0.3589565 0.9990319  0.9993796 1032
## [27]  {,Q-2}    => {NA}   0.3589565 1.0000000  1.0000000 1032
## [28]  {NA,Q-2}  => {}     0.3589565 0.9990319  0.9993796 1032
## [29]  {A-4}     => {Q-6}  0.3133913 0.6708861  1.1775320  901
## [30]  {Q-6}     => {A-4}  0.3133913 0.5500611  1.1775320  901
## [31]  {A-4,Q-6} => {}     0.3133913 1.0000000  1.0003479  901
## [32]  {,A-4}    => {Q-6}  0.3133913 0.6713860  1.1784095  901
## [33]  {,Q-6}    => {A-4}  0.3133913 0.5500611  1.1775320  901
## [34]  {A-4,Q-6} => {NA}   0.3133913 1.0000000  1.0000000  901
## [35]  {A-4,NA}  => {Q-6}  0.3133913 0.6708861  1.1775320  901
## [36]  {NA,Q-6}  => {A-4}  0.3133913 0.5500611  1.1775320  901
## [37]  {A-7}     => {Q-6}  0.2845217 0.5931835  1.0411492  818
## [38]  {A-7,Q-6} => {}     0.2845217 1.0000000  1.0003479  818
## [39]  {,A-7}    => {Q-6}  0.2845217 0.5931835  1.0411492  818
## [40]  {A-7,Q-6} => {NA}   0.2845217 1.0000000  1.0000000  818
## [41]  {A-7,NA}  => {Q-6}  0.2845217 0.5931835  1.0411492  818
## [42]  {A-5}     => {}     0.2765217 1.0000000  1.0003479  795
## [43]  {A-5}     => {NA}   0.2765217 1.0000000  1.0000000  795
## [44]  {,A-5}    => {NA}   0.2765217 1.0000000  1.0000000  795
## [45]  {A-5,NA}  => {}     0.2765217 1.0000000  1.0003479  795
## [46]  {Q-7}     => {}     0.2636522 1.0000000  1.0003479  758
## [47]  {Q-7}     => {NA}   0.2636522 1.0000000  1.0000000  758
## [48]  {,Q-7}    => {NA}   0.2636522 1.0000000  1.0000000  758
## [49]  {NA,Q-7}  => {}     0.2636522 1.0000000  1.0003479  758
## [50]  {A-4}     => {A-7}  0.2497391 0.5346240  1.1146076  718
## [51]  {A-7}     => {A-4}  0.2497391 0.5206672  1.1146076  718
## [52]  {A-4,A-7} => {}     0.2497391 1.0000000  1.0003479  718
## [53]  {,A-4}    => {A-7}  0.2497391 0.5350224  1.1154382  718
## [54]  {,A-7}    => {A-4}  0.2497391 0.5206672  1.1146076  718
## [55]  {A-4,A-7} => {NA}   0.2497391 1.0000000  1.0000000  718
## [56]  {A-4,NA}  => {A-7}  0.2497391 0.5346240  1.1146076  718
## [57]  {A-7,NA}  => {A-4}  0.2497391 0.5206672  1.1146076  718
## [58]  {Q-3}     => {Q-6}  0.2281739 0.5239617  0.9196519  656
## [59]  {Q-3,Q-6} => {}     0.2281739 1.0000000  1.0003479  656
## [60]  {,Q-3}    => {Q-6}  0.2281739 0.5239617  0.9196519  656
## [61]  {Q-3,Q-6} => {NA}   0.2281739 1.0000000  1.0000000  656
## [62]  {NA,Q-3}  => {Q-6}  0.2281739 0.5239617  0.9196519  656
## [63]  {A-3}     => {NA}   0.2208696 1.0000000  1.0000000  635
## [64]  {A-3}     => {}     0.2205217 0.9984252  0.9987726  634
## [65]  {,A-3}    => {NA}  0.2205217 1.0000000  1.0000000  634
## [66]  {A-3,NA}  => {}    0.2205217 0.9984252  0.9987726  634
## [67]  {A-7,Q-3} => {}    0.2156522 1.0000000  1.0003479  620
## [68]  {A-7,Q-3} => {NA}  0.2156522 1.0000000  1.0000000  620
## [69]  {A-4,Q-3} => {}    0.2118261 1.0000000  1.0003479  609
## [70]  {A-4,Q-3} => {NA}  0.2118261 1.0000000  1.0000000  609
## [71]  {Q-2}     => {Q-6} 0.1982609 0.5517909  0.9684975  570
## [72]  {Q-2,Q-6} => {}    0.1982609 1.0000000  1.0003479  570
## [73]  {,Q-2}    => {Q-6} 0.1982609 0.5523256  0.9694359  570
## [74]  {Q-2,Q-6} => {NA}  0.1982609 1.0000000  1.0000000  570
## [75]  {NA,Q-2}  => {Q-6} 0.1982609 0.5517909  0.9684975  570
## [76]  {Q-2}     => {A-4} 0.1805217 0.5024201  1.0755457  519
## [77]  {A-4,Q-2} => {NA}  0.1805217 1.0000000  1.0000000  519
## [78]  {NA,Q-2}  => {A-4} 0.1805217 0.5024201  1.0755457  519
## [79]  {A-4,Q-2} => {}    0.1801739 0.9980732  0.9984205  518
## [80]  {,Q-2}    => {A-4} 0.1801739 0.5019380  1.0745136  518
## [81]  {A-7,Q-2} => {}    0.1770435 1.0000000  1.0003479  509
## [82]  {A-7,Q-2} => {NA}  0.1770435 1.0000000  1.0000000  509
## [83]  {Q-4}     => {}    0.1718261 1.0000000  1.0003479  494
## [84]  {Q-4}     => {NA}  0.1718261 1.0000000  1.0000000  494
## [85]  {,Q-4}    => {NA}  0.1718261 1.0000000  1.0000000  494
## [86]  {NA,Q-4}  => {}    0.1718261 1.0000000  1.0003479  494
## [87]  {A-4,A-7} => {Q-6} 0.1680000 0.6727019  1.1807192  483
## [88]  {A-4,Q-6} => {A-7} 0.1680000 0.5360710  1.1176245  483
## [89]  {A-7,Q-6} => {A-4} 0.1680000 0.5904645  1.2640250  483
## [90]  {A-6}     => {}    0.1568696 1.0000000  1.0003479  451
## [91]  {A-6}     => {NA}  0.1568696 1.0000000  1.0000000  451
## [92]  {,A-6}    => {NA}  0.1568696 1.0000000  1.0000000  451
## [93]  {A-6,NA}  => {}    0.1568696 1.0000000  1.0003479  451
## [94]  {A-2}     => {NA}  0.1547826 1.0000000  1.0000000  445
## [95]  {A-2}     => {}    0.1544348 0.9977528  0.9981000  444
## [96]  {,A-2}    => {NA}  0.1544348 1.0000000  1.0000000  444
## [97]  {A-2,NA}  => {}    0.1544348 0.9977528  0.9981000  444
## [98]  {A-5,Q-6} => {}    0.1360000 1.0000000  1.0003479  391
## [99]  {A-5,Q-6} => {NA}  0.1360000 1.0000000  1.0000000  391
## [100] {A-4,Q-3} => {Q-6} 0.1349565 0.6371100  1.1182487  388
```

```
type_rules_swq = fun.extract.rules (inputAddress = "../sequences/type-nf-all-SWQ.txt",
                                   outputAddress = "result/type_swq_rules_qa.csv",
                                   minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime   support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.0310752      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
```

```
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[348 item(s), 3218 transaction(s)] done [0.00s].
## sorting and recoding items ... [21 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
```

