## **Association Rules**

### **Problem statement**

Prepare rules for the all the data sets

- 1) Try different values of support and confidence. Observe the change in number of rules for different support, confidence values
- 2) Change the minimum length in apriori algorithm
- 3) Visualize the obtained rules using different plots

#### **Answer:**

### Rcode:

#Different set of rule values for Groceries Dataset using apriori algorithm.

```
groceries_data <- read.transactions(file.choose(),format = "basket")

View(groceries_data)

attach(groceries_data)

summary(groceries_data)

str(groceries_data)

class(groceries_data)

install.packages("arules")

install.packages("arulesViz")

library(arules)

library(arules Viz)

#to see most frequent items

FrequentItem <- eclat(groceries_data,parameter = list(support=0.07,maxlen=15))

inspect(FrequentItem)
```

```
itemFrequencyPlot(groceries_data, topN=10, type="absolute", main="Item
Frequency")
rules <- apriori(groceries_data,parameter =
list(support=0.002,confidence=0.5,minlen=2,maxlen=5))
rules
inspect(head(sort(rules,by="lift"),n=15))
inspect(tail(sort(rules,by="lift"),n=15))
plot(rules)
quality(head(rules))
plot(rules, method = "grouped",control = list(cex=0.90))
plot(rules,method = "scatterplot",control = list(cex=0.90))
plot(rules,method = "graph",control = list(cex=0.90))
rules 1 <- apriori(groceries data, parameter =
list(support=0.001,confidence=0.5,minlen=3,maxlen=5))
rules 1
rules_conf <- sort (rules1, by="confidence", decreasing=T)
inspect(head(rules conf))
rules_lift <- sort(rules1,by="lift",decreasing = T)
inspect(head(rules_lift))
plot(rules1, method = "grouped",control = list(cex=0.90))
plot(rules1,method = "scatterplot",control = list(cex=0.90))
plot(rules1,method = "graph",control = list(cex=0.90))
#remove redundant rules
subsetRules <- which(colSums(is.subset(rules1, rules1)) > 1)
length(subsetRules)
```

```
rules1 <- rules1[-subsetRules]
rules 1
subsetRules1 <- which(colSums(is.subset(rules, rules)) > 1)
length(subsetRules1)
rules <- rules[-subsetRules1]
rules
Console:
> groceries_data <- read.transactions(file.choose(),format = "basket")</pre>
> View(groceries_data)
 A_rules.R × Q groceries_data ×
                                 Untitled1* ×
  Show Attributes
                                                                                  Q,
                                                      Value
 Name
                           S4 [9835 x 6928] (arules::transact S4 object of class transactions
  groceries_data
    data
                           S4 [6928 x 9835] (Matrix::ngCMa S4 object of class ngCMatrix
    itemInfo
                           list [6928 x 1] (S3: data.frame)
                                                      A data frame with 6928 rows and 1 column
    itemsetInfo
                           list [0 x 0] (S3: data.frame)
                                                      A data.frame with 0 rows and 0 columns
> summary(groceries_data)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 6928 columns (items) and a density of 0.0005240481
most frequent items:
              bags vegetables, whole
                                                                          whole
                                                       beer
                                   940
                                                        829
               971
                                                                            717
         tropical
                               (Other)
               482
                                 31768
element (itemset/transaction) length distribution:
sizes
         2
               3
                           5
                                 6
                                       7
                                             8
                                                   9
                                                        10
                                                              11
                                                                    12
                                                                          13
                                                                                14
                                                                                      15
   1
16
                                                              76
1380 2743 1782 1246
                         907
                               599
                                     413
                                          294
                                                166
                                                        94
                                                                    43
                                                                          39
                                                                                20
                                                                                      10
        18
              19
                    20
                          21
                                23
  17
                     3
   Min. 1st Qu.
                    Median
                                Mean 3rd Qu.
                                                   Max.
                     3.000
           2.000
                                        5.000
                                                23.000
                               3.631
includes extended item information - examples:
                   labels
             (appetizer)
  (appetizer),bathroom
     (appetizer),cake
3
> str(groceries_data)
Formal class 'transactions' [package "arules"] with 3 slots
```

```
:Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
: int [1:35707] 738 1531 3166 5794 3228 5913 3963 6757 96
   ..@ data
   .. .. ..@ i
1125 ...
                            : int [1:9836] 0 4 6 8 13 20 23 24 28 30 ...
: int [1:2] 6928 9835
   .. .. ..@ p
   .. .. ..@ Dim
   .. .. ..@ Dimnames:List of 2
   .. .. .. ..$ : NULL
               ..$ : NULL
   .. .. ..@ factors : list()
..@ itemInfo :'data.frame': 6928 obs. of 1 variable: ....$ labels: chr [1:6928] "(appetizer)" "(appetizer),bathroom" "(appetizer),cake" "(appetizer),candy,cling" ...
..@ itemsetInfo:'data.frame':
> class(groceries_data)
[1] "transactions"
                                                     0 obs. of 0 variables
attr(,"package")
[1] "arules"
> install.packages("arules")
> install.packages("arulesviz")
  library(arules)
> library(arulesviz)
> #to see most frequent items
  FrequentItem <- eclat(groceries_data,parameter = list(support=0.07,maxlen=1
Eclat
parameter specification:
 tidLists support minlen maxlen
                                                             target
                                        15 frequent itemsets FALSE
                  0.07
algorithmic control:
 sparse sort verbose
              -2
                      TRUE
Absolute minimum support count: 688
create itemset ...
set transactions ...[6928 item(s), 9835 transaction(s)] done [0.06s]. sorting and recoding items ... [4 item(s)] done [0.00s]. creating sparse bit matrix ... [4 row(s), 9835 column(s)] done [0.00s]. writing ... [4 set(s)] done [0.00s]. Creating S4 object ... done [0.00s].
> inspect(FrequentItem)
      items
                                 support
[1] {vegetables,whole} 0.09557702 940
                                0.09872903 971
[2] {bags}
                                0.08429080 829
[3] {beer}
[4] {whole}
                                0.07290290 717
> itemFrequencyPlot(groceries_data, topN=10, type="absolute", main="Item Freq
uency")
                               Item Frequency
tem frequency (absolute)
      800
```

Whole

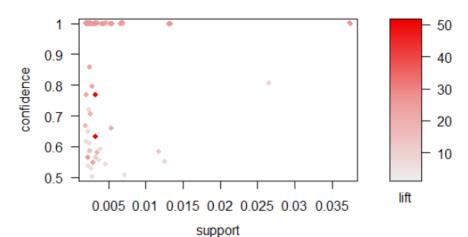
Modical

other

```
> rules <- apriori(groceries_data,parameter = list(support=0.002,confidence=0</pre>
.5,minlen=2,maxlen=5))
Apriori
Parameter specification:
 confidence minval smax arem aval original Support maxtime support minlen max
          0.5
                                                                        5
                                                                             0.002
                   0.1
                            1 none FALSE
                                                          TRUE
                                                                                           2
5
 target
            ext
  rules FALSE
Algorithmic control:
 filter tree heap memopt load sort verbose
     0.1 TRUE TRUE FALSE TRUE
Absolute minimum support count: 19
set item appearances ...[0 item(s)] done [0.00s]. set transactions ...[6928 item(s), 9835 transaction(s)] done [0.05s]. sorting and recoding items ... [257 item(s)] done [0.01s]. creating transaction tree ... done [0.01s]. checking subsets of size 1 2 3 4 done [0.00s].
writing ... [101 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
set of 101 rules
> inspect(head(sort(rules,by="lift"),n=15))
                                                                                           count
      1hs
                                    rhs
                                                   support
                                                                  confidence lift
                                                   0.003050330 0.7692308
                                                                                51.46520
      {salty}
                                    {snack}
                                                                                             30
                               =>
[2]
[3]
       {misc.}
                                    {beverages} 0.003152008 0.6326531
                               =>
                                                                                51.42267
                                                   0.002338587 1.0000000
                                                                                26.72554
                                    {bakery}
                                                                                             23
       {product,shopping}
                               =>
       {product,shopping}
[4]
                                    {life}
                                                   0.002338587 1.0000000
                                                                                26.72554
                                                                                            23
                               =>
                                                                                26.72554
26.72554
 [5]
      {beer,long}
                                                   0.002440264 1.0000000
                                                                                            24
                               =>
                                    {bakery}
 Г6Т
      {beer,long}
                                    {life}
                                                   0.002440264 1.0000000
                                                                                             24
                               =>
                                                   0.002541942 1.0000000
[7]
      {product, specialty} =>
                                    {bakery}
                                                                                26.72554
                                                                                             25
[8]
                                                   0.002541942 1.0000000
                                                                                26.72554
                                                                                             25
      {product, specialty} =>
                                   {life}
      {snack,long}
{snack,long}
 9]
                                                   0.003355363 1.0000000
                                                                                26.72554
                                                                                             33
                                    {bakery}
                               =>
                                                                                26.72554
 [10]
                               =>
                                    {life}
                                                   0.003355363 1.0000000
                                                                                             33
      {juice,long}
{juice,long}
{product}
                                                   0.004270463 1.0000000
[11]
                               =>
                                    {bakery}
                                                                                26.72554
                                                                                             42
                                                   0.004270463 1.0000000
                                                                                26.72554
                                                                                             42
Г121
                               =>
                                    {life}
                                                   0.013218099 1.0000000
[13]
                                    {bakery}
                                                                                26.72554
                                                                                           130
                               =>
[14]
      {product}
                               => {life}
                                                   0.013218099 1.0000000
                                                                                26.72554 130
[15] {bakery}
                                                   0.037417387 1.0000000
                                                                                26.72554 368
                               => {life}
> inspect(tail(sort(rules,by="lift"),n=15))
      lhs
                                                     rhs
                                                                             support
                                                                             0.002338587
      {bakery,life,product,shopping}
                                                 => {bags}
[1]
[2]
[3]
[4]
[5]
      {canned}
                                                     {beer}
                                                                             0.026537875
                                                 =>
      {milk}
                                                                             0.012709710
                                                 =>
                                                     {whole}
      {milk,butter,whipped/sour}
                                                      [vegetables,whole} 0.002338587
                                                      vegetables,whole} 0.002236909
       {fruit,tropical,vegetables,other} =>
[6]
      {fruit,pip,vegetables,other}
                                                      vegetables, whole \ 0.002135231
                                                 =>
Ī7Ī
      {water,bottled}
                                                     {beer}
                                                                             0.004677173
                                                 =>
[8]
[9]
      {fruit, root, fruit, tropical}
                                                     {vegetables, whole} 0.002541942
                                                 =>
                                                     {vegetables, whole} 0.003965430
      {milk,yogurt,whipped/sour}
                                                 => {vegetables,whole} 0.002440264
[10]
      {meat,root}
                                                 -> {vegetables,whole} 0.002440204
-> {vegetables,whole} 0.003558719
-> {vegetables,whole} 0.002236909
-> {vegetables,whole} 0.002745297
-> {vegetables,whole} 0.007320793
-> {vegetables,whole} 0.002745297
[11]
      {fruit,pip,fruit,root}
 12
      {cheese,vegetables,other}
[13]
      {vegetables,onions,other}
     {fruit,root,vegetables,other}
{fruit,root,tropical}
Γ15Ī
      confidence lift
                                 count
[1]
[2]
      1.0000000
                    10.128733
                                 23
      0.8080495
                     9.586450 261
[3]
      0.5530973
                     7.586768 125
                     7.520113
 [4]
      0.7187500
                                  23
                     6.770025
 [5]
      0.6470588
                                  22
Г61
      0.6176471
                     6.462297
```

```
0.5411765
0.6097561
                       6.420351
6.379735
                                      46
25
 *8
[9]
                                      39
       0.5909091
                        6.182544
      0.5853659
0.555556
[10]
                        6.124546
                                      24
[11]
                        5.812648
                                      35
22
      0.5365854
0.5294118
[12]
                        5.614167
[13أ
                        5.539111
                                      27
      0.5070423
                        5.305064
                                      72
Γ14Ī
[15] 0.5000000
                        5.231383
                                      27
> plot(rules)
```

## Scatter plot for 101 rules

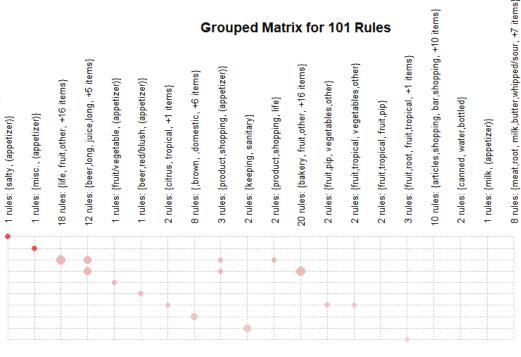


## quality(head(rules))

support confidence lift count 0.003050330 0.7692308 51.46520 30 0.004880529 1.0000000 10.12873 48 0.002236909 0.5641026 26.54521 22 1.0000000 10.12873 0.002033554 20 0.003152008 0.6326531 51.42267 31 1.0000000 10.12873 6 0.003050330 30

plot(rules, method = "grouped",control = list(cex=0.90))

