Data Visualization Lab(L13+L14)

Name: Alpanshu Kataria

Reg. No.: 18BCE1267

Faculty: Dr.Parvathi.R.

Lab 4

Task:

Take a dataset and apply association rule learning.

Solution:

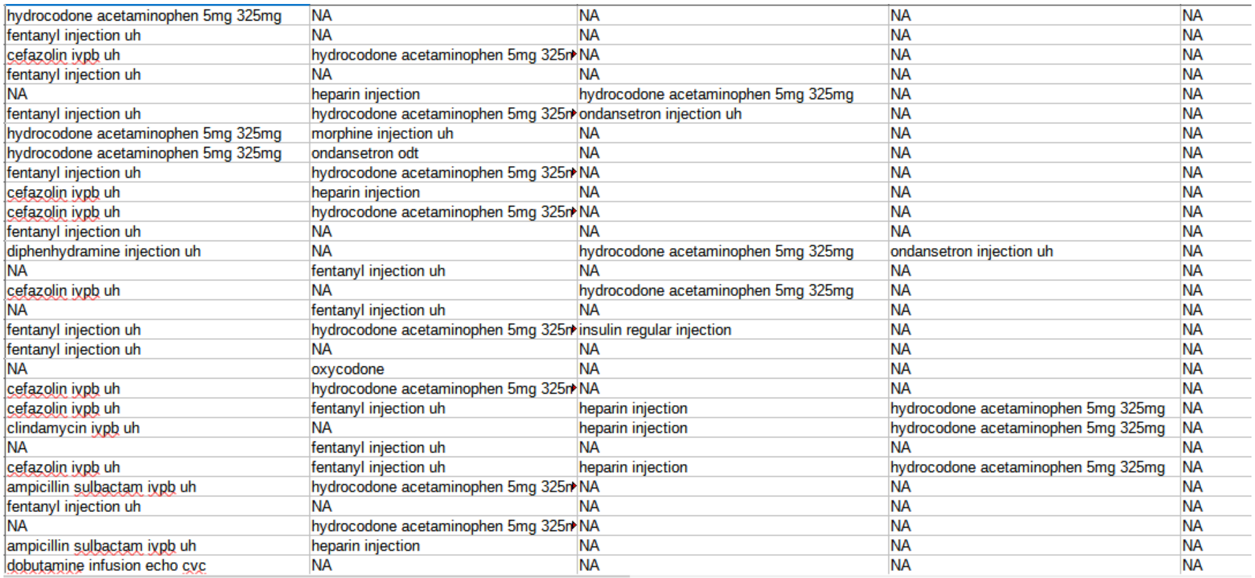
Dataset:

To demonstrate the association rule learning in a real biomedical case-study, I've used a transactional healthcare data representing a subset of the Head and Neck Cancer Medication data,.It consists of inpatient medications for head and neck cancer patients.

Link:

http://www.socr.umich.edu/people/dinov/courses/DSPA_notes/11_Apriory_AssocRuleLearning.html#72_Step_2_-_exploring_and_preparing_the_data

Dataset sample:



hydrocodone acetaminophen 5mg 325mg	NA	NA	NA	NA
fentanyl injection uh	NA	NA	NA	NA
cefazolin ivpb uh	hydrocodone acetaminophen 5mg 325m	NA	NA	NA
fentanyl injection uh	NA	NA	NA	NA
NA	heparin injection	hydrocodone acetaminophen 5mg 325mg	NA	NA
fentanyl injection uh	hydrocodone acetaminophen 5mg 325m	ondansetron injection uh	NA	NA
hydrocodone acetaminophen 5mg 325mg	morphine injection uh	NA	NA	NA
hydrocodone acetaminophen 5mg 325mg	ondansetron odt	NA	NA	NA
fentanyl injection uh	hydrocodone acetaminophen 5mg 325m	NA	NA	NA
cefazolin ivpb uh	heparin injection	NA	NA	NA
cefazolin ivpb uh	hydrocodone acetaminophen 5mg 325m	NA	NA	NA
fentanyl injection uh	NA	NA	NA	NA
diphenhydramine injection uh	NA	hydrocodone acetaminophen 5mg 325mg	ondansetron injection uh	NA
NA	fentanyl injection uh	NA	NA	NA
cefazolin ivpb uh	NA	hydrocodone acetaminophen 5mg 325mg	NA	NA
NA	fentanyl injection uh	NA	NA	NA
fentanyl injection uh	hydrocodone acetaminophen 5mg 325m	insulin regular injection	NA	NA
fentanyl injection uh	NA	NA	NA	NA
NA	oxycodone	NA	NA	NA
cefazolin ivpb uh	hydrocodone acetaminophen 5mg 325m	NA	NA	NA
cefazolin ivpb uh	fentanyl injection uh	heparin injection	hydrocodone acetaminophen 5mg 325mg	NA
clindamycin ivpb uh	NA	heparin injection	hydrocodone acetaminophen 5mg 325mg	NA
NA	fentanyl injection uh	NA	NA	NA
cefazolin ivpb uh	fentanyl injection uh	heparin injection	hydrocodone acetaminophen 5mg 325mg	NA
ampicillin sulbactam ivpb uh	hydrocodone acetaminophen 5mg 325m	NA	NA	NA
fentanyl injection uh	NA	NA	NA	NA
NA	hydrocodone acetaminophen 5mg 325m	NA	NA	NA
ampicillin sulbactam ivpb uh	heparin injection	NA	NA	NA
dobutamine infusion echo cvc	NA	NA	NA	NA

Key Points:

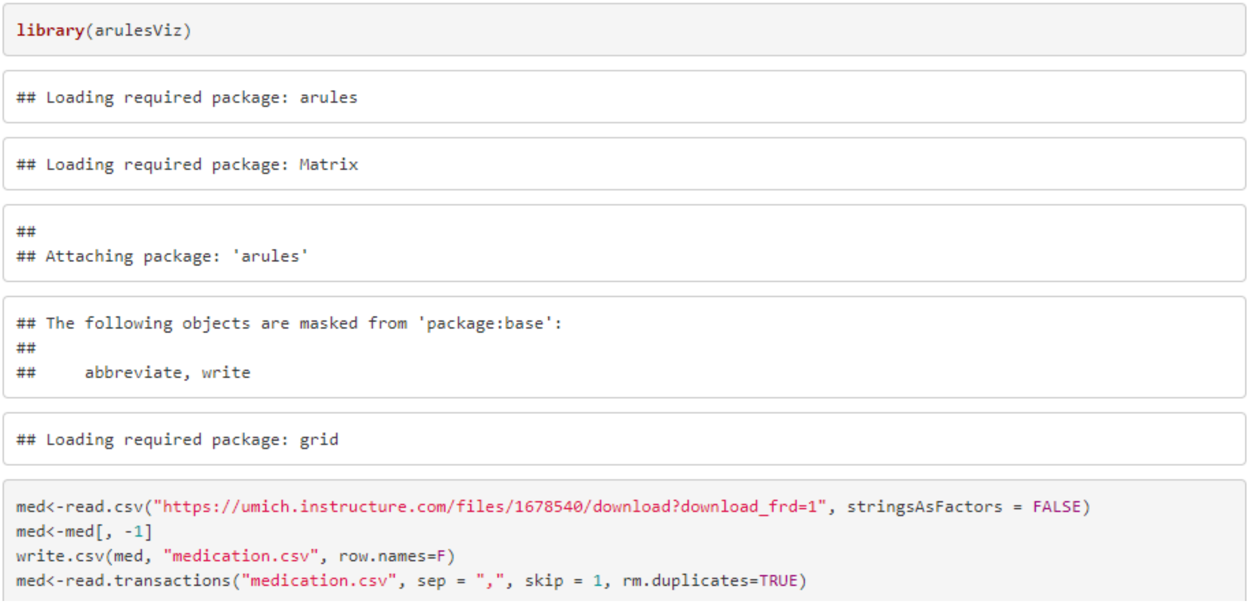
We apply an iterative approach or level-wise search where k-frequent itemsets are used to find itemsets. To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called Apriori property which helps by reducing the search space. Apriori Property – All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Parameters

1. **Support:** This measure gives an idea of how frequent an itemset is in all the transactions.

2. **Confidence:** This measure defines the likeliness of occurrence of consequent on the cart given that the cart already has the antecedents.

3. **Lift:** Lift controls for the support (frequency) of consequent while calculating the conditional probability of occurrence of {Y} given {X}.