```
##  done [0.00s].
## writing ... [117 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##       lhs              rhs    support    confidence lift     count
## [1]   {}            => {}     1.00000000 1.0000000  1.000000 3218
## [2]   {}            => {NA}   1.00000000 1.0000000  1.000000 3218
## [3]   {}            => {NA}   1.00000000 1.0000000  1.000000 3218
## [4]   {NA}          => {}     1.00000000 1.0000000  1.000000 3218
## [5]   {PCLI}        => {}     0.41423244 1.0000000  1.000000 1333
## [6]   {PCLI}        => {NA}   0.41423244 1.0000000  1.000000 1333
## [7]   {,PCLI}       => {NA}   0.41423244 1.0000000  1.000000 1333
## [8]   {NA,PCLI}     => {}     0.41423244 1.0000000  1.000000 1333
## [9]   {ADM2}        => {}     0.34897452 1.0000000  1.000000 1123
## [10]  {ADM2}        => {NA}   0.34897452 1.0000000  1.000000 1123
## [11]  {,ADM2}       => {NA}   0.34897452 1.0000000  1.000000 1123
## [12]  {ADM2,NA}     => {}     0.34897452 1.0000000  1.000000 1123
## [13]  {ADM1}        => {}     0.33903045 1.0000000  1.000000 1091
## [14]  {ADM1}        => {NA}   0.33903045 1.0000000  1.000000 1091
## [15]  {,ADM1}       => {NA}   0.33903045 1.0000000  1.000000 1091
## [16]  {ADM1,NA}     => {}     0.33903045 1.0000000  1.000000 1091
## [17]  {ADM2,PCLI}   => {}     0.17060286 1.0000000  1.000000  549
## [18]  {ADM2,PCLI}   => {NA}   0.17060286 1.0000000  1.000000  549
## [19]  {Q-PPL}       => {}     0.14232443 1.0000000  1.000000  458
## [20]  {Q-PPL}       => {NA}   0.14232443 1.0000000  1.000000  458
## [21]  {,Q-PPL}      => {NA}   0.14232443 1.0000000  1.000000  458
## [22]  {NA,Q-PPL}    => {}     0.14232443 1.0000000  1.000000  458
## [23]  {ADM1,PCLI}   => {}     0.14014916 1.0000000  1.000000  451
## [24]  {ADM1,PCLI}   => {NA}   0.14014916 1.0000000  1.000000  451
## [25]  {ADM1,ADM2}   => {}     0.13020510 1.0000000  1.000000  419
## [26]  {ADM1,ADM2}   => {NA}   0.13020510 1.0000000  1.000000  419
## [27]  {Q-ADM1}      => {}     0.11280298 1.0000000  1.000000  363
## [28]  {Q-ADM1}      => {NA}   0.11280298 1.0000000  1.000000  363
## [29]  {,Q-ADM1}     => {NA}   0.11280298 1.0000000  1.000000  363
## [30]  {NA,Q-ADM1}   => {}     0.11280298 1.0000000  1.000000  363
## [31]  {Q-ADM2}      => {}     0.10006215 1.0000000  1.000000  322
## [32]  {Q-ADM2}      => {NA}   0.10006215 1.0000000  1.000000  322
## [33]  {,Q-ADM2}     => {NA}   0.10006215 1.0000000  1.000000  322
## [34]  {NA,Q-ADM2}   => {}     0.10006215 1.0000000  1.000000  322
## [35]  {PPL}         => {}     0.08514605 1.0000000  1.000000  274
## [36]  {PPL}         => {NA}   0.08514605 1.0000000  1.000000  274
## [37]  {,PPL}        => {NA}   0.08514605 1.0000000  1.000000  274
## [38]  {NA,PPL}      => {}     0.08514605 1.0000000  1.000000  274
## [39]  {RGN}         => {}     0.07582349 1.0000000  1.000000  244
## [40]  {RGN}         => {NA}   0.07582349 1.0000000  1.000000  244
## [41]  {,RGN}        => {NA}   0.07582349 1.0000000  1.000000  244
## [42]  {NA,RGN}      => {}     0.07582349 1.0000000  1.000000  244
## [43]  {Q-ADM1}      => {ADM2} 0.07271597 0.6446281  1.847207  234
## [44]  {ADM2,Q-ADM1} => {}     0.07271597 1.0000000  1.000000  234
## [45]  {,Q-ADM1}     => {ADM2} 0.07271597 0.6446281  1.847207  234
## [46]  {ADM2,Q-ADM1} => {NA}   0.07271597 1.0000000  1.000000  234
## [47]  {NA,Q-ADM1}   => {ADM2} 0.07271597 0.6446281  1.847207  234
## [48]  {HTL}         => {}     0.07240522 1.0000000  1.000000  233
## [49]  {HTL}         => {NA}   0.07240522 1.0000000  1.000000  233
## [50]  {,HTL}        => {NA}   0.07240522 1.0000000  1.000000  233
## [51]  {HTL,NA}      => {}     0.07240522 1.0000000  1.000000  233
## [52]  {ADM1,ADM2}   => {PCLI} 0.07022996 0.5393795  1.302118  226
## [53]  {ADM1,PCLI}   => {ADM2} 0.07022996 0.5011086  1.435946  226
## [54]  {Q-ADM1}      => {PCLI} 0.06556868 0.5812672  1.403239  211
## [55]  {PCLI,Q-ADM1} => {}     0.06556868 1.0000000  1.000000  211
## [56]  {,Q-ADM1}     => {PCLI} 0.06556868 0.5812672  1.403239  211
## [57]  {PCLI,Q-ADM1} => {NA}   0.06556868 1.0000000  1.000000  211
## [58]  {NA,Q-ADM1}   => {PCLI} 0.06556868 0.5812672  1.403239  211
## [59]  {ADM2,Q-PPL}  => {}     0.06432567 1.0000000  1.000000  207
## [60]  {ADM2,Q-PPL}  => {NA}   0.06432567 1.0000000  1.000000  207
## [61]  {AREA}        => {}     0.05842138 1.0000000  1.000000  188
## [62]  {AREA}        => {NA}   0.05842138 1.0000000  1.000000  188
## [63]  {,AREA}       => {NA}   0.05842138 1.0000000  1.000000  188
## [64]  {AREA,NA}     => {}     0.05842138 1.0000000  1.000000  188
## [65]  {ADM1,Q-PPL}  => {}     0.05624612 1.0000000  1.000000  181
## [66]  {ADM1,Q-PPL}  => {NA}   0.05624612 1.0000000  1.000000  181
## [67]  {PCLI,Q-PPL}  => {}     0.05469236 1.0000000  1.000000  176
## [68]  {PCLI,Q-PPL}  => {NA}   0.05469236 1.0000000  1.000000  176
## [69]  {Q-ADM2}      => {PCLI} 0.05220634 0.5217391  1.259532  168
## [70]  {PCLI,Q-ADM2} => {}     0.05220634 1.0000000  1.000000  168
## [71]  {,Q-ADM2}     => {PCLI} 0.05220634 0.5217391  1.259532  168
## [72]  {PCLI,Q-ADM2} => {NA}   0.05220634 1.0000000  1.000000  168
## [73]  {NA,Q-ADM2}   => {PCLI} 0.05220634 0.5217391  1.259532  168
## [74]  {STM}         => {}     0.04536979 1.0000000  1.000000  146
## [75]  {STM}         => {NA}   0.04536979 1.0000000  1.000000  146
## [76]  {,STM}        => {NA}   0.04536979 1.0000000  1.000000  146
## [77]  {NA,STM}      => {}     0.04536979 1.0000000  1.000000  146
## [78]  {Q-PCLI}      => {}     0.04350528 1.0000000  1.000000  140
## [79]  {Q-PCLI}      => {NA}   0.04350528 1.0000000  1.000000  140
## [80]  {,Q-PCLI}     => {NA}   0.04350528 1.0000000  1.000000  140
## [81]  {NA,Q-PCLI}   => {}     0.04350528 1.0000000  1.000000  140
## [82]  {ADM3}        => {}     0.04319453 1.0000000  1.000000  139
## [83]  {ADM3}        => {NA}   0.04319453 1.0000000  1.000000  139
## [84]  {,ADM3}       => {NA}   0.04319453 1.0000000  1.000000  139
## [85]  {ADM3,NA}     => {}     0.04319453 1.0000000  1.000000  139
## [86]  {Q-}          => {}     0.04226227 1.0000000  1.000000  136
## [87]  {Q-}          => {NA}   0.04226227 1.0000000  1.000000  136
## [88]  {,Q-}         => {NA}   0.04226227 1.0000000  1.000000  136
## [89]  {NA,Q-}       => {}     0.04226227 1.0000000  1.000000  136
## [90]  {Q-ADM3}      => {}     0.04164077 1.0000000  1.000000  134
## [91]  {Q-ADM3}      => {NA}   0.04164077 1.0000000  1.000000  134
## [92]  {,Q-ADM3}     => {NA}   0.04164077 1.0000000  1.000000  134
## [93]  {NA,Q-ADM3}   => {}     0.04164077 1.0000000  1.000000  134
## [94]  {LCTY}        => {}     0.04008701 1.0000000  1.000000  129
## [95]  {LCTY}        => {NA}   0.04008701 1.0000000  1.000000  129
## [96]  {,LCTY}       => {NA}   0.04008701 1.0000000  1.000000  129
## [97]  {LCTY,NA}     => {}     0.04008701 1.0000000  1.000000  129
## [98]  {ADM2,Q-ADM1} => {PCLI} 0.03977626 0.5470085  1.320535  128
## [99]  {PCLI,Q-ADM1} => {ADM2} 0.03977626 0.6066351  1.738336  128
## [100] {ADM1,Q-ADM2} => {}     0.03853325 1.0000000  1.000000  124
```

```
type_rules_dwq = fun.extract.rules (inputAddress = "../sequences/type-nf-all-DWQ.txt",
                                    outputAddress = "result/type_dwq_rules_qa.csv",
                                    minsup = 0.0034, minconf = 0.5, window = 3)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime    support minlen
##         0.5    0.1    1 none FALSE            TRUE      15 0.03463803      1
##  maxlen target   ext
##       3  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 100
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[253 item(s), 2887 transaction(s)] done [0.00s].