#### 10 rules: {articles,shopping, bar,shopping, +10 items} **Grouped Matrix for 101 Rules** 3 rules: {fruit,root, fruit,tropical, +1 items} 12 rules: {beer,long, juice,long, +5 items} 20 rules: {bakery, fruit,other, +16 items} rules: {product,shopping, (appetizer)} rules: {fruit,tropical, vegetables,other 8 rules: {,brown, ,domestic, +6 items} rules: {fruit/vegetable, (appetizer)} {beer,red/blush, (appetizer)} 18 rules: {life, fruit,other, +16 items} rules: {fruit,pip, vegetables,other} rules: {citrus, tropical, +1 items} Items in LHS Group rules: {product,shopping, life} 2 rules: {fruit,tropical, fruit,pip rules: {misc., (appetizer)} 2 rules: {keeping, sanitary} rules: {salty, (appetizer)} rules



Size: support Color: lift

### RHS

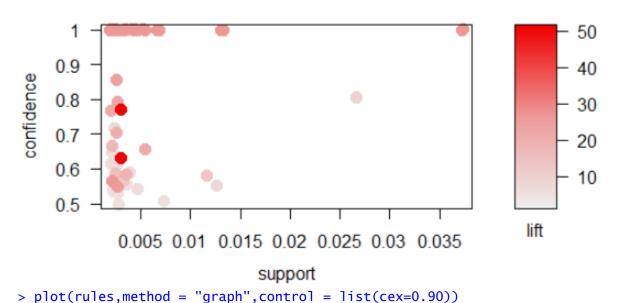
rules: {canned, water,bottled}

{milk, (appetizer)}

{snack} {beverages} {bakery} (life) {juice} (wine) {fruit,root} {cheese} {products} (vegetables, other) + 5 supressed

```
Available control parameters (with default values):
                              Grouped Matrix for 101 Rules
main
rhs_max =
                               10
lhs_items
aggr.fun
                                                                                                UseMethod("mean")
                                                  function (x, ...)
                                                                    "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF",
                              c("#EE0000FF", "#EE1212FF",
                                                                                                                                                                            "#EE0C0CFF"
co1
                                                                                                                                                                          "#EE1E1EFF".
 "#EE0F0FFF"
"#EE2222FF"
                                  "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF"
"#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF"
                                                                                                                                                                          "#EE3131FF".
                                  "#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF", "#EE3737FF", "#EE3A3AFF", "#EE3D3DFF", "#EE4040FF",
"#EE3434FF".
                                                                                                                                                                          "#EE4444FF".
"#EE4747FF", "#EE4A4AFF", "#EE4D4DFF", "#EE5D5DFF", "#EE5353FF",
                                                                                                                                                                          "#EE5656FF".
"#EE5959FF", "#EE5C5CFF", "#EE5F5FFF", "#EE6262FF", "#EE6666FF",
                                                                                                                                                                           "#EE6969FF".
#EE5959FF", "#EE5C5CFF", "#EE5F5FFF", "#EE6262FF", "#EE6C6CFF", "#EE6F6FFF", "#EE7272FF", "#EE7575FF", "#EE7575FF", "#EE8484FF", "#EE8888FF", "#EE8484FF", "#EE8888FF", "#EE8888FF", "#EE8484FF", "#EE8888FF", "#EE88FF", "#EE8888FF", "#EE888FF", "#EE888FF", "#EE88FF", "#EE88F
                                                                                                                                           "#EE7878FF"
                                                                                                                                                                             "#EE7B7BFF"
                                                                                                            "#EE8888FF",
                                                                                                                                              "#EE8B8BFF"
                                                                                                                                                                                "#EE8E8EFF
                                                                            "#EE9797FF"
         "#EE9191FF"
                                          "#EE9494FF"
                                                                                                              "#EE9999FF"
                                                                                                                                                "#EE9B9BFF"
                                                                                                                                                                                   "#EE9D9DF
           "#EE9F9FFF"
                                             "#EEAOAOFF"
                                                                               "#EEA2A2FF"
                                                                                                                 "#EEA4A4FF"
                                                                                                                                                   "#EEA5A5FF"
FF"
             "#EEA9A9FF"
                                               "#EEABABFF"
                                                                                  "#EEACACFF"
                                                                                                                   "#EEAEAEFF"
                                                                                                                                                     "#EEBOBOFF"
                                                                                    "#EEB7B7FF"
                                                  "#EEB5B5FF"
                                                                                                                      "#EEB8B8FF"
1FF"
                "#EEB3B3FF"
                                                                                                                                                        "#EEBABAFF"
                  "#EEBDBDFF"
                                                    "#EEBFBFFF",
                                                                                      "#EEC1C1FF", "
", "#EECBCBFF",
                                                                                                                         "#EEC3C3FF"
                                                                                                                                                           "#EEC4C4FF"
BCFF"
                       '#EEC8C8FF"
                                                            '#EEC9C9FF".
6C6FF"
                                                                                                                                "#EECDCDFF".
                                                                                                                                                                   #EECFCFFF".
                                                                                              "#EED5D5FF",
                                                            "#EED4D4FF"
EDODOFF"
                             #EED2D2FF".
                                                                                                                                 "#EED7D7FF"
                                                                                                                                                                  "#EED9D9FF"
                             "#EEDZDZFF", "#EED4D4FF", "#EED3D3FF", "#EED7D7FF", "#EED9D9FF", "#EEDCDCFF", "#EEE3E3FF", "#EEE7E7FF", "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF")
EEDBDBFF"
#EEE5E5FF"
reverse =
                              TRUE
xlab
                               NULL
ylab
                               NULL
                              Size: support Color: lift
legend
                      =
spacing =
                              -1
panel.function = function (row, size, shading, spacing) { size[size == 0] <- NA shading[is.na(shading)] <- 1 grid.circle(x = c(1:length(size )), y = row, r = size/2 * (1 - spacing), default.units = "native", gp = gpar(fill = shading, col = shading, alpha = 0.9)) } gp_main = list(cex = 1.2, fontface = "bold", font = c(bold = 2))
                                          = list(cex = 0.8)
gp_labels
                              list(cex = 1.2, fontface = "bold", font = c(bold = 2))
qp_1abs =
                                           = list(col = "gray", lty = 3)
gp_lines
newpage = TRUE
max.shading
engine
                      =
                               default
verbose =
                              FALSE
> plot(rules,method = "scatterplot",control = list(cex=0.90))
```

# Scatter plot for 101 rules



## Graph for 100 rules

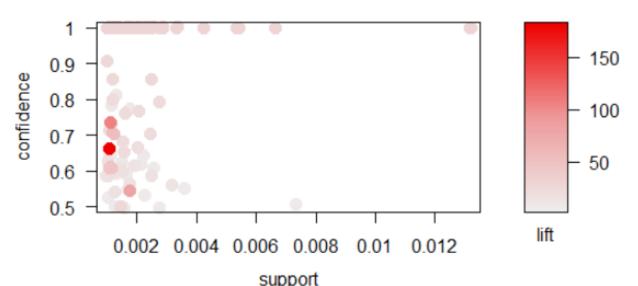
size: support (0.002 - 0.037) color: lift (5.231 - 51.465)

```
bar, siloppingspingspingvegetable
                                wine shopping O Obeer shopping juice
                                      spack, shopping
             whole
                                        Shopping
                            bags
                                                                  unned
                                                      water.bottled
                                                               sanitary
                                                                    0
snack, long
                                     citrus
                                                            keepinoducts
 beer long
                                         douit tropical
                                                             wine
beeored/blush
      produ
      product specialty tropy enetable
                                            milk, yourt, whipped
                                 Ó
                                           vegetables onions other
                       cheese
       milk,cream
                                              meat,root
                                omilk, butter, whipped/sour
      milk,yogurt,cream o o frozen
              domestic cream cream
cheese, vegetables, cream
> rules1 <- apriori(groceries_data,parameter = list(support=0.001,confidence=</pre>
0.5, \min_{n=3, \max} en=5)
Apriori
Parameter specification:
 confidence minval smax arem aval original Support maxtime support minlen
                     0.1
                               1 none FALSE
 maxlen target
                        ext
         5 rules FALSE
Algorithmic control:
 filter tree heap memopt load sort verbose 0.1 TRUE TRUE FALSE TRUE 2 TRUE
Absolute minimum support count: 9
set item appearances ...[0 item(s)] done [0.00s]. set transactions ...[6928 item(s), 9835 transaction(s)] done [0.05s].
sorting and recoding items ... [483 item(s)] done [0.00s]. creating transaction tree ... done [0.01s]. checking subsets of size 1 2 3 4 done [0.00s]. writing ... [186 rule(s)] done [0.00s]. creating S4 object ... done [0.01s].
> rules1
set of 186 rules
> rules_conf <- sort (rules1, by="confidence", decreasing=T)</pre>
> inspect(head(rules_conf))
      1hs
                                                   rhs
                                                                                 confidence lift
                                                               support
     {long,product}
{long,product}
{bakery,long}
{life,long}
                                                  {bakery} 0.001118454 1
                                                                                                26.72554
[1]
[2]
                                                  {life}
                                                               0.001118454 1
                                                                                                26.72554
 [3]
                                              => {life}
                                                               0.001830198 1
                                                                                                26.72554
     {life,long} => {bakery} 0.001830198 
{bakery,product,newspapers} => {life} 0.001220132
                                                                                                26.72554
                                                               0.001220132 1
                                                                                                26.72554
```