Code:



```
library(arulesViz)
```

```
## Loading required package: arules
```

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

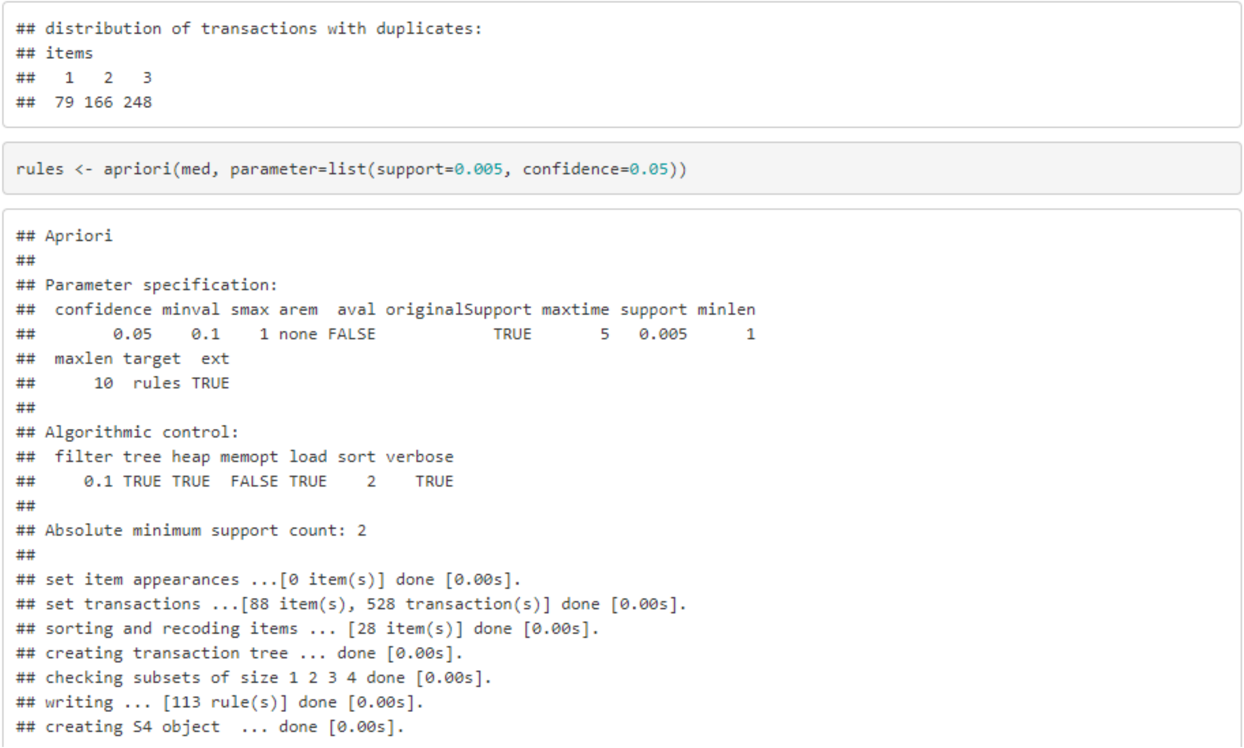
```
## 
## Attaching package: 'arules'
```

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

```
## Loading required package: grid
```

```
med<-read.csv("https://umich.instructure.com/files/1678540/download?download_frd=1", stringsAsFactors = FALSE)
med<-med[, -1]
write.csv(med, "medication.csv", row.names=F)
med<-read.transactions("medication.csv", sep = ",", skip = 1, rm.duplicates=TRUE)
```

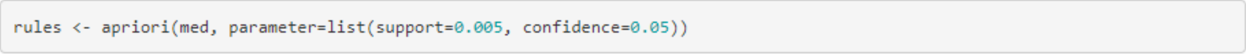
We've loaded our dataset now we'll apply apriori algorithm with support as 0.005 and confidence as 0.05 the data



```
## distribution of transactions with duplicates:
## items
##   1   2   3
##  79 166 248
```

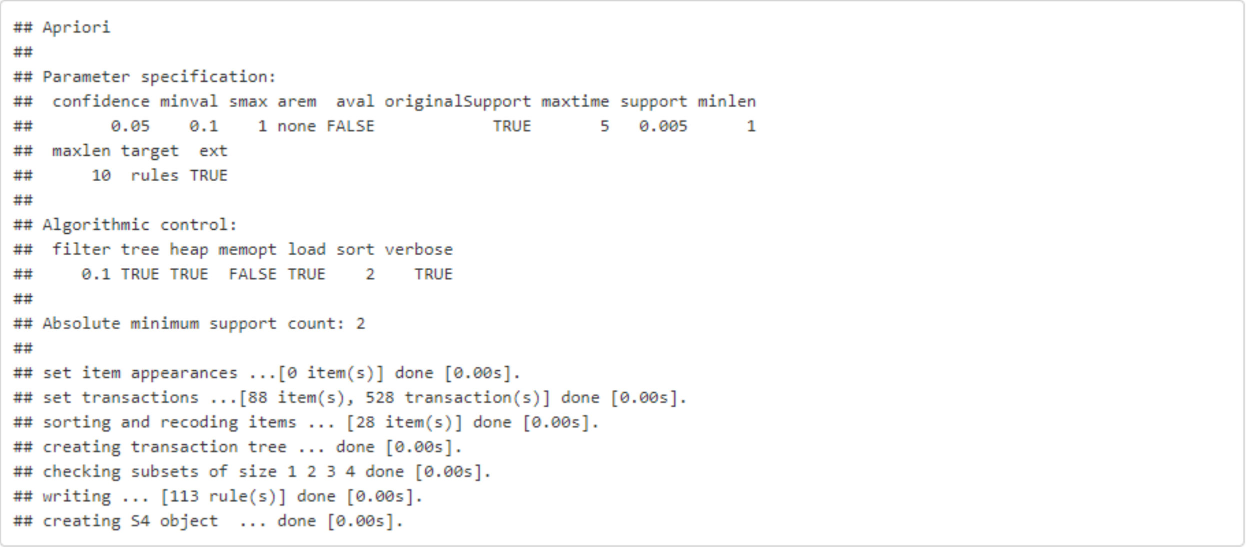
```
rules <- apriori(med, parameter=list(support=0.005, confidence=0.05))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5   0.005      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[88 item(s), 528 transaction(s)] done [0.00s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [113 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

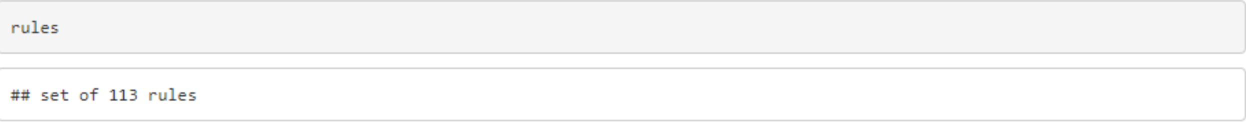


```
rules <- apriori(med, parameter=list(support=0.005, confidence=0.05))
```

The values of support and confidence are user defined and based on these values rules are generated.



```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5   0.005      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[88 item(s), 528 transaction(s)] done [0.00s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [113 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

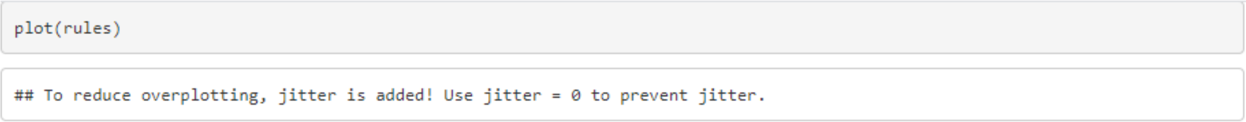


```
rules
```

```
## set of 113 rules
```

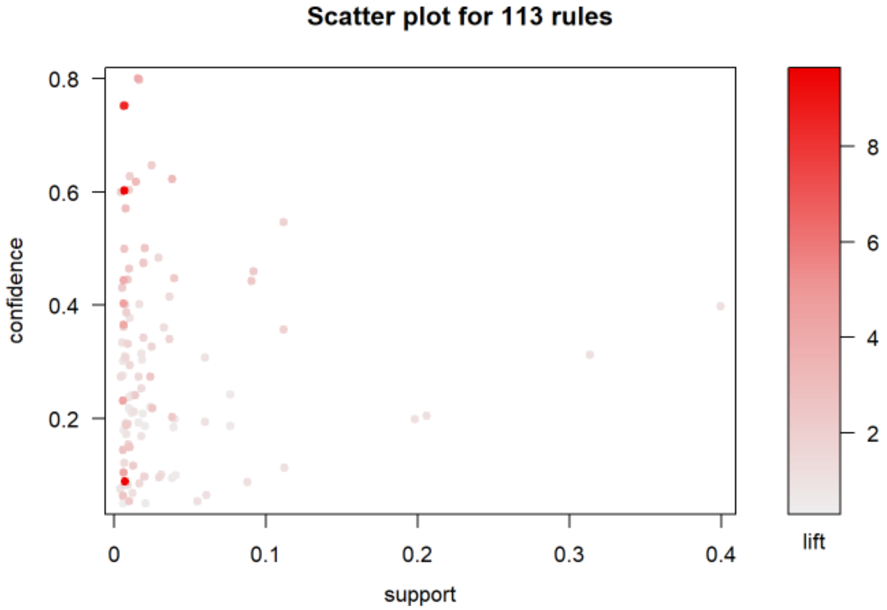
In our case 113 rules are generated.