## sorting and recoding items ... [15 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(trans1, parameter = list(support = 100/trans1@data@Dim[2], :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
```

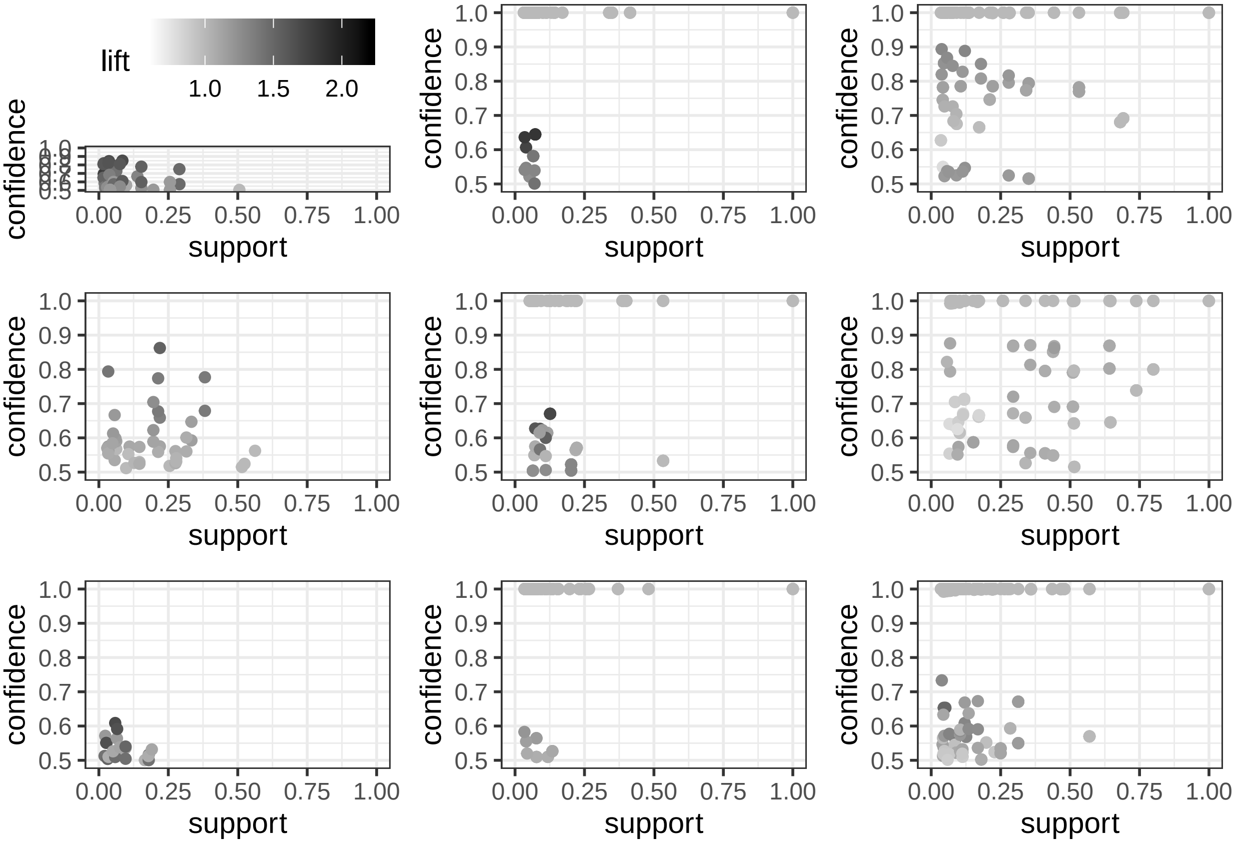
```
##  done [0.00s].
## writing ... [164 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
##       lhs                rhs      support    confidence lift      count
## [1]   {}              => {NA}     1.00000000 1.0000000  1.0000000 2887
## [2]   {}              => {}       0.99965362 0.9996536  1.0000000 2886
## [3]   {}              => {NA}     0.99965362 1.0000000  1.0000000 2886
## [4]   {NA}            => {}       0.99965362 0.9996536  1.0000000 2886
## [5]   {}              => {Q-ADM1} 0.69137513 0.6913751  1.0000000 1996
## [6]   {Q-ADM1}        => {}       0.69137513 1.0000000  1.0003465 1996
## [7]   {}              => {Q-ADM1} 0.69137513 0.6916147  1.0003465 1996
## [8]   {Q-ADM1}        => {NA}     0.69137513 1.0000000  1.0000000 1996
## [9]   {NA}            => {Q-ADM1} 0.69137513 0.6913751  1.0000000 1996
## [10]  {,Q-ADM1}       => {NA}     0.69137513 1.0000000  1.0000000 1996
## [11]  {NA,Q-ADM1}     => {}       0.69137513 1.0000000  1.0003465 1996
## [12]  {,NA}           => {Q-ADM1} 0.69137513 0.6916147  1.0003465 1996
## [13]  {}              => {ADM2}   0.68063734 0.6806373  1.0000000 1965
## [14]  {ADM2}          => {NA}     0.68063734 1.0000000  1.0000000 1965
## [15]  {NA}            => {ADM2}   0.68063734 0.6806373  1.0000000 1965
## [16]  {ADM2}          => {}        0.68029096 0.9994911  0.9998374 1964
## [17]  {}              => {ADM2}    0.68029096 0.6805267  0.9998374 1964
## [18]  {,ADM2}         => {NA}      0.68029096 1.0000000  1.0000000 1964
## [19]  {ADM2,NA}       => {}        0.68029096 0.9994911  0.9998374 1964
## [20]  {,NA}           => {ADM2}    0.68029096 0.6805267  0.9998374 1964
## [21]  {ADM2}          => {Q-ADM1}  0.53204018 0.7816794  1.1306154 1536
## [22]  {Q-ADM1}        => {ADM2}    0.53204018 0.7695391  1.1306154 1536
## [23]  {ADM2,Q-ADM1}   => {}        0.53204018 1.0000000  1.0003465 1536
## [24]  {,ADM2}         => {Q-ADM1}  0.53204018 0.7820774  1.1311911 1536
## [25]  {,Q-ADM1}       => {ADM2}    0.53204018 0.7695391  1.1306154 1536
## [26]  {ADM2,Q-ADM1}   => {NA}      0.53204018 1.0000000  1.0000000 1536
## [27]  {ADM2,NA}       => {Q-ADM1}  0.53204018 0.7816794  1.1306154 1536
## [28]  {NA,Q-ADM1}     => {ADM2}    0.53204018 0.7695391  1.1306154 1536
## [29]  {PCLI}          => {}        0.44198130 1.0000000  1.0003465 1276
## [30]  {PCLI}          => {NA}      0.44198130 1.0000000  1.0000000 1276
## [31]  {,PCLI}         => {NA}      0.44198130 1.0000000  1.0000000 1276
## [32]  {NA,PCLI}       => {}        0.44198130 1.0000000  1.0003465 1276
## [33]  {PCLI}          => {ADM2}    0.35088327 0.7938871  1.1663879 1013
## [34]  {ADM2}          => {PCLI}    0.35088327 0.5155216  1.1663879 1013
## [35]  {ADM2,PCLI}     => {}        0.35088327 1.0000000  1.0003465 1013
## [36]  {,PCLI}         => {ADM2}    0.35088327 0.7938871  1.1663879 1013
## [37]  {,ADM2}         => {PCLI}    0.35088327 0.5157841  1.1669818 1013
## [38]  {ADM2,PCLI}     => {NA}      0.35088327 1.0000000  1.0000000 1013
## [39]  {NA,PCLI}       => {ADM2}    0.35088327 0.7938871  1.1663879 1013
## [40]  {ADM2,NA}       => {PCLI}    0.35088327 0.5155216  1.1663879 1013
## [41]  {PCLI}          => {Q-ADM1}  0.34187738 0.7735110  1.1188007  987
## [42]  {PCLI,Q-ADM1}   => {}        0.34187738 1.0000000  1.0003465  987
## [43]  {,PCLI}         => {Q-ADM1}  0.34187738 0.7735110  1.1188007  987
## [44]  {PCLI,Q-ADM1}   => {NA}      0.34187738 1.0000000  1.0000000  987
## [45]  {NA,PCLI}       => {Q-ADM1}  0.34187738 0.7735110  1.1188007  987
## [46]  {Q-PPL}         => {NA}      0.28229997 1.0000000  1.0000000  815
## [47]  {Q-PPL}         => {}        0.28195359 0.9987730  0.9991191  814
## [48]  {,Q-PPL}        => {NA}      0.28195359 1.0000000  1.0000000  814
## [49]  {NA,Q-PPL}      => {}        0.28195359 0.9987730  0.9991191  814
## [50]  {ADM2,PCLI}     => {Q-ADM1}  0.27918254 0.7956565  1.1508318  806
## [51]  {PCLI,Q-ADM1}   => {ADM2}    0.27918254 0.8166160  1.1997814  806
## [52]  {ADM2,Q-ADM1}   => {PCLI}    0.27918254 0.5247396  1.1872439  806
## [53]  {Q-ADM2}        => {}        0.25978524 1.0000000  1.0003465  750
## [54]  {Q-ADM2}        => {NA}      0.25978524 1.0000000  1.0000000  750
## [55]  {,Q-ADM2}       => {NA}      0.25978524 1.0000000  1.0000000  750
## [56]  {NA,Q-ADM2}     => {}        0.25978524 1.0000000  