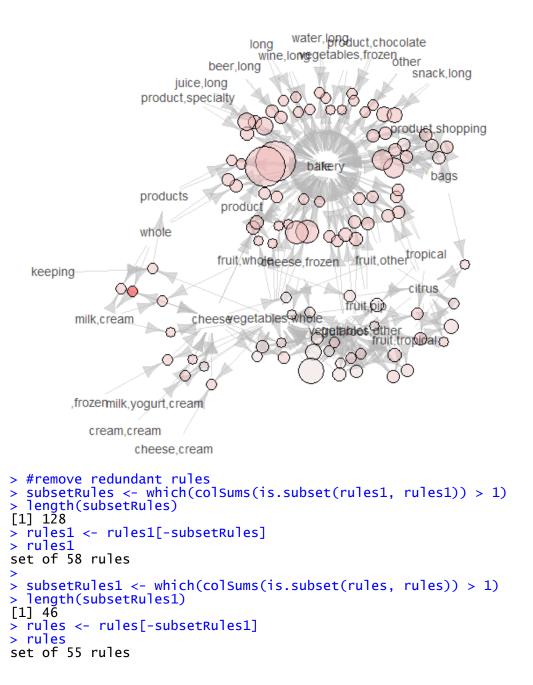
```
[6] {life,product,newspapers} => {bakery} 0.001220132 1
                                                                                                                      26.72554
       count
[1]
       11
[2]
[3]
       11
       18
[4]
[5]
      18
      12
Ī6Ī
      12
   rules_lift <- sort(rules1,by="lift",decreasing = T)</pre>
   inspect(head(rules_lift))
                                                        rhs
                                                                                            support
                                                                                                                 confidence
                                                         {products, shopping} 0.001016777 0.6666667
[1]
[2]
[3]
       {bags,keeping}
                                                         {beer,shopping}
                                                                                            0.001118454 0.7333333
        bags,canned}
                                                   =>
       {cheese, whole}
                                                         {milk,cream}
                                                                                            0.001728521 0.5483871
                                                   =>
[4]
       {bakery, water, long}
                                                        {product}
                                                                                            0.001220132 0.7058824
                                                   =>
       {life,water,long}
                                                        {product}
                                                                                            0.001220132 0.7058824
                                                   =>
                                                                                           0.001220132 0.7058824
       {bakery, life, water, long} => {product}
       lift
                         count
       182.12963 10
[1]
[2]
[3]
[4]
[5]
       106.06373
                        11
        79.31452
                        17
        53.40271 12
53.40271 12
        53.40271 12
[6]
> plot(rules1, method = "grouped",control = list(cex=0.90))
                                                                                             φ
                                            Grouped Matrix for 186 Rules
                                                                                             (milk,frozen, milk,yogurt,whipped/sour,
                                                                                   {fruit,root, cheese,frozen, +6 items}
                                                                          ±5 item
                                                                                        (beer,shopping, products,shopping,
                                                                              +3 items}
                                            7
                                   +6 items}
                                                                                                  {beer, hamburger, +18 items}
                                             cream,
                              items}
                                                      [keeping, vegetables,whole]
                                                                         (fruit,pip, vegetables,other,
                    +2 items}
                                                           {citrus, vegetables,other}
                                                                              (fruit,tropical, fruit,whole,
                                                                                                                      Size: support
                                                                (bakery, bar, +32 items)
 tems in LHS Group
                                            Ě
                                        +2 items}
                                                                                                                            Color: lift
                         Ŧ
                              Ŧ
                                                                    {life, bar, +32 items}
                                   juice,long,
                    water,long,
                         water,long,
                             water,long,
                                            (cheese,cream,
      rules: {keeping, bags}
                cheese
          {canned, bags}
                                                 ê
                                        citrus,
                                   (product,
                                                 fruit,root,
                {whole,
                    {long,
                         {long,
                              {long,
                                        (pip
                                                                                                  rules:
                                            rules
                                                                                    rules
           rules
                                   rules:
                                                                rules
                                                                     rules
                rules:
                    rules:
                         rules:
                              rules:
                                        rules:
                                                  rules:
                                                       ru es:
                                                           rules:
                                                                          rules:
                                                                               rules:
                                                                                        rules:
                                                                                   0
                                   7
                                            4
                                                                46
                                                                     4
                                                                                                  17
                                                                                                       RHS
                                                                                                     {products, shopping}
                                                                                                     {beer,shopping}
                                                                                                     {milk,cream}
                                                                                                     {product}
                                                                                                     {bakery}
                                                                                                     {life}
                                                                                                     {fruit,tropical}
                                                                                                     {fruit,root}
                                                                                                     {cheese}
                                                                                                     {products}
                                                                                                     + 6 supressed
Available control parameters (with default values):
main
                    Grouped Matrix for 186 Rules
                    20
k
                    10
rhs_max =
1hs_items
aggr.fun
                                 function (x, ...) UseMethod("mean")
```

```
çol
             c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF",
                                                                           "#EE0C0CFF"
                                                                          "#EE1E1EFF",
 #EE0F0FFF"
               "#EE2525FF",
                             "#EE2828FF",
                                            "#EE2B2BFF",
"#EE2222FF",
                                                           "#EE2E2EFF".
                                                                           "#EE3131FF".
               "#EE3737FF",
                              "#EE3A3AFF"
                                            "#EE3D3DFF",
                                                           "#EE4040FF",
"#EE3434FF"
                                                                          "#EE4444FF"
              "#EE4A4AFF",
                                            "#EE5050FF",
"#EE4747FF",
                             "#EE4D4DFF",
                                                           "#EE5353FF",
                                                                          "#EE5656FF"
"#EE5959FF".
              "#EE5C5CFF".
                             "#EE5F5FFF".
                                            "#EE6262FF",
                                                           "#EE6666FF".
                                                                          "#EE6969FF"
                                            "LEUZOZFF",
"#EE7575FF",
"#EE6C6CFF",
              "#EE6F6FFF", "#EE7272FF",
                                                              '#EE7878FF"
   #EE7E7EFÉ".
                  #EE8181FF",
                                "#EE8484FF"
                                               "#EE8888FF"
                                                              "#EE8B8BFF"
                  "#EE9494FF"
                                                "#EE9999FF"
                                                               "#EE9B9BFF"
   "#EE9191FF"
                                 "#EE9797FF"
                                                                              "#EE9D9DF
    "#EE9F9FFF"
                   "#EEAOAOFF"
                                  "#EEA2A2FF"
                                                 "#EEA4A4FF"
                                                                "#EEA5A5FF"
FF"
      '#EEA9A9FF"
                    "#EEABABFF",
                                   "#EEACACFF"
                                                  "#EEAEAEFF",
                                                                 "#EEBOBOFF"
                     "#EEB5B5FF"
                                    "#EEB7B7FF"
                                                   "#EEB8B8FF".
1FF"
       "#EEB3B3FF"
                                                                  "#EEBABAFF"
                       "#EEBFBFFF",
                                     "#EEC1C1FF", "
", "#EECBCBFF",
BCFF"
        "#EEBDBDFF"
                                                    "#EEC3C3FF",
                                                                   "#EEC4C4FF"
                         "#EEC9C9FF",
                                                       "#EECDCDFF".
6C6FF"
         "#EEC8C8FF"
                                                                      "#EECFCFFF".
                                         "#EED5D5FF"
                          "#EED4D4FF"
                                                        "#EED7D7FF"
EDODOFÉ"
            "#EED2D2FÉ"
                                                                       "#EED9D9FF"
            "#EEDZDZFF, #EED4D4FF, #EED3D3FF, #EED7D7FF, "
"#EEDCDCFF", "#EEDEDEFF", "#EEE0E0FF", "#EEE1E1FF", 
"#EEE7E7FF", "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF",
                                                                       "#EEE3E3FF"
#EEE5E5FF",
                                                                        , "#EEEEEEFF")
reverse
             TRUE
xlab
             NULL
ylab
             NULL
         =
             Size: support Color: lift
legend
spacing
             -1
)), y = row, r = size/2 * (1 - spacing), default.units = "native", gp = gpar(fill = shading, col = shading, alpha = 0.9)) }
gp_main = list(cex = 1.2, fontface = "bold", font = c(bold = 2))
gp_labels
                  = list(cex = 0.8)
             gp_labs =
gp_lines
newpage =
max.shading
engine =
             default
verbose =
             FALSE
> plot(rules1,method = "scatterplot",control = list(cex=0.90))
```

# Scatter plot for 186 rules



> plot(rules1,method = "graph",control = list(cex=0.90))



### **Rcode:**

#Different set of rule values for my\_movies Dataset using apriori algorithm.
mymovies <- read.transactions(file.choose(),format = "basket")
View(mymovies)</pre>

summary(mymovies)

```
str(mymovies)
class(mymovies)
#to see most frequent items
FrequentItem <- eclat(mymovies,parameter = list(support=0.07,maxlen=15))
inspect(FrequentItem)
itemFrequencyPlot(mymovies, topN=5, type="absolute", main="Item Frequency")
dev.off()
rules3 <- apriori(mymovies,parameter =
list(support=0.002,confidence=0.5,minlen=2,maxlen=5))
rules3
inspect(head(sort(rules3,by="lift"),n=15))
inspect(tail(sort(rules3,by="lift"),n=15))
plot(rules3)
quality(head(rules3))
plot(rules3, method = "grouped",control = list(cex=0.90))
plot(rules3,method = "scatterplot",control = list(cex=0.90))
plot(rules3,method = "graph",control = list(cex=0.90))
mymovies1 <- read.csv(file.choose())</pre>
names(mymovies1)
summary(mymovies1)
attach(mymovies1)
sd(Sixth.Sense)
sd(Gladiator)
sd(LOTR1)
sd(Harry.Potter1)
```

```
sd(Patriot)
sd(LOTR2)
sd(Harry.Potter2)
sd(LOTR)
sd(Braveheart)
sd(Green.Mile)
var(Sixth.Sense)
var(Gladiator)
var(LOTR1)
var(Harry.Potter1)
var(Patriot)
var(LOTR2)
var(Harry.Potter2)
var(LOTR)
var(Braveheart)
var(Green.Mile)
library(moments)
skewness(Sixth.Sense)
skewness(Gladiator)
skewness(LOTR1)
skewness(Harry.Potter1)
skewness(Patriot)
skewness(LOTR2)
skewness(Harry.Potter2)
skewness(LOTR)
skewness(Braveheart)
skewness(Green.Mile)
```

```
kurtosis(Sixth.Sense)
kurtosis(Gladiator)
kurtosis(LOTR1)
kurtosis(Harry.Potter1)
kurtosis(Patriot)
kurtosis(LOTR2)
kurtosis(Harry.Potter2)
kurtosis(LOTR)
kurtosis(Braveheart)
kurtosis(Green.Mile)
rules2 <- apriori(as.matrix(mymovies1[,6:15],parameter=list(support=0.2,
confidence = 0.5,minlen=5)))
rules2
rules_conf <- sort (rules2, by="confidence", decreasing=T)
inspect(head(rules_conf))
rules_lift <- sort(rules2,by="lift",decreasing = T)</pre>
inspect(head(rules_lift))
plot(rules2, method = "grouped",control = list(cex=0.90))
plot(rules2,method = "scatterplot",control = list(cex=0.90))
plot(rules2,method = "graph",control = list(cex=0.90))
#remove redundant rules
subsetRules <- which(colSums(is.subset(rules2, rules2)) > 1)
length(subsetRules)
rules2 <- rules2[-subsetRules]</pre>
rules2
```

```
subsetRules1 <- which(colSums(is.subset(rules3, rules)) > 1)
length(subsetRules1)
rules3 <- rules3[-subsetRules1]
rules3
Console:
> #Different set of rule values for my_movies Dataset using apriori algorithm
 mymovies <- read.transactions(file.choose(), format = "basket")</pre>
  View(mymovies)
  Q mymovies ×
                                     mymovies ×

□ Untitled1* ×
  Show Attributes
                                                                                         Q
                                                              Value
 Name
                                Type
  mymovies
                                S4 [11 x 22] (arules::transactions) S4 object of class transactions
    data
                                                              S4 object of class ngCMatrix
                                S4 [22 x 11] (Matrix::ngCMatrix)
    itemInfo
                               list [22 x 1] (S3: data.frame)
                                                              A data frame with 22 rows and 1 column
     itemsetInfo
                                                              A data frame with 0 rows and 0 columns
                               list [0 x 0] (S3: data.frame)
> summary(mymovies)
transactions as itemMatrix in sparse format with
 11 rows (elements/itemsets/transactions) and
 22 columns (items) and a density of 0.1404959
most frequent items:
                              Gladiator
                                                                ,"Patriot","Sixth
Sense","","",1,1,0,0,1,0,0,0,0,0
                                                                        Sixth Sense
element (itemset/transaction) length distribution:
sizes
2 3 4 6
3 6 1 1
                                   Mean 3rd Qu.
   Min. 1st Qu.
                      Median
                                                         Max.
             2.500
                       3.000
                                  3.091
                                             3.000
                                                        6.000
includes extended item information - examples:
                             labels
    LOTR","Gladiator","Green,
LOTR1","Harry,"
> str(mymovies)
Formal class 'transactions' [package "arules"] with 3 slots
..@ data :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
.....@ i : int [1:34] 7 11 15 17 19 21 2 13 14 20 ...
.....@ p : int [1:12] 0 6 10 12 14 17 20 23 26 28 ...
.....@ Dim : int [1:2] 22 11
```

```
.. .. ..@ Dimnames:List of 2
   .....$: NULL
.....@ factors : list()
..@ itemInfo :'data.frame': 22 obs. of 1 variable:
....$ labels: chr [1:22] ",\"Harry" ",\"LOTR\",\"Gladiator\",\"Green" ",\"
LOTR1\",\"Harry" ",\"LOTR2\",\"\",\"\",0,0,1,0,0,1,0,0,0,0" ...
..@ itemsetInfo:'data.frame': 0 obs. of 0 variables
> class(mymovies)
[1] "transactions"
attr(,"package")
[1] "arules"
> #to see most frequent items
> FrequentItem <- eclat(mymovies,parameter = list(support=0.07,maxlen=15))</pre>
Eclat
parameter specification:
  tidLists support minlen maxlen
                                                                        target
                     0.07
                                               15 frequent itemsets FALSE
                                      1
algorithmic control:
 sparse sort verbose
                -2
                           TRUE
Absolute minimum support count: 0
create itemset ...
set transactions ...[22 item(s), 11 transaction(s)] done [0.00s]. sorting and recoding items ... [22 item(s)] done [0.00s]. creating sparse bit matrix ... [22 row(s), 11 column(s)] done [0.00s]. writing ... [105 set(s)] done [0.00s]. Creating S4 object ... done [0.00s].
> inspect(FrequentItem)
          items
                                                                                                      support count
          {,"Patriot","","","",0,1,0,0,1,0,0,0,0,0,
[1]
          Gladiator} {,"LOTR2","","",",0,0,1,0,0,1,0,0,0,0,
                                                                                                 0.09090909
                                                                                                                          1
[2]
            LOTR1}
                                                                                                 0.09090909
                                                                                                                          1
          {,"Patriot","Braveheart","",0,1,0,0,1,0,0,1,0,0
[3]
                                                                                                 0.09090909
            Gladiator}
                                                                                                                          1
          {,"LOTR","Gladiator","Green,
Mile","",1,1,0,0,0,0,0,1,0,1,
Sixth Sense}
{Mile","",1,1,0,0,0,0,0,1,0,1,
Sixth Sense}
[4]
                                                                                                 0.09090909
                                                                                                                          1
[5]
                                                                                                 0.09090909
                                                                                                                          1
          {,"LOTR","Gladiator","Green,
Mile","",1,1,0,0,0,0,0,1,0,1}
{,"LOTR","Gladiator","Green,
F61
                                                                                                 0.09090909
                                                                                                                          1
[7]
            Sixth Sense}
                                                                                                 0.09090909
                                                                                                                          1
          {,"Harry,
[8]
           Harry Potter1, Potter2","",",0,0,0,1,0,0,1,0,0,0}
                                                                                                 0.09090909
                                                                                                                          1
         {,"Harry,
Potter2","","",0,0,0,1,0,0,1,0,0,0}
{Harry Potter1,
Potter2","","",0,0,0,1,0,0,1,0,0,0,0}
[9]
                                                                                                 0.09090909
                                                                                                                          1
Γ10]
                                                                                                 0.09090909
                                                                                                                          1
          {,"Harry,
[11]
          Harry Potter1}
{,"LOTR1","Harry,
Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,
Potter1","Green,
                                                                                                 0.09090909
                                                                                                                          1
[12]
          Sixth Sense}
{Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,
Potter1","Green,
                                                                                                 0.09090909
                                                                                                                          1
Γ13]
            Sixth Sense}
                                                                                                 0.09090909
                                                                                                                          1
          {,"LOTR1","Harry,
Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,
Potter1","Green}
[14]
                                                                                                 0.09090909
                                                                                                                          1
```