Now we'll plot them.



```
plot(rules)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```

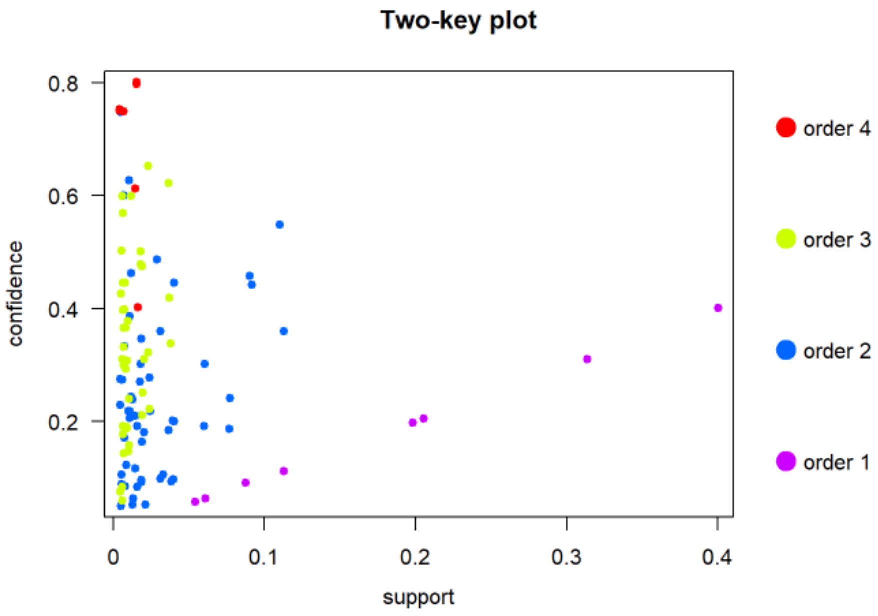


Above scatter plot shows the 113 rules generated by the apriori algorithm for our dataset with support=0.005, confidence=0.05

Now we’ll group the rules based on order.

```
plot(rules, shading="order", control=list(main = "Two-key plot",col=rainbow(5)))
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```

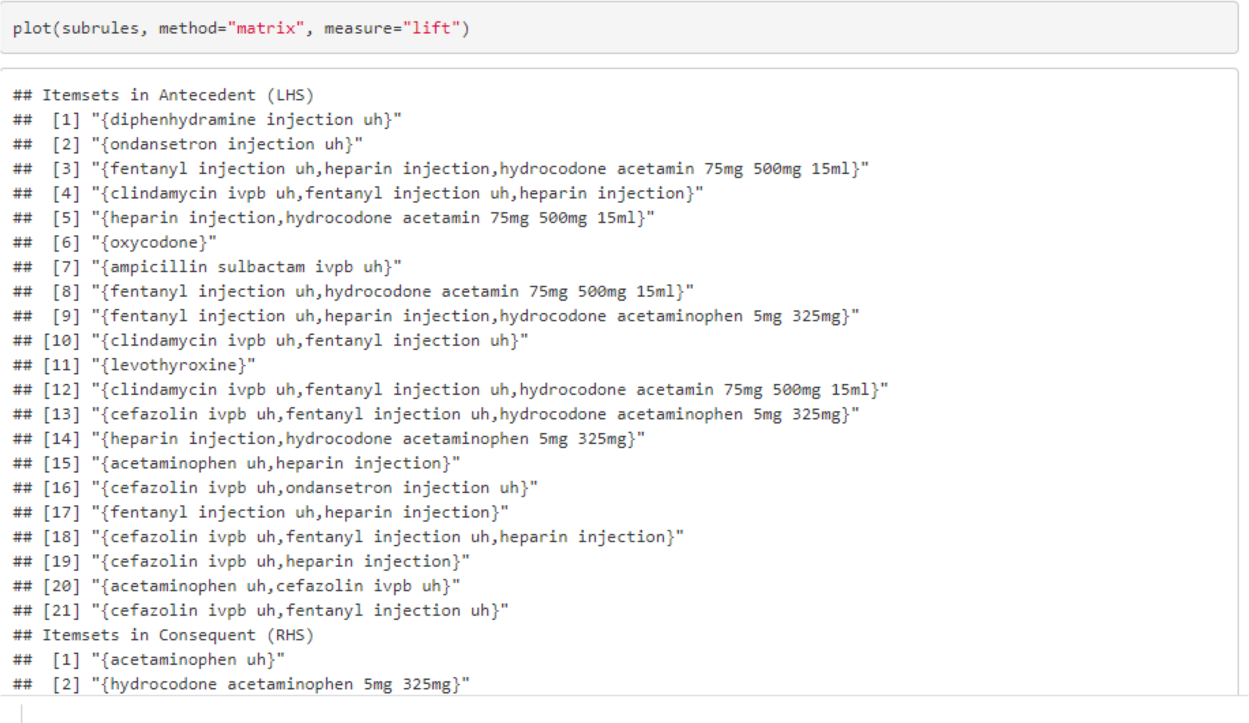


This visualization method draws a two dimensional scatterplot with different measures of interestingness (support and confidence) on the axes and a third measure (parameter “shading”) is represented by the points color. There is a special value for shading called “order”. With this value the color of the points represents the length (order) of the rule. This is used for two-key plots

2D matrix with shading.
The following techniques work better with fewer rules for better visualization

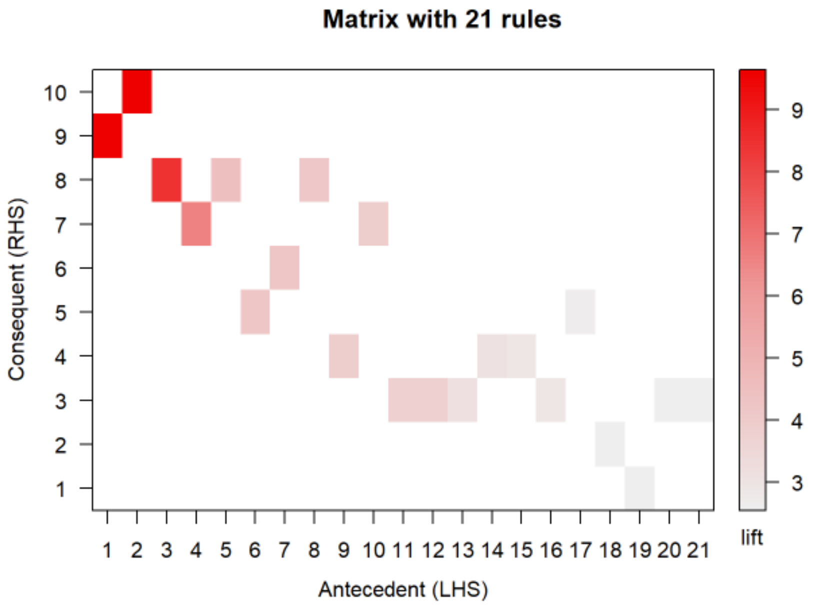
```
subrules <- subset(rules, lift>2.5)
subrules
```

```
## set of 21 rules
```



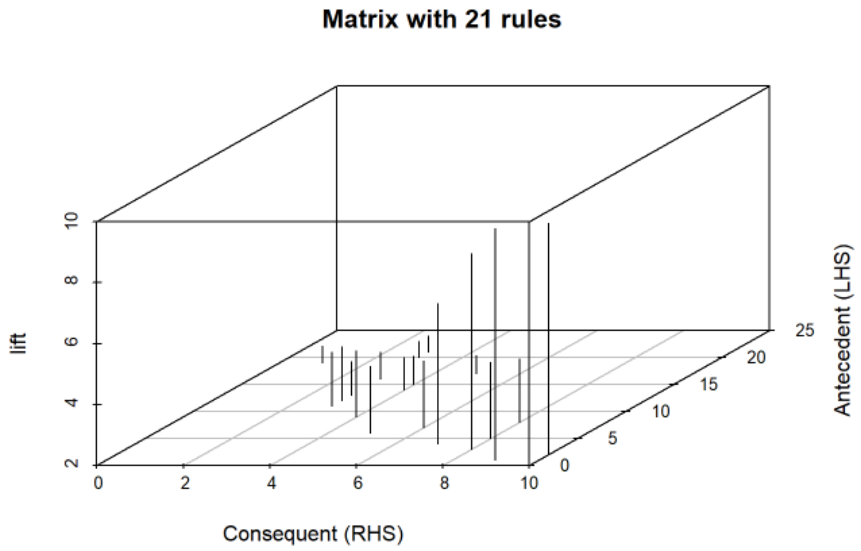
```
plot(subrules, method="matrix", measure="lift")
```