1.0003465  750
## [57]  {Q-PPL}         => {ADM2}    0.22168341 0.7852761  1.1537364  640
## [58]  {ADM2,Q-PPL}    => {NA}      0.22168341 1.0000000  1.0000000  640
## [59]  {NA,Q-PPL}      => {ADM2}    0.22168341 0.7852761  1.1537364  640
## [60]  {ADM2,Q-PPL}    => {}        0.22133703 0.9984375  0.9987835  639
## [61]  {,Q-PPL}        => {ADM2}    0.22133703 0.7850123  1.1533488  639
## [62]  {Q-PPL}         => {Q-ADM1}  0.21059924 0.7460123  1.0790268  608
## [63]  {Q-ADM1,Q-PPL}  => {}        0.21059924 1.0000000  1.0003465  608
## [64]  {,Q-PPL}        => {Q-ADM1}  0.21059924 0.7469287  1.0803524  608
## [65]  {Q-ADM1,Q-PPL}  => {NA}      0.21059924 1.0000000  1.0000000  608
## [66]  {NA,Q-PPL}      => {Q-ADM1}  0.21059924 0.7460123  1.0790268  608
## [67]  {ADM2,Q-PPL}    => {Q-ADM1}  0.17907863 0.8078125  1.1684142  517
## [68]  {Q-ADM1,Q-PPL}  => {ADM2}    0.17907863 0.8503289  1.2493128  517
## [69]  {Q-ADM2}        => {ADM2}    0.17284378 0.6653333  0.9775152  499
## [70]  {ADM2,Q-ADM2}   => {}       0.17284378 1.0000000  1.0003465  499
## [71]  {,Q-ADM2}       => {ADM2}   0.17284378 0.6653333  0.9775152  499
## [72]  {ADM2,Q-ADM2}   => {NA}     0.17284378 1.0000000  1.0000000  499
## [73]  {NA,Q-ADM2}     => {ADM2}   0.17284378 0.6653333  0.9775152  499
## [74]  {PCLI,Q-PPL}    => {}       0.13647385 1.0000000  1.0003465  394
## [75]  {PCLI,Q-PPL}    => {NA}     0.13647385 1.0000000  1.0000000  394
## [76]  {Q-}            => {}       0.13543471 1.0000000  1.0003465  391
## [77]  {Q-}            => {NA}     0.13543471 1.0000000  1.0000000  391
## [78]  {,Q-}           => {NA}     0.13543471 1.0000000  1.0000000  391
## [79]  {NA,Q-}         => {}       0.13543471 1.0000000  1.0003465  391
## [80]  {PCLI,Q-ADM2}   => {}       0.12885348 1.0000000  1.0003465  372
## [81]  {PCLI,Q-ADM2}   => {NA}     0.12885348 1.0000000  1.0000000  372
## [82]  {PCLI,Q-PPL}    => {ADM2}   0.12123311 0.8883249  1.3051368  350
## [83]  {ADM2,Q-PPL}    => {PCLI}   0.12123311 0.5468750  1.2373261  350
## [84]  {Q-ADM1,Q-ADM2} => {}       0.11707655 1.0000000  1.0003465  338
## [85]  {Q-ADM1,Q-ADM2} => {NA}     0.11707655 1.0000000  1.0000000  338
## [86]  {PCLI,Q-PPL}    => {Q-ADM1} 0.11291999 0.8274112  1.1967615  326
## [87]  {Q-ADM1,Q-PPL}  => {PCLI}   0.11291999 