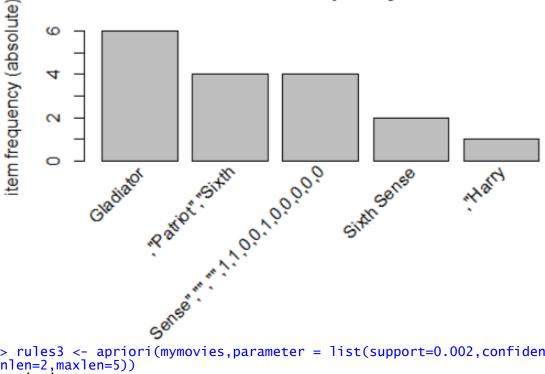
[15]	<pre>{,"LOTR1","Harry, Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,</pre>		
[16]	Sixth Sense} {Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,	0.09090909	1
[17]	Sixth Sense} {,"LOTR1","Harry,	0.09090909	1
	Mile" "LOTP2" 1 0 1 1 0 1 0 0 1 1	0.09090909	1
[18]	{Mile","LOTR2",1,0,1,1,0,1,0,0,0,1, Potter1","Green} {,"LOTR1","Harry,	0.09090909	1
[19]	Potterl","Green,		
[20]	Sixth Sense} {Potter1","Green,	0.09090909	1
[21]	Sixth Sense} {,"LOTR1","Harry, Potter1","Green}	0.09090909	1
[22]	Potter1","Green} {,"LOTR1","Harry,	0.09090909	1
	Sixth Sense}	0.09090909	1
[23]	{,"V2","V3", <sup>"</sup> V4","V5","Sixth, Mile",		
	Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry,		
[24]	V1} {,"V2","V3","V4","V5","Sixth,	0.09090909	1
	Mile", Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green,		
[25]	Sense", "Gladiator", "LOTR1", "Harry} {Mile",	0.09090909	1
[23]	Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry,		
[26]	V13	0.09090909	1
[20]	{,"V2","V3","V4","V5","Sixth, Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green,		
	Sense","Gladiator","LOTR1","Harry,	0.09090909	1
[27]	<pre>{,"V2","V3","V4","V5","Sixth, Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry}</pre>	0.03030303	
	Potter1 , Patriot , LOIR2 , Harry, Potter2","LOTR","Braveheart","Green,		_
[28]	{Potter1","Patriot","LOTR2","Harry,	0.09090909	1
	Potter2", "LOTR", "Bŕavehearť", "Gréén, Sense", "Gladiator", "LOTR1", "Harry,		
[29]	V1} {Mile",	0.09090909	1
[20]	Potter1", "Patriot", "LOTR2", "Harry, Potter2", "LOTR", "Braveheart", "Green, Sense", "Gladiator", "LOTR1", "Harry} {, "V2", "V3", "V4", "V5", "Sixth,	0.09090909	1
[30]	MITE,		
	Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry,		
[31]	V1} {."V2"."V3"."V4"."V5"."Sixth.	0.09090909	1
	{,"V2","V3","V4","V5","Sixth, Mile", Potter2","LOTR","Braveheart","Green,		
[32]	Sense", "Gladiator", "LOTR1", "Harry} {Mile",	0.09090909	1
	Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry,		
רככי	V1}	0.09090909	1
[33]	{,"V2","V3","V4","V5","Sixth, Potter2","LOTR","Braveheart","Green,		
	Sense","Gladiator","LOTR1","Harry, V1}	0.09090909	1

[34]	<pre>{,"v2","v3","v4","v5","sixth, Potter2","LOTR","Braveheart","Green,</pre>		
[35]	Potterz , LOIR , Braveneart , Green, Sense","Gladiator","LOTR1","Harry} {Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry,	0.09090909	1
[36]	V1} {Mile",	0.09090909	1
[37]	Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry} {Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green,	0.09090909	1
[38]	Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry} {,"V2","V3","V4","V5","Sixth, Mile",	0.09090909	1
	Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green, V1}	0.09090909	1
[39]	{,"ṽ2","v3","v4","v5","sixth, Mile",		
[40]	Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green} {Mile",	0.09090909	1
[41]	Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green, V1} {."V2"."V3"."V4"."V5"."Sixth.	0.09090909	1
	{, v2", "v3", "v4", "v5", "Sixth, Potter1", "Patriot", "LOTR2", "Harry, Potter2", "LOTR", "Braveheart", "Green, v1}	0.09090909	1
[42]	<pre>{,"V2","V3","V4","V5","Sixth, Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green}</pre>	0.09090909	1
[43]	{Potter1", "Patriot", "LOTR2", "Harry, Potter2", "LOTR", "Braveheart", "Green, V1}	0.09090909	1
[44]		0.09090909	1
[45]	<pre>{Mile", Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green} {,"V2","V3","V4","V5","Sixth, Mile", Potter2" "LOTR" "Braveheart" "Creen</pre>	0.09090909	1
[46]	Potter2","LOTR","Braveheart","Green, V1} {,"V2","V3","V4","V5","Sixth, Mile",	0.09090909	1
F 4 7 7	Potter2","LOTR","Braveneart","Green}	0.09090909	1
[47] [48]	<pre>{Mile", Potter2","LOTR","Braveheart","Green, V1} {."V2"."V3"."V4"."V5"."Sixth.</pre>	0.09090909	1
[]	<pre>{,"V2","V3","V4","V5","Sixth, Potter2","LOTR","Braveheart","Green, V1}</pre>	0.09090909	1
[49]	<pre>{,"V2","V3","V4","V5","Sixth, Potter2","LOTR","Braveheart","Green} {Potter2","LOTR","Braveheart","Green,</pre>	0.09090909	1
[50]	V1}	0.09090909	1
[51]	<pre>{Mile", Potter2","LOTR","Braveheart","Green}</pre>	0.09090909	1
[52]	<pre>Potter2 , LOTR , Braveneart , Green} {Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green} {Potter2","LOTR","Braveheart","Green, Sense" "Gladiator" "LOTR1" "Harry}</pre>	0.09090909	1
[53]	{Potter2","LOTR","Braveheart","Green, Sense"."Gladiator"."LOTR1"."Harry}	0.09090909	1
[54]	{Potter2","LOTR","Braveheart","Green, Sense","Gladiator","LOTR1","Harry} {,"V2","V3","V4","V5","Sixth, Mile", Potter1","Patriot","LOTR2","Harry,		_
[55]	Potter1","Patriot","LOTR2","Harry, Sense","Gladiator","LOTR1","Harry, V1} {,"V2","V3","V4","V5","Sixth, Mile",	0.09090909	1

[[C]	Potter1","Patriot","LOTR2","Harry, Sense","Gladiator","LOTR1","Harry}	0.09090909	1
[56]	<pre>{Mile", Potter1","Patriot","LOTR2","Harry, Sense","Gladiator","LOTR1","Harry,</pre>		
[57]	V11	0.09090909	1
	{,"V2","V3","V4","V5","Sixth, Potter1","Patriot","LOTR2","Harry, Sense","Gladiator","LOTR1","Harry, V1}	0.09090909	1
[58]	{,"V2","V3","V4","V5","Sixth,		_
[59]	Sense", "Gladiator", "LOTR1", "Harry} {Potter1", "Patriot", "LOTR2", "Harry, Sense", "Gladiator", "LOTR1", "Harry,	0.09090909	1
[60]	V1} {Mila"	0.09090909	1
[61]	Potter1","Patriot","LOTR2","Harry, Sense","Gladiator","LOTR1","Harry} {,"V2","V3","V4","V5","Sixth, Mile",	0.09090909	1
	Sense","Gladiator","LOTR1","Harry,	0.09090909	1
[62]	{,"V2","V3","V4","V5","Sixth, Mile",	0.0000000	4
[63]	Sense","Gladiator","LOTR1","Harry} {Mile", Sense","Gladiator","LOTR1","Harry,	0.09090909	1
[64]	V1} {,"V2","V3","V4","V5","Sixth,	0.09090909	1
[6E]	Sense","Gládiatór","ĹOTR1", "Harry, V1} [ "V2" "V2" "V4" "V5" "six+h	0.09090909	1
[65]	<pre>{,"V2","V3","V4","V5","Sixth, Sense","Gladiator","LOTR1","Harry} {Sense","Gladiator","LOTR1","Harry,</pre>	0.09090909	1
[66]	V1}	0.09090909	1
[67]	{Mile", Sense","Gladiator","LOTR1","Harry}	0.09090909	1
[68]	{Potter1","Patriot","LOTR2","Harry,	0.09090909	1
[69]	{,"V2","V3","V4","V5","Sixth, Mile",	0.0303030	_
[70]	Potter1","Patriot","LOTR2","Harry, V1} {,"V2","V3","V4","V5","Sixth, Mile",	0.09090909	1
[71]	Potter1","Patriot","LOTR2","Harry} {Mile",	0.09090909	1
	Potter1","Patriot","LOTR2","Harry, V1}	0.09090909	1
[72]	{,"V2","V3","V4","V5","Sixth, Potter1","Patriot","LOTR2","Harry,		
[73]	V1} {,"V2","V3","V4","V5","Sixth,	0.09090909	1
[74]	{,"V2","V3","V4","V5","Sixth, Potter1","Patriot","LOTR2","Harry} {Potter1","Patriot","LOTR2","Harry,	0.09090909	1
[75]	V1} {Mile",	0.09090909	1
[76]	Potter1","Patriot","LOTR2","Harry} {."V2"."V3"."V4"."V5"."Sixth.	0.09090909	1
	Mile", V1}	0.09090909	1
[77]	{,"V2","V3","V4","V5","Sixth, Mile"}	0.09090909	1
[78]	{Mile", V1}	0.09090909	1
[79]	{, "v2", "v3", "v4", "v5", "sixth, v1}	0.09090909	1
	· <b>,</b>	2.23030303	-

```
[80]
              {, "Patriot", "Sixth,
             {, Fair Tot , Sixten,
Gladiator,
Sense","",",1,1,0,0,1,0,0,0,0,0)
{,"Patriot","Sixth,
Gladiator}
{,"Patriot","Sixth,
Sense","","",1,1,0,0,1,0,0,0,0,0)
}
                                                                                                                                   0.36363636
                                                                                                                                                                     4
 [81]
                                                                                                                                   0.36363636
                                                                                                                                                                     4
 [82]
                                                                                                                                   0.36363636
                                                                                                                                                                     4
              {Gladiator, Sense","",",1,1,0,0,1,0,0,0,0,0}
 [83]
                                                                                                                                   0.36363636
                                                                                                                                                                     4
              {Gladiator}
{Sense","",",1,1,0,0,1,0,0,0,0,0}
{,"Patriot","Sixth}
                                                                                                                                   0.54545455
 [84]
                                                                                                                                                                     6
  [85]
                                                                                                                                   0.36363636
                                                                                                                                                                     4
                                                                                                                                                                     4
 [86]
                                                                                                                                   0.36363636
              {Sixth Sense}
{,"V2","V3","V4","V5","Sixth}
                                                                                                                                                                     2
 [87]
                                                                                                                                   0.18181818
                                                                                                                                                                     1
 Ī88Ī
                                                                                                                                   0.09090909
 [89]
              {V1}
                                                                                                                                   0.09090909
                                                                                                                                                                    1
              {Miĺe"}
                                                                                                                                                                     1
  ۲90٦
                                                                                                                                   0.09090909
             {Potter1", "Patriot", "LOTR2", "Harry}
{Sense", "Gladiator", "LOTR1", "Harry}
{Potter2", "LOTR", "Braveheart", "Green}
{, "LOTR1", "Harry}
{Potter1", "Green}
 [91]
                                                                                                                                                                     \bar{1}
                                                                                                                                   0.09090909
 [92]
                                                                                                                                  0.09090909
                                                                                                                                                                    1
                                                                                                                                                                     1
  Г931
                                                                                                                                   0.09090909
                                                                                                                                                                     1
  [94]
                                                                                                                                   0.09090909
                                                                                                                                                                     1
 [95]
                                                                                                                                   0.09090909
              {Mile","LOTR2",1,0,1,1,0,1,0,0,0,1}
{,"Harry}
                                                                                                                                                                     1
 [96]
                                                                                                                                   0.09090909
  Ī97Ī
                                                                                                                                                                     1
                                                                                                                                   0.09090909
[97] {,"Harry}
[98] {Harry Potter1}
[99] {Potter2","","","",0,0,0,1,0,0,1,0,0,0}
[100] {,"LOTR","Gladiator","Green}
[101] {Mile","",1,1,0,0,0,0,0,1,0,1}
[102] {,"Patriot","Braveheart","","",0,1,0,0,1,0,0,0,1,0}
[103] {,"LOTR2","","","",0,0,1,0,0,0,0,0,0}
[104] {LOTR1}
[105] {,"Patriot","","","",0,1,0,0,1,0,0,0,0,0}
[105] {,"Patriot","","","",0,1,0,0,1,0,0,0,0,0}
[106] > itemFrequencyPlot(mymovies, topN=5, type="absolute", main="Item Frequency")
```