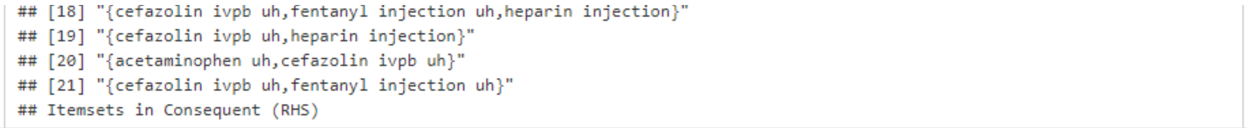
```
## Itemsets in Antecedent (LHS)
##  [1] "{diphenhydramine injection uh}"
##  [2] "{ondansetron injection uh}"
##  [3] "{fentanyl injection uh,heparin injection,hydrocodone acetamin 75mg 500mg 15ml}"
##  [4] "{clindamycin ivpb uh,fentanyl injection uh,heparin injection}"
##  [5] "{heparin injection,hydrocodone acetamin 75mg 500mg 15ml}"
##  [6] "{oxycodone}"
##  [7] "{ampicillin sulbactam ivpb uh}"
##  [8] "{fentanyl injection uh,hydrocodone acetamin 75mg 500mg 15ml}"
##  [9] "{fentanyl injection uh,heparin injection,hydrocodone acetaminophen 5mg 325mg}"
## [10] "{clindamycin ivpb uh,fentanyl injection uh}"
## [11] "{levothyroxine}"
## [12] "{clindamycin ivpb uh,fentanyl injection uh,hydrocodone acetamin 75mg 500mg 15ml}"
## [13] "{cefazolin ivpb uh,fentanyl injection uh,hydrocodone acetaminophen 5mg 325mg}"
## [14] "{heparin injection,hydrocodone acetaminophen 5mg 325mg}"
## [15] "{acetaminophen uh,heparin injection}"
## [16] "{cefazolin ivpb uh,ondansetron injection uh}"
## [17] "{fentanyl injection uh,heparin injection}"
## [18] "{cefazolin ivpb uh,fentanyl injection uh,heparin injection}"
## [19] "{cefazolin ivpb uh,heparin injection}"
## [20] "{acetaminophen uh,cefazolin ivpb uh}"
## [21] "{cefazolin ivpb uh,fentanyl injection uh}"
## Itemsets in Consequent (RHS)
##  [1] "{acetaminophen uh}"
##  [2] "{hydrocodone acetaminophen 5mg 325mg}"
```



It arranges the association rules as a matrix with the itemsets in the antecedents on one axis and the itemsets in the consequent on the other

:3D Matrix visulization

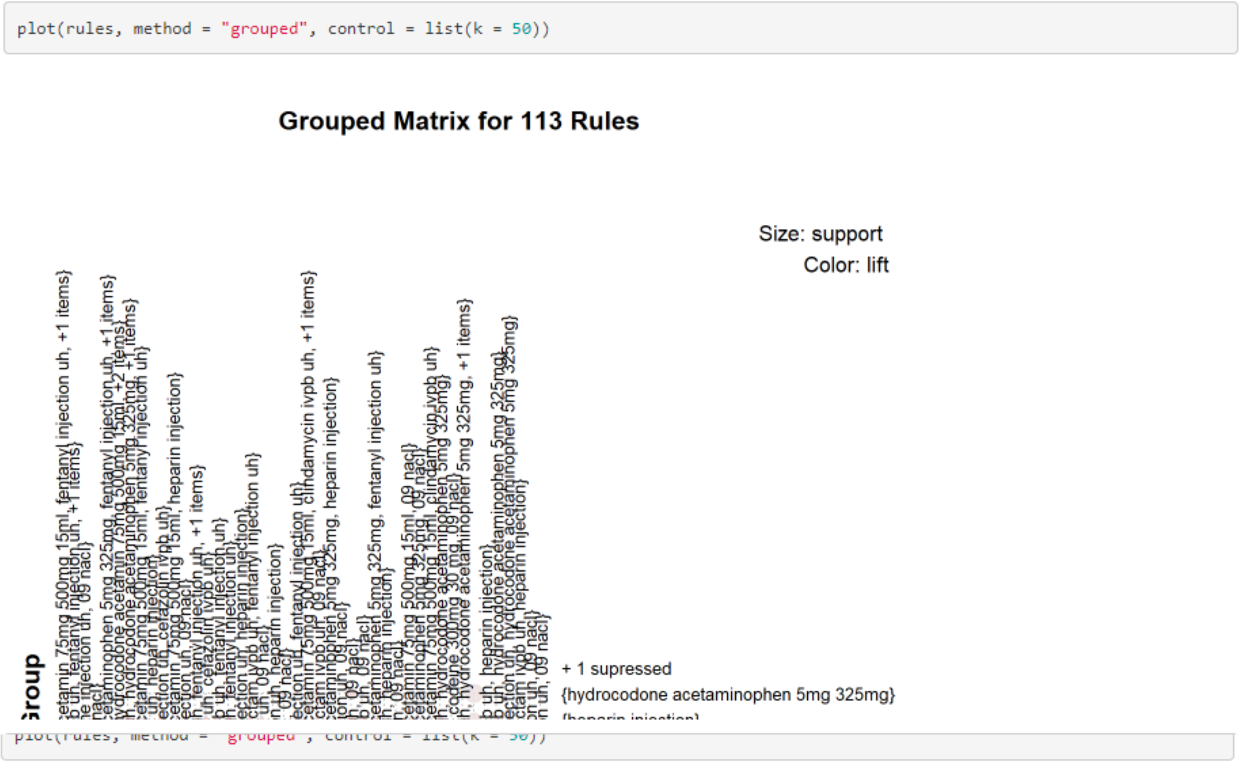




```
## [18] "{cefazolin ivpb uh,fentanyl injection uh,heparin injection}"
## [19] "{cefazolin ivpb uh,heparin injection}"
## [20] "{acetaminophen uh,cefazolin ivpb uh}"
## [21] "{cefazolin ivpb uh,fentanyl injection uh}"
## Itemsets in Consequent (RHS)
```

Matrix3D Arranges the association rules as a matrix with the itemsets in the antecedents on one axis, the itemsets in the consequent on the other and lift in the third dimension

:grouped matrix plot



```
plot(rules, method = "grouped", control = list(k = 50))
```

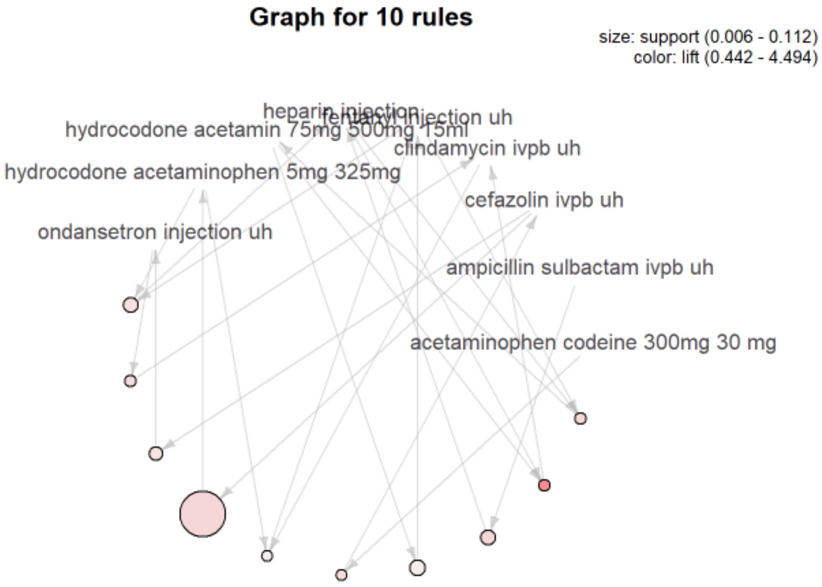
Grouped Matrix for 113 Rules
Size: support
Color: lift
Group
+ 1 supressed
{hydrocodone acetaminophen 5mg 325mg}

plot(rules, method = "grouped", control = list(k = 50))

Grouped matrix-based visualization Antecedents (columns) in the matrix are grouped using clustering. Groups are represented by the most interesting item (highest ratio of support in the group to support in all rules) in the group. Balloons in the matrix are used to represent with what consequent the antecedents are connected

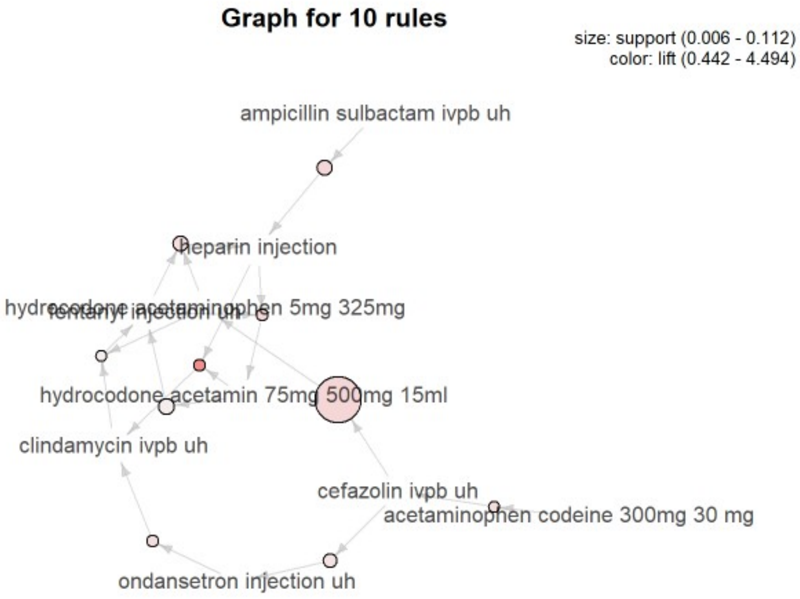
:Visualization of rules using Graphs

```
plot(subrules2, method="graph", control=list(layout=igraph::in_circle()))
```