0.5361842  1.2131378  326
## [88]  {Q-}            => {ADM2}   0.10633876 0.7851662  1.1535750  307
## [89]  {ADM2,Q-}       => {}       0.10633876 1.0000000  1.0003465  307
## [90]  {,Q-}           => {ADM2}   0.10633876 0.7851662  1.1535750  307
## [91]  {ADM2,Q-}       => {NA}     0.10633876 1.0000000  1.0000000  307
## [92]  {NA,Q-}         => {ADM2}   0.10633876 0.7851662  1.1535750  307
## [93]  {Q-}            => {Q-ADM1} 0.09144441 0.6751918  0.9765926  264
## [94]  {Q-,Q-ADM1}     => {}       0.09144441 1.0000000  1.0003465  264
## [95]  {,Q-}           => {Q-ADM1} 0.09144441 0.6751918  0.9765926  264
## [96]  {Q-,Q-ADM1}     => {NA}     0.09144441 1.0000000  1.0000000  264
## [97]  {NA,Q-}         => {Q-ADM1} 0.09144441 0.6751918  0.9765926  264
## [98]  {PCLI,Q-ADM2}   => {ADM2}   0.09075165 0.7043011  1.0347670  262
## [99]  {ADM2,Q-ADM2}   => {PCLI}   0.09075165 0.5250501  1.1879464  262
## [100] {Q-ADM1,Q-ADM2} => {ADM2}   0.08001386 0.6834320  1.0041059  231
```

```
t_p = ggplot(type_rules@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "top", legend.box = "horizontal") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))
ts_p = ggplot(type_rules_swq@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))
td_p = ggplot(type_rules_dwq@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))

s_p = ggplot(scale_rules@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))
ss_p = ggplot(scale_rules_swq@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))
sd_p = ggplot(scale_rules_dwq@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() +theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))

p_p = ggplot(imp_rules@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))
ps_p = ggplot(prom_rules_swq@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))
pd_p = ggplot(prom_rules_dwq@quality, aes(x=support, y=confidence)) +
  geom_point(aes(colour= lift)) +
  scale_color_gradient(low = "#ffffff", high = "#000000",  limits=c(0.6, 2.2)) +
  theme_bw() + theme(legend.position = "none") +
  coord_cartesian(xlim =c(0.0, 1.0), ylim = c(0.5, 1.0))

grid.arrange(t_p, ts_p, td_p, s_p, ss_p, sd_p, p_p, ps_p, pd_p, ncol=3, nrow = 3, layout_matrix = rbind
```


lift
1.0
1.5
2.0
confidence
1.0
0.9
0.8
0.7
0.6
0.5
support
0.00
0.25
0.50
0.75
1.00