# Item Frequency



> rules3 <- apriori(mymovies,parameter = list(support=0.002,confidence=0.5,mi</pre> nlen=2,maxlen=5)) Apriori

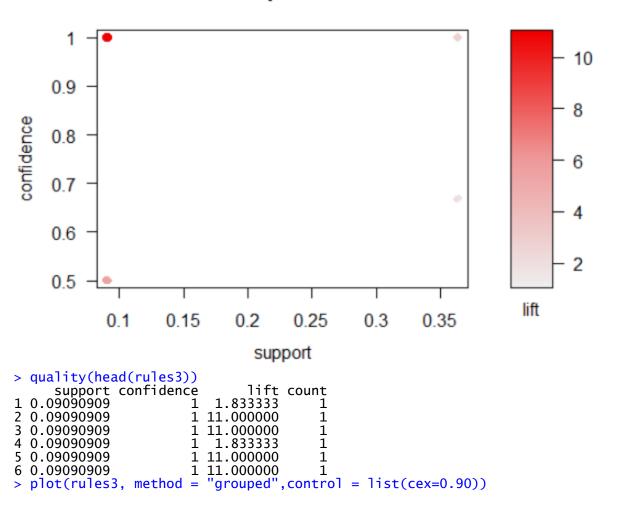
```
Parameter specification:
confidence minval smax arem aval original Support maxtime support minlen
              0.1
        0.5
                     1 none FALSE
                                              TRUE
maxlen target ext
```

#### 5 rules FALSE

```
Algorithmic control:
  filter tree heap memopt load sort verbose
      0.1 TRUE TRUE FALSE TRUE
Absolute minimum support count: 0
set item appearances _...[0 item(s)] done [0.00s].
set transactions ...[22 item(s), 11 transaction(s)] done [0.00s]. sorting and recoding items ... [22 item(s)] done [0.00s]. creating transaction tree ... done [0.00s]. checking subsets of size 1 2 3 4 5 done [0.00s]. writing ... [239 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> rules3
set of 239 rules
 > inspect(head(sort(rules3,by="lift"),n=15))
                                                                 rhs
support confidence lift count
                                                             => {,"LOTR2","","","",0,0,1,0,0,
 [1]
      {LOTR1}
[3] {,"LOTR","Gladiator"
} 0.09090909
                                                             => {Mile","",1,1,0,0,0,0,0,1,0,1
[4] {Mile","",1,1,0,0,0,0,0,1,0,1}
0.09090909 1 11
                                                         1
                                                             => {,"LOTR","Gladiator","Green}
 0.09090909
                                                             => {Potter2","","","",0,0,0,1,0,
                                                             => {Potter2","","","",0,0,0,1,0,
 \bar{0}, \bar{1}, 0, \bar{0}, 0} 0.09090909
                                                11
 [9] {Harry Potter1}
                                                             => {,"Harry}
0.09090909
[10] {,"Harry}
0.09090909
                                 11
                                           1
                                                             => {Harry Potter1}
                                           1
[11] {,"LOTR1","Harry} 0,0,1} ___0.09090909
                                                             => {Mile","LOTR2",1,0,1,1,0,1,0,
                                           1
                                                11
                                                         1
 [12] {Mile","LOTR2",1,0,1,1,0,1,0,0,0,1}
                                                             => {,"LOTR1","Harry}
0.09090909
                                 11
                                          1
 [13] {,"LOTR1","Harry}
                                                             => {Potter1", "Green}
0.09090909
                                 11
                                           1
 [14] {Potter1", "Green}
                                                             => {,"LOTR1","Harry}
0.09090909
                                 11
                           1
 [15] {Mile","LOTR2",1,0,1,1,0,1,0,0,0,1}
                                                             => {Potter1", "Green}
0.09090909
 > inspect(tail(sort(rules3,by="lift"),n=15))
Ihs
support confidence lift count
[1] {,"LOTR1","Harry,
    Mile","LOTR2",1,0,1,1,0,1,0,0,0,1}
0.09090909 1.0000000 5.500000 1
[2] {,"LOTR1","Harry,
    Potter1","Green}
0.09090909 1.0000000 5.500000 1
[3] {Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,
    Potter1","Green}
0.09090909 1.0000000 5.500000 1
[4] {,"LOTR1","Harry,
    Mile","LOTR2",1,0,1,1,0,1,0,0,0,1,
    Potter1","Green}
       Ìhs
                                                                                  rhs
                                                                             => {Sixth Sense}
                                                                             => {Sixth Sense}
                                                                             => {Sixth Sense}
Potter1", "Green}
0.09090909 1.0000000 5.500000
                                                                             => {Sixth Sense}
 [5] {,"Patriot","Sixth}
,0,0,1,0,0,0,0,0} 0.36363636 1.0000000 2.750000
                                                                             => {Sense","","",1,1
```

```
{Sense","",1,1,0,0,1,0,0,0,0,0,0}
0.36363636 1.0000000 2.750000
                                                                                                                 => {,"Patriot","Sixt
 [6]
h̄} ¯
[7]
                                                                                                          4
          {,"Patriot","Sixth,
  Gladiator}
                                                                                                                  => {Sense","","",1,1
  0,0,1,0,0,0,0,0,0 0.36363636 1.0000000 2.750000
                                                                                                          4
[8] {Gladiator,
	Sense","","",1,1,0,0,1,0,0,0,0,0} => {,"Patriot",
	h} 0.36363636 1.00000000 2.750000 4
[9] {,"Patriot","","",0,1,0,0,1,0,0,0,0,0} => {Gladiator}
0.09090909 1.0000000 1.833333 1
[10] {,"Patriot","Braveheart","","",0,1,0,0,1,0,0,0,1,0} => {Gladiator}
0.09090909 1.0000000 1.833333 1
=> {Gladiator}
                                                                                                                  => {,"Patriot","Sixt
[11] {,"Patriot","Sixth} 0.36363636 1.0000000 1.833333
[12] {Gladiator}
                                                                                                                 => {,"Patriot","Sixt
h} 0.36363636 0.6666667 1.833333
[13] {Sense","","",1,1,0,0,1,0,0,0,0,0}
0.36363636 1.0000000 1.833333 4
                                                                                                          4
                                                                                                                  => {Gladiator}
[14] {Gladiator}
,0,0,1,0,0,0,0} 0.36363636 0.6666667 1.833333
[15] {,"Patriot","Sixth,
Sense","","",1,1,0,0,1,0,0,0,0,0}
0.36363636 1.0000000 1.833333 4
                                                                                                                 => {Sense","","",1,1
                                                                                                          4
                                                                                                                 => {Gladiator}
> plot(rules3)
```

# Scatter plot for 239 rules

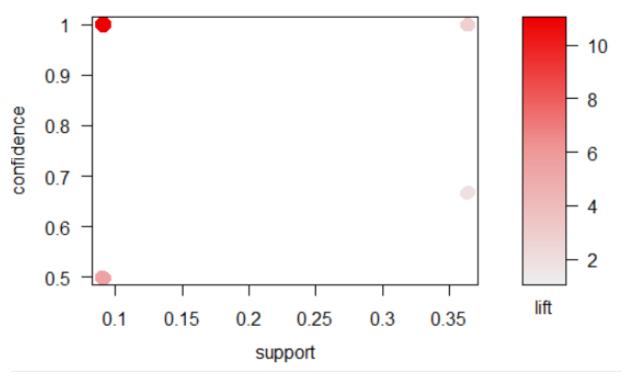


```
rules: {,"V2","V3","V4","V5","Sixth, Sense","Gladiator","LOTR1","Harry, +3 item
                                               rules: {Potter1","Patriot","LOTR2","Harry, Potter2","LOTR","Braveheart","Green
                                                             ,"Sixth, Potter1","Patriot","LOTR2","Harry, +2 items
                                                                              Sense","Gladiator","LOTR1","Harr
                                                                                            Grouped Matrix for 239 Rules
                                                                                                     rules: {,"\2","\3","\4","\5","Sixth, Potter2","LOTR","Braveheart","Gre
                                                                                                                     rules: {Sense","Gladiator","LOTR1","Harry, Potter1","Patriot","LOTR2'
                                                                                                                             items}
                                                                                                                             8 rules: {,"LOTR1","Harry, Mile","LOTR2",1,0,1,1,0,1,0,0,0,1, +1
                                                                                                                                     {Mile","LOTR2",1,0,1,1,0,1,0,0,0,1, Potter1","Green, +1
                                                                      rules: {Mile", Potter2","LOTR","Braveheart","Green, +2 items}
                                                                                                                                                                                                                                                Size: support
                                                                                                              rules: {Mile", Potter1","Patriot","LOTR2","Harry, +1 items}
                                                                                                                                                                                                                                                           Color: lift
                       {V1, Potter1", "Patriot", "LOTR2", "Harry, +3 items}
               1 rules: {,"\2","\3","\4","\5","Sixth, Mile", +3 items}
                                                                                             9 rules: {,"V2","V3","V4","V5","Sixth, Mile", +4 items}
                                                                                      rules: {,"\\2","\3","\4","\5","Sixth, \\1, +2 items}
                                                                                                                                            {,"LOTR","Gladiator","Green, Sixth Sense
                                                                                                                                                    3 rules: {Mile","",1,1,0,0,0,0,1,0,1, Sixth Sense}
                                                                                                                                                             Sense, +1 items}
                                                                              rules: {Potter2","LOTR","Braveheart","Green,
                                                                                                                                                                    rules: {,"Patriot","Sixth, Gladiator, +10 items}
                                                      9 rules: {,"Harry, Harry Potter1, +1 items}
                                                              rules: {,"V2","V3","V4","V5",
                               (Mile", V1, +2
                                                                                                                                     8 rules: {
                                                                                                                                            3 rules:
                                                                                                                                                                          {"LOTR2","","","",0,0,1,0,0,1,0,0,0,0}
{LOTR1}
{Larry Potter1}
{"terzy,",",",",0,0,0,1,0,0,1,0,0,0}
{Sense","Gladiator","LOTR1" "'
{Potter2" "'
                                                                                                              9
                                                                                                                     7
                                                                                                                                                                                 '1}
otter2","LOTR","Braveheart","Green}
otter1","Patriot","LOTR2","Harry}
                                                                                                                                                                            (Mile")
+ 10 supressed
Available control parameters (with default values):
                            = Grouped Matrix for 239 Rules
main
                                       20
 k
                             =
                                       10
 rhs_max =
 lhs_items
                                      = function (x, ...) UseMethod("mean")
c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0COCFF"
, "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF", "#EE1E1EFF",
, "#EE2525FF", "#EE2828FF", "#EE282BFF", "#EE282EFF", "#EE3131FF",
aggr.fun
 col
                                                                                                                                                                                                                         #EEUCUCFF",
"#EE1E1EFF".
   #EE0F0FFF"
"#EE2222FF".
                                                                                    #EEZ8Z8FF", "#EE2B2BFF",
"#EE3A3AFF", "#EE3D3DFF",
"#EE4D4DFF", "#EE5050FF",
"#EE5F5FFF", "#EE6262FF",
"#EE7272FF", "#EE7575FF",
"#FF8484EF" "#FF8484EF"
                                           #EEZ3Z3FF",
"#EE3737FF",
                                                                                                                                                                           #EEZEZEFF",
"#EE4040FF",
"#EE5353FF",
"#EE6666FF",
"#EE3434FF",
                                                                                                                                                                                                                       "#EE4444FF",
"#EE3434FF", "#EE3737FF", "
"#EE4747FF", "#EE4A4AFF", "
"#EE5959FF", "#EE5C5CFF", "
"#EE6C6CFF", "#EE6F6FFF", "
, "#EE72FFF", "#EE8181FF",
                                                                                                                                                                                                                         "#EE5656FF",
                                                                                                                                                                                                                         "#EE6969FF",
                                                                                                                                                                                 "#EE7878FF"
                                                                                                                                                                                                                             "#EE7B7BFF"
                                                                                             "#EE8484FF",
"#EE9797FF"
                                                                                                                                         "#EE8888FF",
                                                                                                                                                                                     "#EE8B8BFF"
                                                                                                                                            "#EE9999FF"
                                                     "#EE9494FF"
                                                                                                                                                                                       "#EE9B9BFF"
          "#EE9191FF"
                                                                                                                                                                                                                                    "#EE9D9DF
                                                        "#EEAOAOFF",
"#EEABABFF",
                                                                                                                                               "#EEA4A4FF",
"#EEAEAEFF"
                                                                                                                                                                                           "#EEA5A5FF",
"#EEB0B0FF"
             "#EE9F9FFF",
"#EEA9A9FF",
                                                                                                    "#EEA2A2FF",
"#EEACACFF",
FF"
 1FF"
                     "#EEB3B3FF"
                                                                "#EEB5B5FF",
                                                                                                           "#EEB7B7FF",
                                                                                                                                                      "#EEB8B8FF"
                                                                                                                                                                                                  "#EEBABAFF"
                                     33B3FF", "#EEB5B5FF", "#EEB7B7FF", "#EEB8B8FF", "#EEBABAFF", "#EEB
BDBDFF", "#EEBFBFFF", "#EEC1C1FF", "#EEC3C3FF", "#EEC4C4FF", "#EE
EC8C8FF", "#EEC9C9FF", "#EECBCBFF", "#EECDCDFF", "#EECFCFFF", "#
'#EED2D2FF", "#EED4D4FF", "#EED5D5FF", "#EED7D7FF", "#EED9D9FF", "
"#EEDCDCFF", "#EEDEDEFF", "#EEE0E0FF", "#EEE1E1FF", "#EEE3E3FF",
"#EEE7E7FF", "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF")
                                                                                                                                                                                                           #EEC4C4FF", "#EEC
"#EECFCFFF", "#E
"#EED9D9FF", "#
', "#EEE3E3FF", "
BCFF"
                        "#EEBDBDFF"
6C6FF"
                           "#EEC8C8FF",
', "#EED2D2FF",
EDODOFF"
EEDBDBFF"
#EEE5E5FF"
                                    TRUE
 reverse =
xlab
                            =
                                       NULL
 ylab
                                       NULL
                                       Size: support Color: lift
  legend
```