Graph for 10 rules
size: support (0.006 - 0.112)
color: lift (0.442 - 4.494)
heparin injection
fentanyl injection uh
hydrocodone acetamin 75mg 500mg 15ml
clindamycin ivpb uh
hydrocodone acetaminophen 5mg 325mg
cefazolin ivpb uh
ondansetron injection uh
ampicillin sulbactam ivpb uh
acetaminophen codeine 300mg 30 mg

```
plot(subrules2, method="graph", control=list(layout=igraph::with_graphopt(spring.const=5, mass=50)))
```

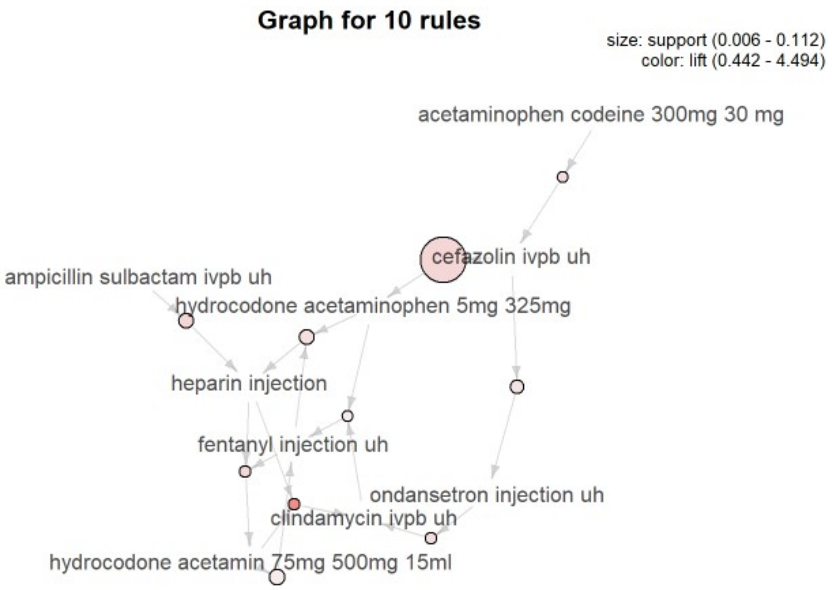
```
plot(subrules2, method="graph", control=list(layout=igraph::with_graphopt(spring.const=5, mass=50)))
```


Graph for 10 rules
size: support (0.006 - 0.112)
color: lift (0.442 - 4.494)
ampicillin sulbactam ivpb uh
heparin injection
hydrocodone acetaminophen 5mg 325mg
fentanyl injection uh
hydrocodone acetamin 75mg 500mg 15ml
clindamycin ivpb uh
cefazolin ivpb uh
acetaminophen codeine 300mg 30 mg
ondansetron injection uh

```
plot(subrules2, method="graph", control=list(type="itemsets"))
```

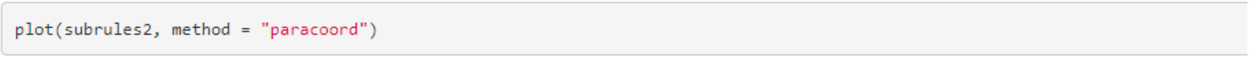
```
## Warning: Unknown control parameters: type
```

```
## Available control parameters (with default values):
## main  =  Graph for 10 rules
## nodeColors    =  c("#66CC6680", "#9999CC80")
## nodeCol   =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#EE1515FF", "#
EE1818FF", "#EE1B1BFF", "#EE1E1EFF", "#EE2222FF", "#EE2525FF", "#EE2828FF", "#EE2B2BFF", "#EE2E2EFF", "#EE3131FF", "#EE3434F
F", "#EE3737FF", "#EE3A3AFF", "#EE3D3DFF", "#EE4040FF", "#EE4444FF", "#EE4747FF", "#EE4A4AFF", "#EE4D4DFF", "#EE5050FF", "#E
E5353FF", "#EE5656FF", "#EE5959FF", "#EE5C5CFF", "#EE5F5FFF", "#EE6262FF", "#EE6666FF", "#EE6969FF", "#EE6C6CFF", "#EE6F6FF
F", "#EE7272FF", "#EE7575FF",  "#EE7878FF", "#EE7B7BFF", "#EE7E7EFF", "#EE8181FF", "#EE8484FF", "#EE8888FF", "#EE8B8BFF", "#
EE8E8EFF", "#EE9191FF", "#EE9494FF", "#EE9797FF", "#EE9999FF", "#EE9B9BFF", "#EE9D9DFF", "#EE9F9FFF", "#EEA0A0FF", "#EEA2A2F
F", "#EEA4A4FF", "#EEA5A5FF", "#EEA7A7FF", "#EEA9A9FF", "#EEABABFF", "#EEACACFF", "#EEAEAEFF", "#EEB0B0FF", "#EEB1B1FF", "#E
EB3B3FF", "#EEB5B5FF", "#EEB7B7FF", "#EEB8B8FF", "#EEBABAFF", "#EEBCBCFF", "#EEBDBDFF", "#EEBFBFFF", "#EEC1C1FF", "#EEC3C3F
F", "#EEC4C4FF", "#EEC6C6FF", "#EEC8C8FF",  "#EEC9C9FF", "#EECBCBFF", "#EECDCDFF", "#EECFCFFF", "#EED0D0FF", "#EED2D2FF", "#
EED4D4FF", "#EED5D5FF", "#EED7D7FF", "#EED9D9FF", "#EEDBDBFF", "#EEDCDCFF", "#EEDEDEFF", "#EEE0E0FF", "#EEE1E1FF", "#EEE3E3F
F", "#EEE5E5FF", "#EEE7E7FF", "#EEE8E8FF", "#EEEAEAFF", "#EEECECFF", "#EEEEEEFF")
## edgeCol   =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353FF", "#555555FF", "#
575757FF", "#595959FF", "#5B5B5BFF", "#5E5E5EFF", "#606060FF", "#626262FF", "#646464FF", "#666666FF", "#686868FF", "#6A6A6AF
F", "#6C6C6CFF", "#6E6E6EFF", "#707070FF", "#727272FF", "#747474FF", "#767676FF", "#787878FF", "#7A7A7AFF", "#7C7C7CFF", "#7
E7E7EFF", "#808080FF", "#828282FF", "#848484FF", "#868686FF", "#888888FF", "#8A8A8AFF", "#8C8C8CFF", "#8D8D8DFF", "#8F8F8FF
F", "#919191FF", "#939393FF",  "#959595FF", "#979797FF", "#999999FF", "#9A9A9AFF", "#9C9C9CFF", "#9E9E9EFF", "#A0A0A0FF", "#
A2A2A2FF", "#A3A3A3FF", "#A5A5A5FF", "#A7A7A7FF", "#A9A9A9FF", "#AAAAAAFF", "#ACACACFF", "#AEAEAEFF", "#AFAFAFFF", "#B1B1B1F
F", "#B3B3B3FF", "#B4B4B4FF", "#B6B6B6FF", "#B7B7B7FF", "#B9B9B9FF", "#BBBBBBFF", "#BCBCBCFF", "#BEBEBEFF", "#BFBFBFFF", "#C
1C1C1FF", "#C2C2C2FF", "#C3C3C4FF", "#C5C5C5FF", "#C6C6C6FF", "#C8C8C8FF", "#C9C9C9FF", "#CACACAFF", "#CCCCCCFF", "#CDCDCDF
F", "#CECECEFF", "#CFCFCFFF", "#D1D1D1FF",  "#D2D2D2FF", "#D3D3D3FF", "#D4D4D4FF", "#D5D5D5FF", "#D6D6D6FF", "#D7D7D7FF", "#
D8D8D8FF", "#D9D9D9FF", "#DADADAFF", "#DBDBDBFF", "#DCDCDCFF", "#DDDDDDFF", "#DEDEDEFF", "#DEDEDEFF", "#DFDFDFFF", "#E0E0E0F
F", "#E0E0E0FF", "#E1E1E1FF", "#E1E1E1FF", "#E2E2E2FF", "#E2E2E2FF", "#E2E2E2FF")
## alpha     =  0.5
## cex   =  1
```

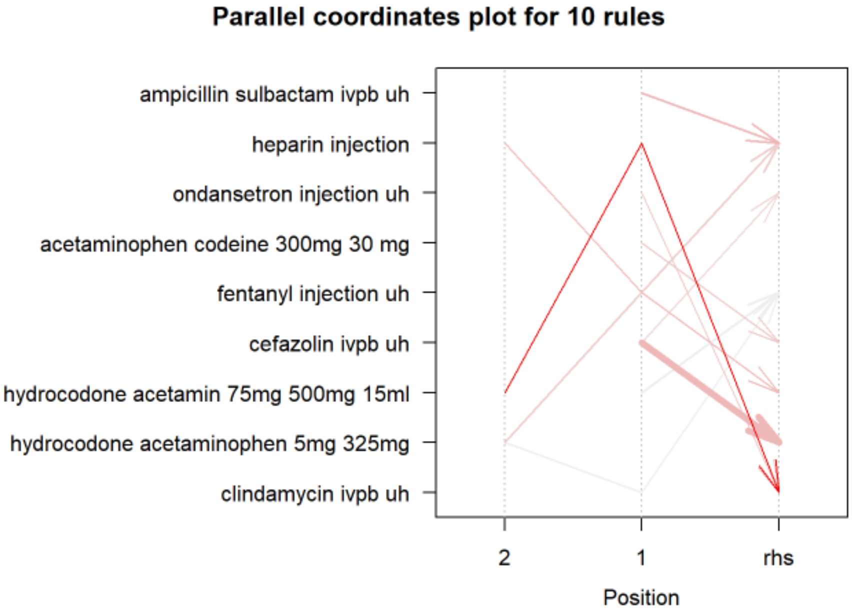

Graph for 10 rules
size: support (0.006 - 0.112)
color: lift (0.442 - 4.494)
acetaminophen codeine 300mg 30 mg
cefazolin ivpb uh
ampicillin sulbactam ivpb uh
hydrocodone acetaminophen 5mg 325mg
heparin injection
fentanyl injection uh
ondansetron injection uh
clindamycin ivpb uh
hydrocodone acetamin 75mg 500mg 15ml

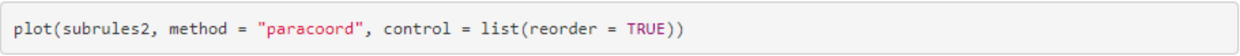
It represents the rules (or itemsets) as a graph with items as labeled vertices, and rules (or itemsets) represented as vertices connected to items using arrows. For rules, the LHS items are connected with arrows pointing to the vertex representing the rule and the RHS has an arrow pointing to the item

: Visualization of rules using parallel coordinates

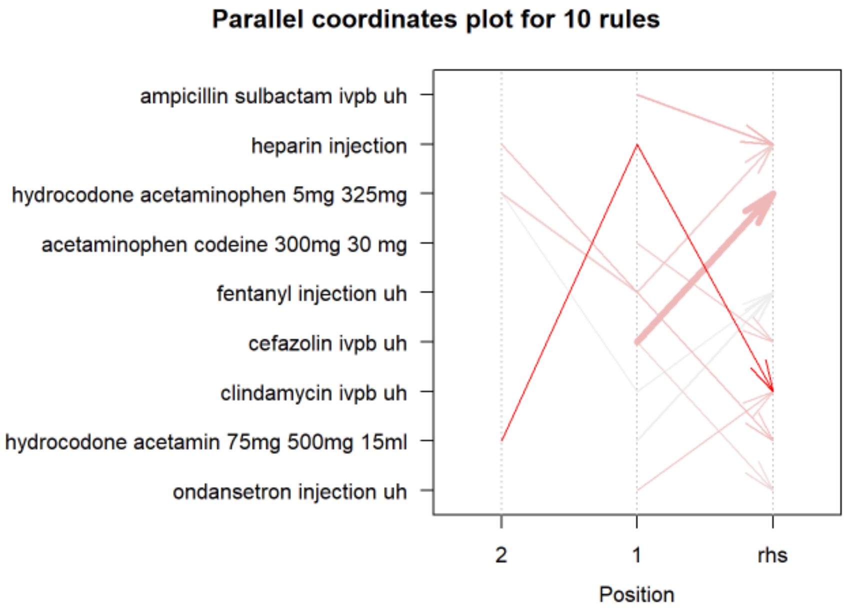


```
plot(subrules2, method = "paracoord")
```



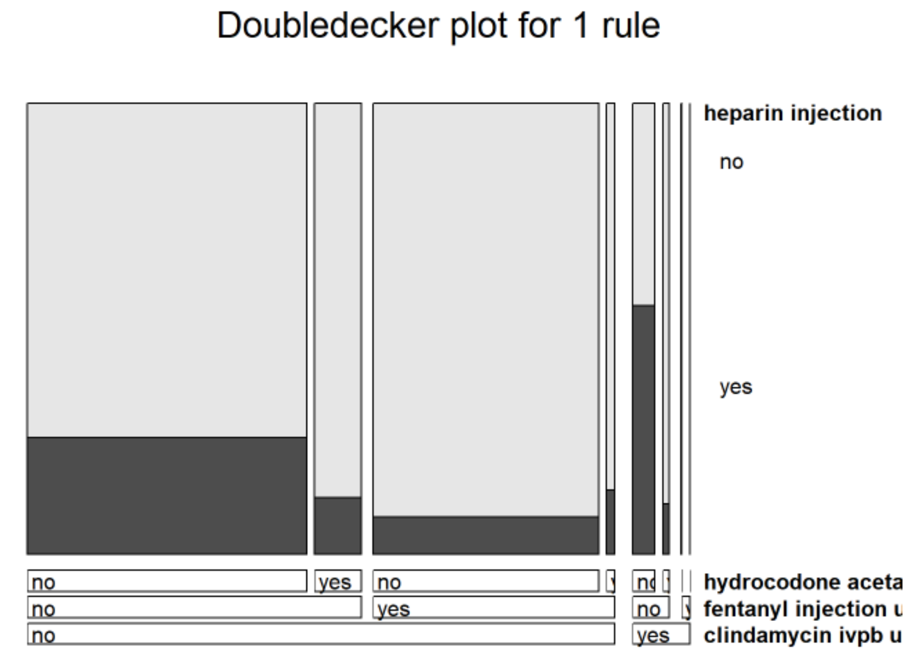


```
plot(subrules2, method = "paracoord", control = list(reorder = TRUE))
```



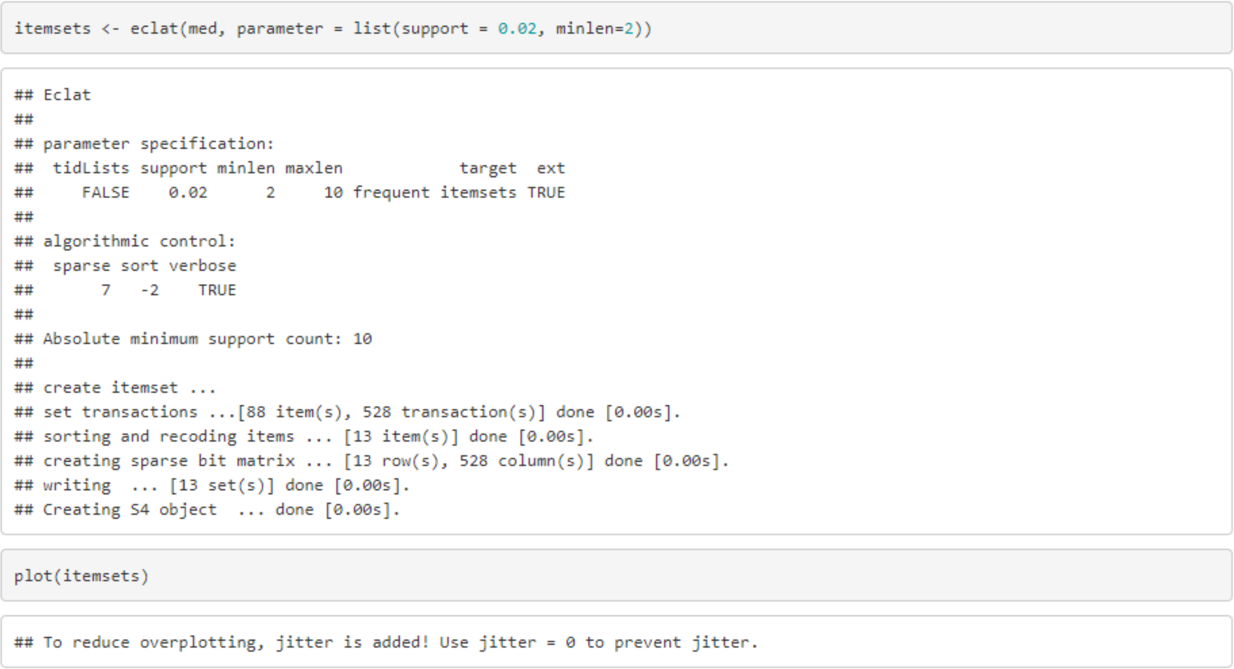
It represents the rules as a parallel coordinate plot.

:Visualization of one rule using doubledecker plot



It represents a single rule as a doubledecker or mosaic plot. Parameter data has to be specified to compute the needed contingency table.