×

Plot Zoom

# Scatter plot for 239 rules

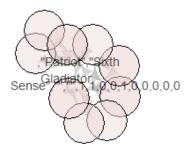


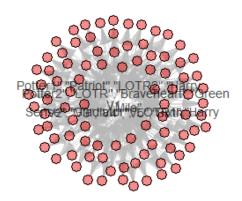
> plot(rules3,method = "graph",control = list(cex=0.90))

## Graph for 100 rules

size: support (0.091 - 0.364) color: lift (1.833 - 11)

Mile","LOTR2",1,0,1,1,50xth05e,0\$e
,"LOTR1","HarRotter1","Green





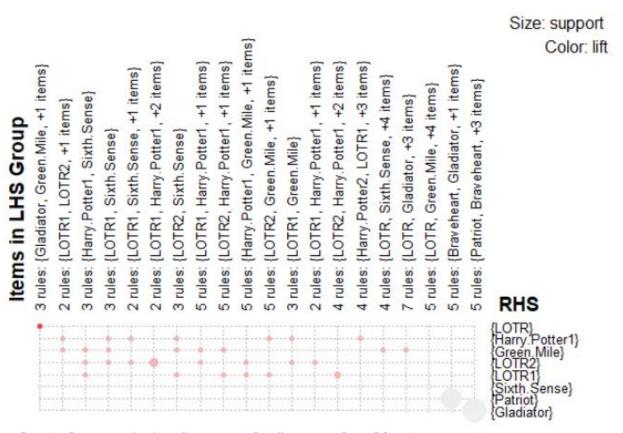
```
> mymovies1 <- read.csv(file.choose())</pre>
> names(mymovies1)
[1] "V1"
[5] "V5"
                      "v2"
                                      "v3"
                                                       "v4"
                      "Sixth.Sense"
                                      "Gladiator"
                                                       "LOTR1"
[9] "Harry.Potter1"
[13] "LOTR"
                     "Patriot"
                                      "LOTR2"
                                                       "Harry.Potter2"
                      "Braveheart"
                                      "Green.Mile"
> summary(mymovies1)
                                                   V3
                                                                  ٧4
                   Harry Potter2:1
 Gladiator
              :6
                                                    : 3
                                                                    :8
 Harry Potter1:1
                   LOTR
                                 :1
                                      Braveheart
                                                    :1
                                                         Green Mile:2
              :1
                                 :1
 LOTR1
                   LOTR1
                                      Gladiator
                                                    :1
                                 :1
 Sixth Sense
                   LOTR2
                                      Harry Potter1:1
                                      Sixth Sense :4
                   Patriot
                                 :6
     V5
            Sixth.Sense
                            Gladiator
                                             LOTR1
                                                        Harry.Potter1
      :9
           Min. :0.0
                         Min.
                                :0.00
                                         Min. :0.0
                                                        Min. :0.0
                          1st Qu.:0.25
                                                        1st Qu.:0.0
 LOTR2:1
           1st Qu.:0.0
                                         1st Qu.:0.0
                         Median :1.00
           Median :1.0
                                         Median :0.0
                                                        Median :0.0
                  :0.6
                         Mean :0.70
                                         Mean
                                                        Mean :0.2
           Mean
                                               :0.2
                          3rd Qu.:1.00
                                         3rd Qu.:0.0
                                                        3rd Qu.:0.0
           3rd Qu.:1.0
                 :1.0
                         Max.
                               :1.00
                                         Max.
                                               :1.0
                                                        Max.
                                                              :1.0
           Max.
    Patriot
                   LOTR2
                                                 LOTR
                              Harry.Potter2
                                                             Braveheart
       :0.0
                     :0.0
 Min.
               Min.
                              Min. :0.0
                                            Min. :0.0
                                                           Min. :0.0
               1st Qu.:0.0
                              1st Qu.:0.0
                                            1st Qu.:0.0
                                                           1st Qu.:0.0
 1st Qu.:0.0
 Median :1.0
               Median:0.0
                              Median:0.0
                                            Median:0.0
                                                           Median:0.0
 Mean :0.6
               Mean :0.2
                              Mean :0.1
                                            Mean :0.1
                                                           Mean :0.1
                                            3rd Qu.:0.0
 3rd Qu.:1.0
               3rd Qu.:0.0
                              3rd Qu.:0.0
                                                           3rd Qu.:0.0
```

```
Max.
         :1.0
                 Max.
                         :1.0
                                 Max.
                                         :1.0
                                                 Max.
                                                         :1.0
                                                                 Max.
                                                                         :1.0
   Green.Mile
 Min.
         :0.0
 1st Qu.:0.0
Median :0.0
 Mean
        :0.2
 3rd Qu.:0.0
        :1.0
 Max.
> attach(mymovies1)
> sd(Sixth.Sense)
[1] 0.5163978
  sd(Gladiator)
[1] 0.4830459
  sd(LOTR1)
[1] 0.421637
> sd(Harry.Potter1)
[1] 0.421637
> sd(Patriot)
[1] 0.5163978
> sd(LOTR2)
[1] 0.421637
> sd(Harry.Potter2)
[1] 0.3162278
> sd(LOTR)
[1] 0.3162278
> sd(Braveheart)
[1] 0.3162278
> sd(Green.Mile)
[1] 0.421637
> var(Sixth.Sense)
[1] 0.2666667
 var(Gladiator)
[1] 0.2333333
> var(LOTR1)
[1] 0.1777778
> var(Harry.Potter1)
[1] 0.1777778
> var(Patriot)
[1] 0.2666667
> var(LOTR2
[1] 0.1777778
> var(Harry.Potter2)
[1] 0.1
> var(LOTR)
[1] 0.1
> var(Braveheart)
[1] 0.1
> var(Green.Mile)
[1] 0.177778 > library(moments)
 skewness(Sixth.Sense)
[1] -0.4082483
 skewness(Gladiator)
[1] -0.8728716
> skewness(LOTR1)
[1] 1.5
> skewness(Harry.Potter1)
[1] 1.5
> skewness(Patriot)
[1] -0.4082483
> skewness(LOTR2)
[1] 1.5
> skewness(Harry.Potter2)
[1] 2.666667
> skewness(LOTR)
[1] 2.666667
> skewness(Braveheart)
```

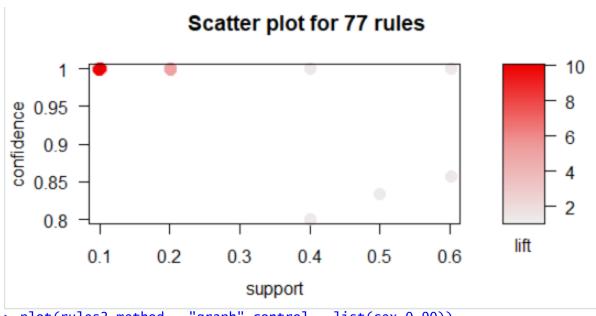
```
[1] 2.666667
> skewness(Green.Mile)
[1] 1.5
> kurtosis(Sixth.Sense)
[1] 1.166667
> kurtosis(Gladiator)
[1] 1.761905
> kurtosis(LOTR1)
[1] 3.25
 kurtosis(Harry.Potter1)
[1] 3.25
  kurtosis(Patriot)
[1] 1.166667
 kurtosis(LOTR2)
[1] 3.25
> kurtosis(Harry.Potter2)
[1] 8.111111
> kurtosis(LOTR)
[1] 8.111111
> kurtosis(Braveheart)
[1] 8.111111
> kurtosis(Green.Mile)
[1] 3.25
> rules2 <- apriori(as.matrix(mymovies1[,6:15],parameter=list(support=0.2, co</pre>
nfidence = 0.5,minlen=5)))
Apriori
Parameter specification:
 confidence minval smax arem aval originalSupport maxtime support minlen
                        1 none FALSE
         0.8 0.1
                                                     TRUE
                                                                 - 5
                                                                        0.1
 maxlen target ext
     10 rules FALSE
Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE FALSE TRUE
Absolute minimum support count: 1
set item appearances ...[0 item(s)] done [0.00s]. set transactions ...[10 item(s), 10 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 5 done [0.00s]. writing ... [77 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> rules2
set of 77 rules
> rules_conf <- sort (rules2, by="confidence", decreasing=T)</pre>
> inspect(head(rules_conf))
    1hs
                                             support confidence lift
                          rhs
                                                                             count
[1] {Harry.Potter2} => {Harry.Potter1} 0.1
                                                                   5.000000 1
                                                      1
    {Braveheart}
                     => {Patriot}
                                             0.1
                                                      1
                                                                   1.666667 1
[3] {Braveheart}
                                             0.1
                                                      1
                                                                   1.428571 1
                       => {Gladiator}
                      => {Green.Mile}
=> {Gladiator}
[4] {LOTR}
[5] {LOTR}
                                             0.1
                                                                   5.000000 1
                                                      1
                                             0.1
                                                                   1.428571
                       => {Sixth.Sense}
[6] {LOTR}
                                             0.1
                                                                   1.666667 1
> rules_lift <- sort(rules2, by="lift", decreasing = T)
> inspect(head(rules_lift))
                                                                   support confidence
                                                rhs
    {Gladiator,Green.Mile}
                                             => {LOTR}
                                                                   0.1
                                                                            1
    {Sixth.Sense,Gladiator,Green.Mile} => {LOTR}
                                                                   0.1
                                                                            1
[3]
                                                                            1
    {Harry.Potter2}
                                             => {Harry.Potter1} 0.1
[4]
    {LOTR}
                                             => {Green.Mile}
                                                                   0.1
                                                                            1
    {LOTR1}
                                                                            1
 <sup>-</sup>51
                                             => {LOTR2}
                                                                   0.2
                                                                   0.2
                                                                            1
[6] {LOTR2}
                                             => {LOTR1}
```

```
lift count
10   1
10   1
5   1
5   1
5   2
1ot(rules2, method = "grouped",control = list(cex=0.90))
[1]
[2]
[3]
[4]
[5]
[6]
> p]
```