:Itemset Visualization as Graph

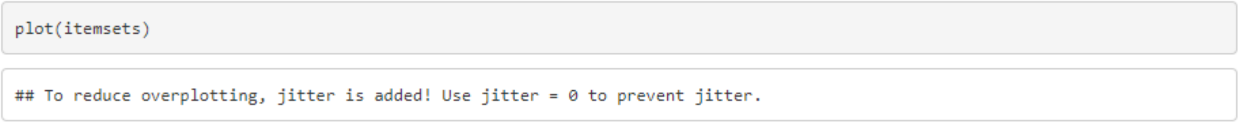


```
itemsets <- eclat(med, parameter = list(support = 0.02, minlen=2))
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target  ext
##     FALSE    0.02      2     10 frequent itemsets TRUE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 10
##
## create itemset ...
## set transactions ...[88 item(s), 528 transaction(s)] done [0.00s].
## sorting and recoding items ... [13 item(s)] done [0.00s].
## creating sparse bit matrix ... [13 row(s), 528 column(s)] done [0.00s].
## writing  ... [13 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

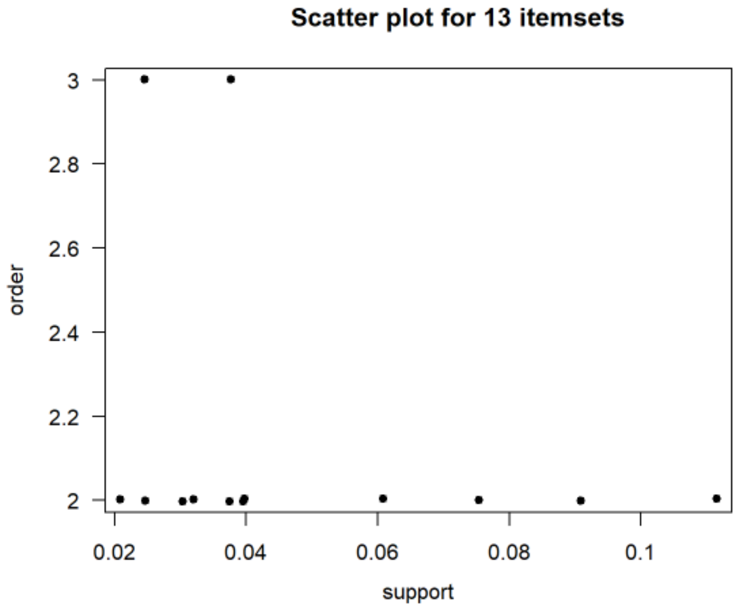
```
plot(itemsets)
```

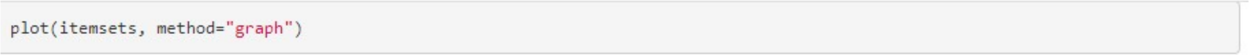
```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



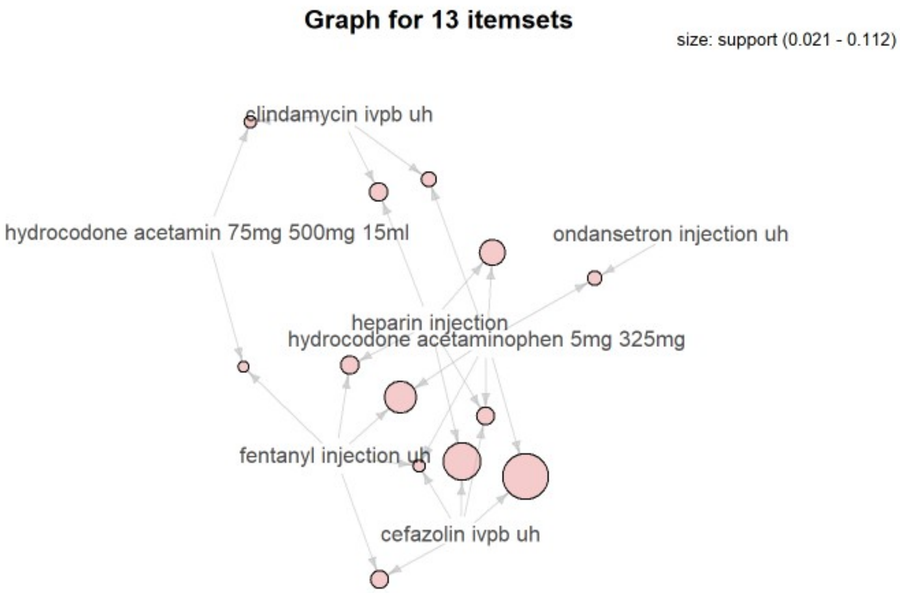
```
plot(itemsets)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```

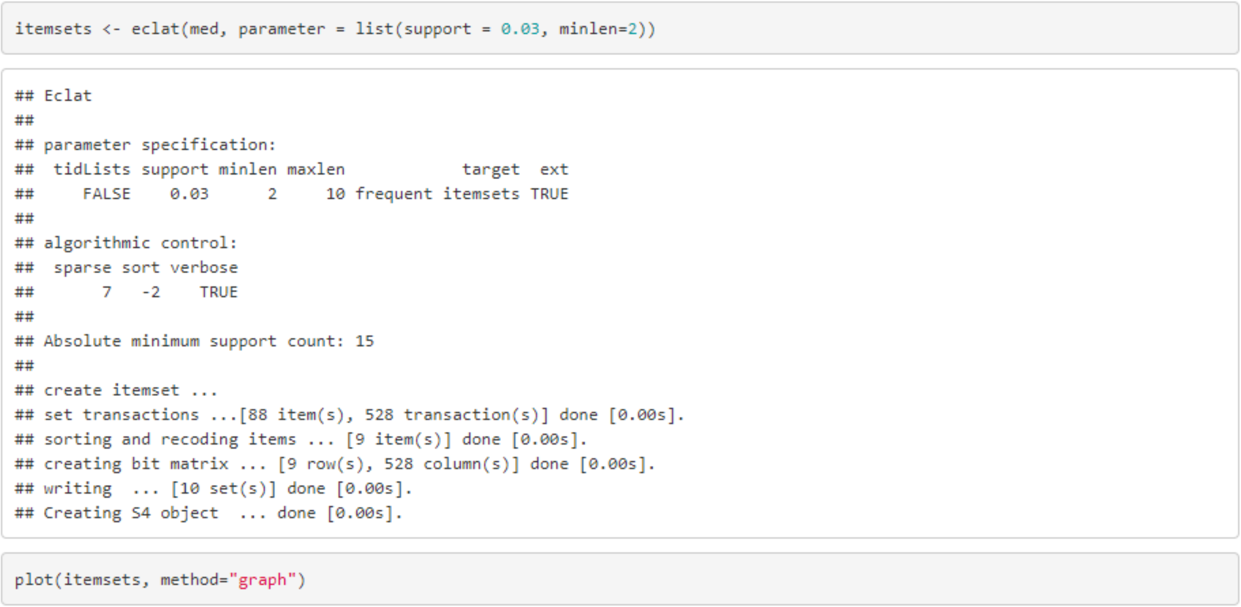