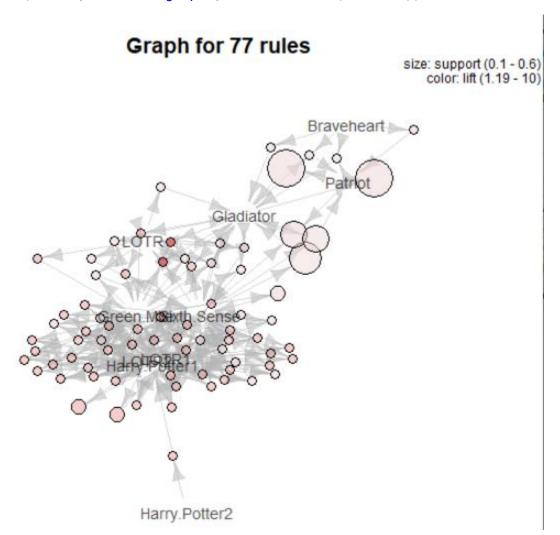
# **Grouped Matrix for 77 Rules**



> plot(rules2,method = "scatterplot",control = list(cex=0.90))



> plot(rules2,method = "graph",control = list(cex=0.90))

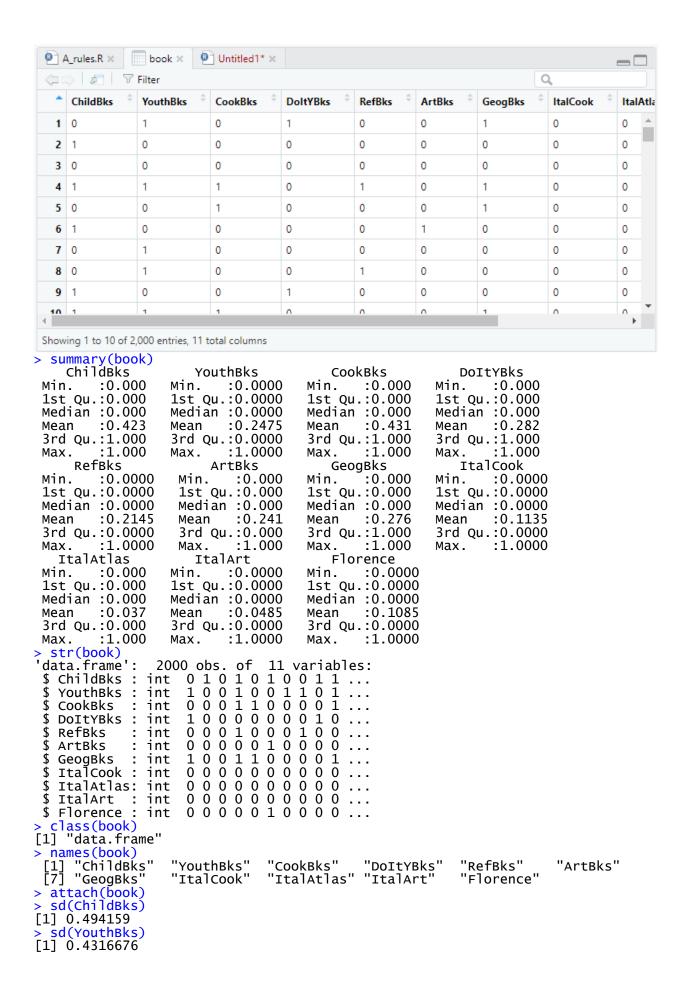


## **Rcode:**

#Different set of rule values for book Dataset using apriori algorithm. book <- read.csv(file.choose())</pre> View(book) summary(book) str(book) class(book) names(book) attach(book) sd(ChildBks) sd(YouthBks) sd(CookBks) sd(DoItYBks) sd(RefBks) sd(ArtBks) sd(GeogBks) sd(ItalCook) sd(ItalAtlas) sd(ItalArt) sd(Florence) var(book) skewness(ChildBks) skewness(YouthBks) skewness(CookBks) skewness(DoItYBks) skewness(RefBks)

```
skewness(ArtBks)
skewness(GeogBks)
skewness(ItalCook)
skewness(ItalAtlas)
skewness(ItalArt)
skewness(Florence)
kurtosis(ChildBks)
kurtosis(YouthBks)
kurtosis(CookBks)
kurtosis(DoItYBks)
kurtosis(RefBks)
kurtosis(ArtBks)
kurtosis(GeogBks)
kurtosis(ItalCook)
kurtosis(ItalAtlas)
kurtosis(ItalArt)
kurtosis(Florence)
rules4 <- apriori(as.matrix(book,parameter=list(support=0.02, confidence =
0.5, \minlen=5)))
rules4
rules_confidence <- sort (rules4, by="confidence", decreasing=T)</pre>
inspect(head(rules_confidence))
rules_lift <- sort(rules4,by="lift",decreasing = T)</pre>
inspect(head(rules_lift))
plot(rules4, method = "grouped",control = list(cex=0.90))
plot(rules4,method = "scatterplot",control = list(cex=0.90))
```

```
plot(rules4,method = "graph",control = list(cex=0.90))
rules5 <- apriori(as.matrix(book,parameter=list(support=0.05, confidence =
0.7, minlen=6)))
rules5
rules_confidence <- sort (rules5, by="confidence", decreasing=T)
inspect(head(rules_confidence))
rules lift <- sort(rules5,by="lift",decreasing = T)
inspect(head(rules_lift))
plot(rules5, method = "grouped",control = list(cex=0.90))
plot(rules5,method = "scatterplot",control = list(cex=0.90))
plot(rules5,method = "graph",control = list(cex=0.90))
#remove redundant rules
subsetRules <- which(colSums(is.subset(rules4, rules4)) > 1)
length(subsetRules)
rules4 <- rules4[-subsetRules]
rules4
subsetRules <- which(colSums(is.subset(rules5, rules5)) > 1)
length(subsetRules)
rules5 <- rules5[-subsetRules]</pre>
rules5
Console:
> #Different set of rule values for book Dataset using apriori algorithm.
> book <- read.csv(file.choose())
> View(book)
```



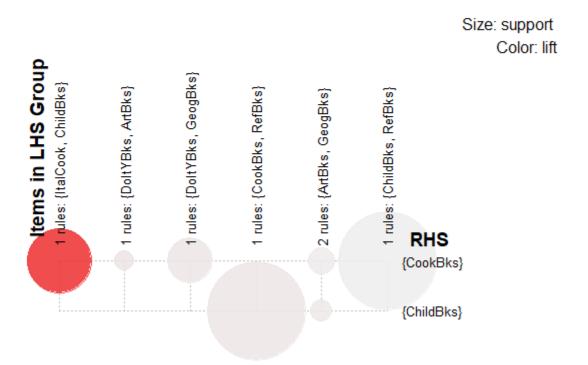
```
[1] 0.49534
 sd(DoItYBks)
[1] 0.4500859
> sd(RefBks)
[1] 0.4105777
 sd(ArtBks)
[1] 0.4277973
> sd(GeogBks)
[1] 0.4471286
 sd(ItalCook)
[1] 0.3172823
  sd(ItalAtlas)
[1] 0.188809
 sd(ItalArt)
[1] 0.214874
> sd(Florence)
[1] 0.3110886
> var(book)
             ChildBks
                           YouthBks
                                         CookBks
                                                      DoItYBks
                                                                    RefBks
                       0.060337669 0.0737238619 0.0647463732 0.060796898
ChildBks
          0.244193097
                       0.186336918 0.0553551776 0.0457278639 0.043432966
YouthBks
          0.060337669
          0.073723862
                       0.055355178 0.2453616808 0.0659909955 0.060080540
CookBks
DoItYBks
          0.064746373
                       0.045727864 0.0659909955 0.2025772886 0.045033517
RefBks
          0.060796898
                       0.043432966 0.0600805403 0.0450335168 0.168574037
          0.060587294
                       0.041373187 0.0631605803 0.0555657829 0.037824412
ArtBks
          0.078291146
                       0.052216108 0.0735807904 0.0546953477 0.051323662
GeogBks
          0.037008004
ItaĺCook
                       0.030924212 0.0646138069 0.0265062531 0.022165333
                       0.008346673  0.0125592796  0.0085702851  0.029078039
ItalAtlas 0.012855428
          0.015492246
                       0.011001751 0.0201065533 0.0163311656 0.009601551
ItalArt
                      -0.001354427 0.0007368684 0.0009034517 0.007730615
Florence
          0.002605803
               ArtBks
                          GeogBks
                                     ItalCook
                                                ItalAtlas
                                                               ItalArt
ChildBks
          0.060587294 0.07829115 0.037008004 0.012855428 0.015492246
          0.041373187 0.05221611 0.030924212 0.008346673 0.011001751
YouthBks
          0.063160580 0.07358079 0.064613807 0.012559280 0.020106553
CookBks
          0.055565783 0.05469535 0.026506253 0.008570285 0.016331166
DoItYBks
          0.037824412 0.05132366 0.022165333 0.029078039 0.009601551
RefBks
          0.183010505 0.06101451 0.029161081 0.009087544 0.036829915
ArtBks
GeogBks
          0.061014507 0.19992396 0.032690345 0.010293147 0.016122061
          0.029161081 0.03269035 0.100668084 0.018809905 0.032011256
ItalCook
Italatlas 0.009087544 0.01029315 0.018809905 0.035648824 0.014712856
ItalArt
          0.036829915 0.01612206 0.032011256 0.014712856 0.046170835
Florence
          0.022362681 0.01256028 0.005187844 0.002486743 0.007241371
               Florence
           0.0026058029
ChildBks
YouthBks
          -0.0013544272
CookBks
           0.0007368684
DoItYBks
           0.0009034517
RefBks
           0.0077306153
ArtBks
           0.0223626813
           0.0125602801
GeogBks
           0.0051878439
ItalCook
ItalAtlas
           0.0024867434
           0.0072413707
ItalArt
           0.0967761381
Florence
  skewness(ChildBks)
[1] 0.3117185
> skewness(YouthBks)
[1] 1.170174
> skewness(CookBks)
[1] 0.2786662
 skewness(DoItYBks)
[1] 0.9689463
 skewness(RefBks)
[1] 1.391071
> skewness(ArtBks)
[1] 1.211157
```

sd(CookBks)

```
> skewness(GeogBks)
[1] 1.0022
> skewness(ItalCook)
[1] 2.436925
> skewness(ItalAtlas)
[1] 4.905655
> skewness(ItalArt)
[1] 4.203514
> skewness(Florence)
[1] 2.517597
> kurtosis(ChildBks)
[1] 1.097168
> kurtosis(YouthBks)
[1] 2.369308
> kurtosis(CookBks)
[1] 1.077655
> kurtosis(DoItYBks)
[1] 1.938857
> kurtosis(RefBks)
[1] 2.935079
> kurtosis(ArtBks)
[1] 2.466901
> kurtosis(GeogBks)
[1] 2.004404
> kurtosis(ItalCook)
[1] 6.938604
> kurtosis(ItalAtlas)
[1] 25.06545
 kurtosis(ItalArt)
[1] 18.66953
> kurtosis(Florence)
[1] 7.338295
> rules4 <- apriori(as.matrix(book,parameter=list(support=0.02, confidence =</pre>
0.5,minlen=5)))
Apriori
Parameter specification:
 confidence minval smax arem aval original Support maxtime support minlen
                         1 none FALSE
                                                     TRUE
                                                                        0.1
 maxlen target
                   ext
     10 rules FALSE
Algorithmic control:
 filter tree heap memopt load sort verbose
     0.1 TRUE TRUE FALSE TRUE
Absolute minimum support count: 200
set item appearances ...[0 item(s)] done [0.00s]. set transactions ...[11 item(s), 2000 transaction(s)] done [0.00s]. sorting and recoding items ... [9 item(s)] done [0.00s].
creating transaction tree ... done [0.00s]. checking subsets of size 1 2 3 4 done [0.00s].
writing ... [7 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> rules4
set of 7 rules
> rules_confidence <- sort (rules4, by="confidence", decreasing=T)</p>
> inspect(head(rules_confidence))
                                          support confidence lift
    1hs
                              rhs
                                                                          count
                                          0.1135
                                                   1.0000000
                                                                2.320186 227
    {ItalCook}
                              {CookBks}
    {DoItYBks,ArtBks} =>
                              {CookBks}
                                          0.1015
                                                   0.8218623
                                                                1.906873 203
                                          0.1085
                                                                1.899926 217
    {DoItYBks,GeogBks} => {CookBks}
                                                   0.8188679
 [4] {ArtBks,GeogBks} => {CookBks}
                                          0.1035
                                                   0.8117647
                                                                1.883445 207
    {ChildBks,RefBks} => {CookBks}
                                          0.1225
                                                   0.8085809
                                                                1.876058 245
                          => {ChildBks} 0.1225
                                                   0.8032787
                                                                1.899004 245
[6] {CookBks,RefBks}
```

```
> rules_lift <- sort(rules4,by="lift",decreasing = T)</pre>
  inspect(head(rules_lift))
    1hs
                                           support confidence lift
                                           0.1135
                                                     1.0000000
                                                                  2.320186 227
    {ItalCook}
                               {CookBks}
    {DoItYBks, ArtBks} =>
                                                                  1.906873 203
                               {CookBks}
                                           0.1015
                                                     0.8218623
[3]
    {DoItYBks,GeogBks} =>
                              {CookBks}
                                           0.1085
                                                     0.8188679
                                                                  1.899926 217
                                                                  1.899004 245
                              {ChildBks} 0.1225
                                                     0.8032787
    {CookBks,RefBks}
                           =>
                           => {ChildBks} 0.1020
                                                     0.8000000
                                                                  1.891253 204
    {ArtBks,GeogBks}
[6] {ArtBks,GeogBks} => {CookBks} 0.1035 0.8117647 1.8
> plot(rules4, method = "grouped",control = list(cex=0.90))
                                                                  1.883445 207
```

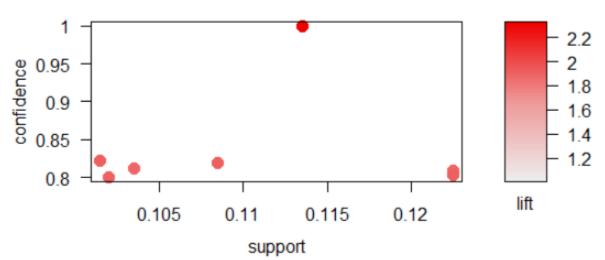
## **Grouped Matrix for 7 Rules**



```
Available control parameters (with default values):
                  Grouped Matrix for 7 Rules
main
                   20
rhs_max =
                   10
lhs_items
                               aggr.fun
                  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF",
                                                                                     "#EE0909FF."
col = "#EE0F0FFF"
                                                                                                         "#EE0C0CFF"
                                                                                                        #EE1E1EFF",
"#EE2222FF",
                    "#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF",
                                                                                                        "#EE3131FF",
                                         "#EE3A3AFF",
"#EE3434FF",
                     "#EE3737FF",
                                                              "#EE3D3DFF",
                                                                                   "#EE4040FF",
                                                                                                         "#EE4444FF".
                                        "#EE3A3AFF", "
"#EE4D4DFF", "
"#EE5F5FFF", "
"#EE7272FF", "
", "#EE8484FF",
                                                              #EE5050FF",
                    "#EE4A4AFF",
"#EE4747FF",
                                                                                   "#EE5353FF",
                                                                                                        "#EE5656FF",
                                                              #EE5U5UFF",
"#EE6262FF",
"#EE7575FF",
                    "#EE5C5CFF",
"#EE5959FF",
                                                                                   "#EE6666FF",
                                                                                                         "#EE6969FF",
"#EE5959FF", "#EE5C5CFF", 
"#EE6C6CFF", "#EE6F6FFF", 
, "#EE7E7EFF", "#EE8181FF
                                                                                      "#EE7878FF"
                                                                                                          "#EE7B7BFF"
                                                                                       "#EE8B8BFF",
"#EE9B9BFF"
                          #EE8181FF".
                                                                   #EE8888FF",
                                                                                                            "#EE8E8EFF
                                               "#EE9797FF"
     "#EE9191FF"
                          "#EE9494FF"
                                                                    "#EE9999FF"
                                                                                                              "#EE9D9DF
                                                #EE9/9/FF , #E
"#EEA2A2FF", "#
"#EEACACFF", '
', "#EEB7B7FF",
"", "#EEC1C1FF",
                                                                     "#EEA4A4FF"
                           "#EEAOAOFF"
                                                                                          "#EEA5A5FF"
       "#EE9F9FFF"
FF"
        "#EEA9A9FF"
                             "#EEABABFF",
"#EEB5B5FF"
                                                                       "#EEAEAEFF"
                                                                                            "#EEBOBOFF"
                                                                        "#EEB8B8FF"
          "#EEB3B3FF"
                                                                                              "#EEBABAFF"
6C6FF", "#EEC8C8FF", "#EEC9C9FF", "#EEC1C1FF", "#EEC3C3FF", "#EEC4C4FF", "#EEC9C9FF", "#EEC9C9FF", "#EEC9C9FF", "#EEC9C9FF", "#EEC9C9FF", "#EEC9C9FF", "#EEC9C9FF", "#EEC9C9FF", "#EED9D9FF", "#EED9D9FF", "#EED9D9FF", "#EED9D9FF", "#EED9D9FF", "#EED9D9FF", "#EED9D9FF", "#EEE3E3FF", "#EEE5E5FF", "#EEE7E7FF", "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF") reverse = TRUE
                                "#EEBFBFFF",
                                                                          "#EEC3C3FF",
           "#EEBDBDFF"
BCEE,
```

```
xlab
                     NULL
                =
ylab
                     NULL
                     Size: support Color: lift
legend
spacing = -1
panel.function = function (row, size, shading, spacing) { size[size == 0] <- NA shading[is.na(shading)] <- 1 grid.circle(x = c(1:length(size )), y = row, r = size/2 * (1 - spacing), default.units = "native", gp = gpar(fill = shading, col = shading, alpha = 0.9)) } gp_main = list(cex = 1.2, fontface = "bold", font = c(bold = 2))
                     = list(cex = 0.8)
list(cex = 1.2, fontface = "bold", font = c(bold = 2))
= list(col = "gray", lty = 3)
gp_labels
gp_labs = gp_lines
newpage = TRUE
max shading
                     default
engine = T
verbose = FALSE
> plot(rules4,method = "scatterplot",control = list(cex=0.90))
```

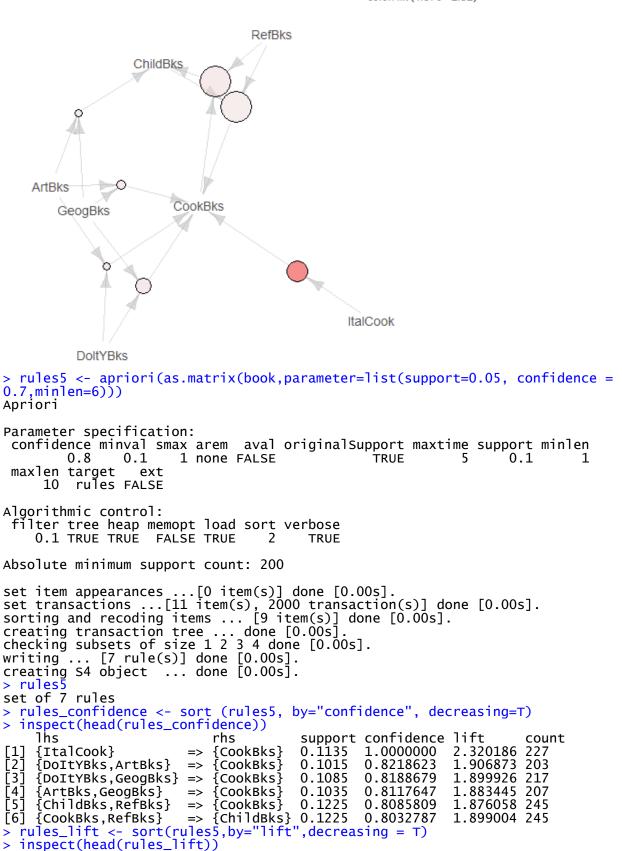
# Scatter plot for 7 rules



> plot(rules4,method = "graph",control = list(cex=0.90))

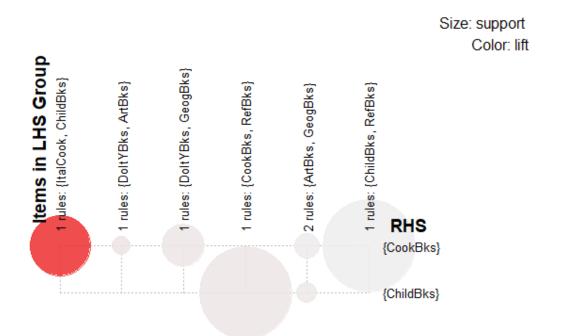
## Graph for 7 rules

size: support (0.102 - 0.122) color: lift (1.876 - 2.32)



```
support confidence lift
     1hs
                               rhs
                                                                   2.320186 227
                               {CookBks}
                                            0.1135
                                                      1.0000000
    {ItalCook}
                           =>
     {DoItYBks,ArtBks} =>
                                            0.1015
                                                      0.8218623
                                                                   1.906873 203
                               {CookBks}
                                            0.1085
                                                      0.8188679
    {DoItYBks,GeogBks} =>
                               {CookBks}
                                                                   1.899926 217
                               {ChildBks} 0.1225
                                                      0.8032787
                                                                   1.899004 245
    {CookBks, RefBks}
                           =>
    {ArtBks,GeogBks}
                           =>
                               {ChildBks} 0.1020
                                                      0.8000000
                                                                   1.891253 204
[6] {ArtBks,GeogBks} => {CookBks} 0.1035 0.8117647 1.8
> plot(rules5, method = "grouped",control = list(cex=0.90))
                                                                  1.883445 207
```

## **Grouped Matrix for 7 Rules**



```
Available control parameters (with default values):
                        Grouped Matrix for 7 Rules
                         20
rhs_max =
lhs_items
aggr.fun = function (x, ...) UseMethod("mean")

col = c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF",

"#EE0F0FFF", "#EE1212FF", "#EE1515FF", "#EE1818FF", "#EE1B1BFF",

"#EE2222FF", "#EE2525FF", "#EE2828FF", "#EE282BFF", "#EE2E2EFF",

"#EE3434FF", "#EE3737FF", "#EE3A3AFF", "#EE3D3DFF", "#EE4040FF",

"#EE4747FF", "#EE444AFF", "#EE4D4DFF", "#EE5050FF", "#EE5353FF",

"#EE5959FF", "#EE5C5CFF", "#EE5F5FFF", "#EE6266FF", "#EE6666FF",

"#EE6C6CFF", "#EE6F6FFF", "#EE7272FF", "#EE7575FF", "#EE7878FF",

"#EE7E7FFF", "#EE8181FF", "#EE8484FF", "#EE8888FF", "#EE888BFF"

" "#FF9191FF" "#FF9494FF" "#FF979FF" "#FF9999FF" "#FF989RFF
                                                                                                                                            "#EE0C0CFF"
                                                                                                                                          "#EE1E1EFF",
                                                                                                                                          "#EE5656FF".
                                                                                                                                          "#EE6969FF".
                                                                                                                                             "#EE7B7BFF"
                                                                                                                   "#EE8B8BFF"
                                                                                                                                               "#EE8E8EFF
                                         "#EE9494FF"
      "#EE9191FF"
                                                                                                                                                 "#EE9D9DF
                                    "#EEAOAOFF",
         "#EE9F9FFF"
                                                                                                                                                   "#EEA7A7
                                      "#EEABABFF",
"#EEB5B5FF"
          "#EEA9A9FF"
             "#EEB3B3FF"
                '#EEBDBDFF"
6C6FF"
                 "#EEC8C8FF"
                                                                                                                                  "#EECFCFFF",
                                                 #EEC9C9FF", "#EECBCBFF", "#EECDCDFF", "#EECFCFFF", "#
"#EED4D4FF", "#EED5D5FF", "#EED7D7FF", "#EED9D9FF", "
"#EEDEDEFF", "#EEE0E0FF", "#EEE1E1FF", "#EEE3E3FF",
, "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF")
                                                 "#EED4D4FF"
                     "#EED2D2FF",
"#EEDCDCFF",
EDODOFÉ"
EEDBDBFF"
                         "#EEE7E7FF",
#EEE5E5FF".
                        TRUE
reverse =
xlab
                        NULL
ylab
                        NULL
legend
                        Size: support Color: lift
spacing = -1
```

```
gp_1abs =
gp_lines
newpage =
        TRUE
max.shading
              NΑ
engine = defau
verbose = FALSE
         default
> plot(rules5,method = "scatterplot",control = list(cex=0.90))
               Scatter plot for 7 rules
                                             2.2
                                             2
  0.95
confidence
                                             1.8
   0.9
                                             1.6
                                             1.4
  0.85
                                              1.2
   8.0
                                         lift
          0.105
                  0.11
                        0.115
                               0.12
                    support
> plot(rules5,method = "graph",control = list(cex=0.90))
```

# Graph for 7 rules

size: support (0.102 - 0.122) color: lift (1.876 - 2.32)