```
plot(itemsets, method="graph")
```



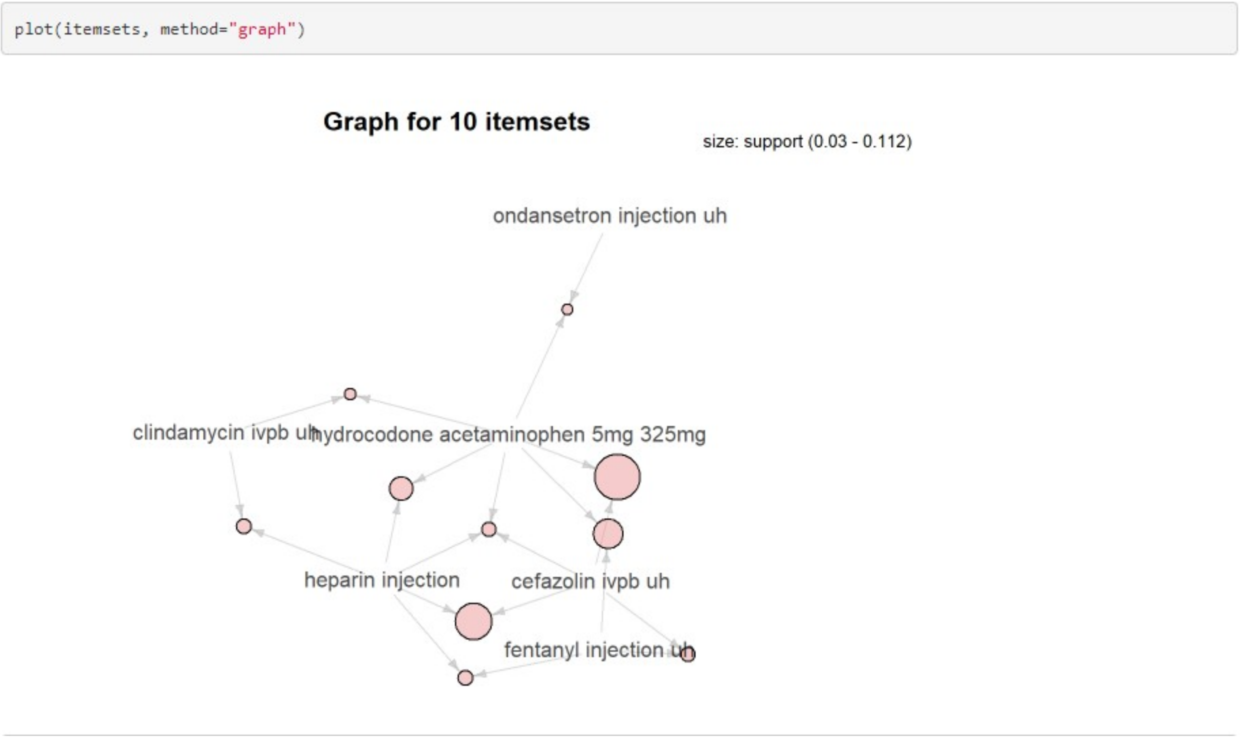
The eclat() takes in a transactions object and gives the most frequent items in the data based the support argument. The maxlen defines the maximum number of items in each itemset of frequent items.This plot used to represent the most frequent items in the dataset as graph with support value as 0.03 and length as 2



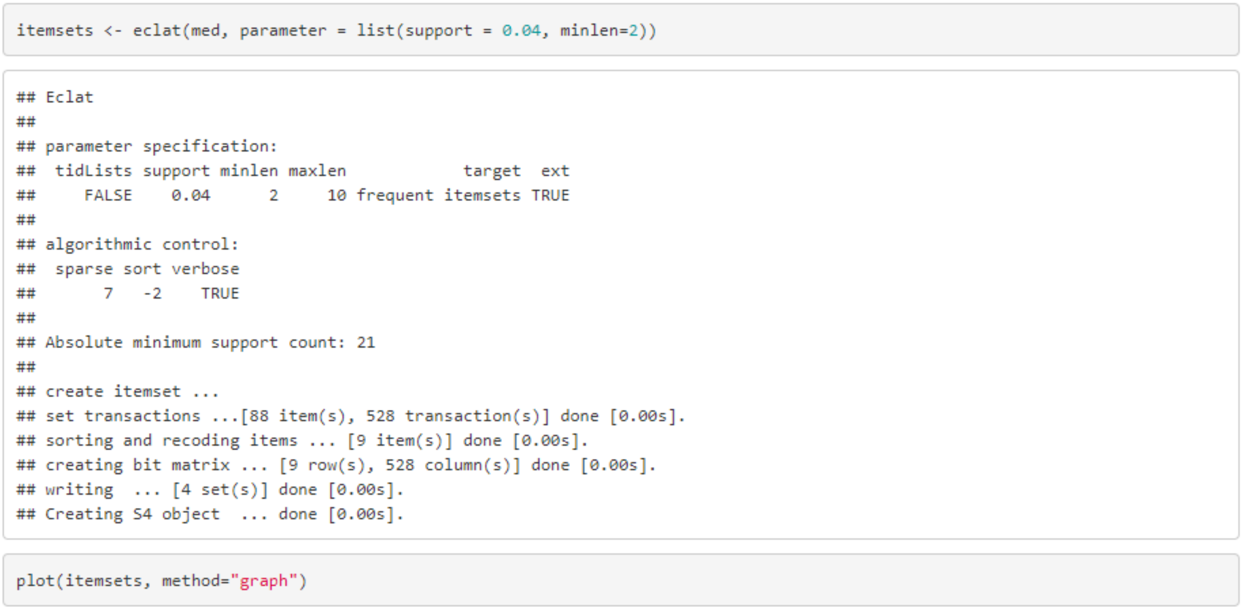
```
itemsets <- eclat(med, parameter = list(support = 0.03, minlen=2))
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target  ext
##     FALSE    0.03      2     10 frequent itemsets TRUE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 15
##
## create itemset ...
## set transactions ...[88 item(s), 528 transaction(s)] done [0.00s].
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating bit matrix ... [9 row(s), 528 column(s)] done [0.00s].
## writing  ... [10 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```
plot(itemsets, method="graph")
```



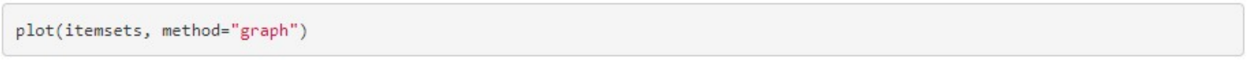
```
plot(itemsets, method="graph")
```



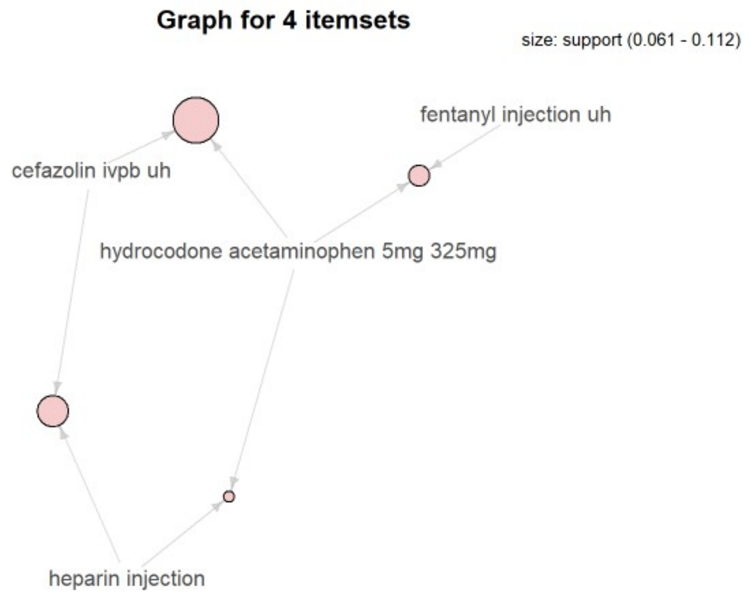
```
itemsets <- eclat(med, parameter = list(support = 0.04, minlen=2))
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target  ext
##     FALSE    0.04      2     10 frequent itemsets TRUE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 21
##
## create itemset ...
## set transactions ...[88 item(s), 528 transaction(s)] done [0.00s].
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating bit matrix ... [9 row(s), 528 column(s)] done [0.00s].
## writing  ... [4 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```
plot(itemsets, method="graph")
```

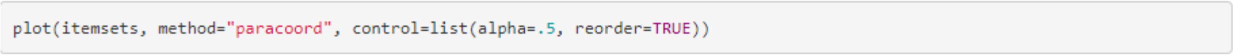


```
plot(itemsets, method="graph")
```

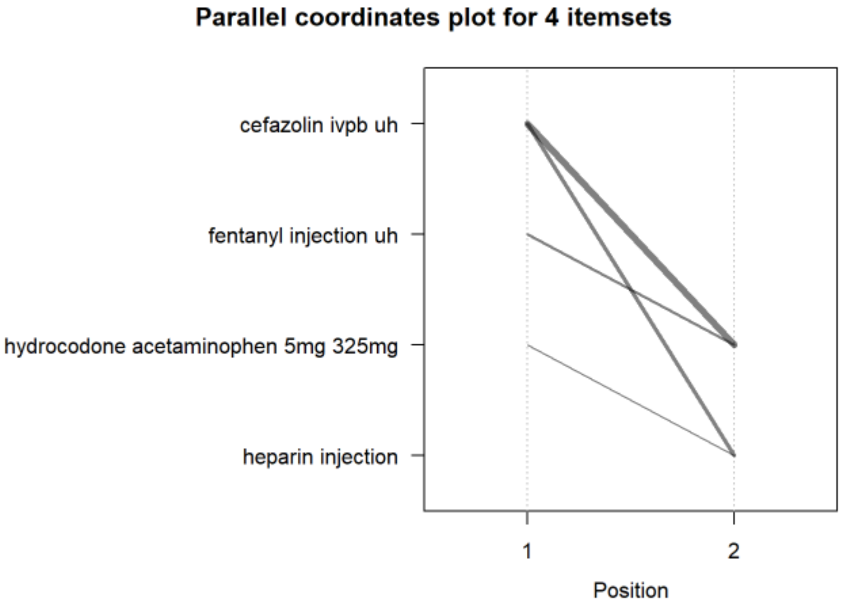


The eclat() takes in a transactions object and gives the most frequent items in the data based the support argument. The maxlen defines the maximum number of items in each itemset of frequent items.This plot used to represent the most frequent items in the dataset as graph with support value as 0.03,0.04 respectively and length as 2

:Itemset visualization using parallel coordinates

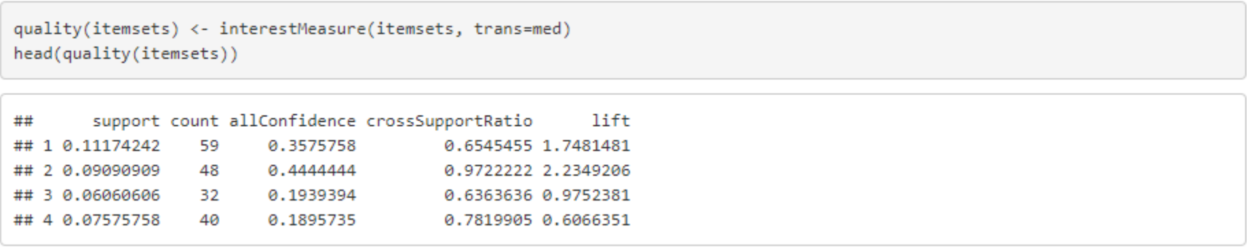


```
plot(itemsets, method="paracoord", control=list(alpha=.5, reorder=TRUE))
```



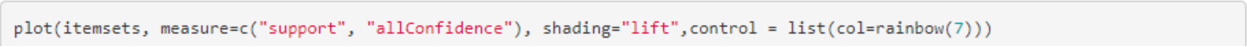
It represents the itemset as a parallel coordinate plot.

:Add more quality measures to the scatterplot

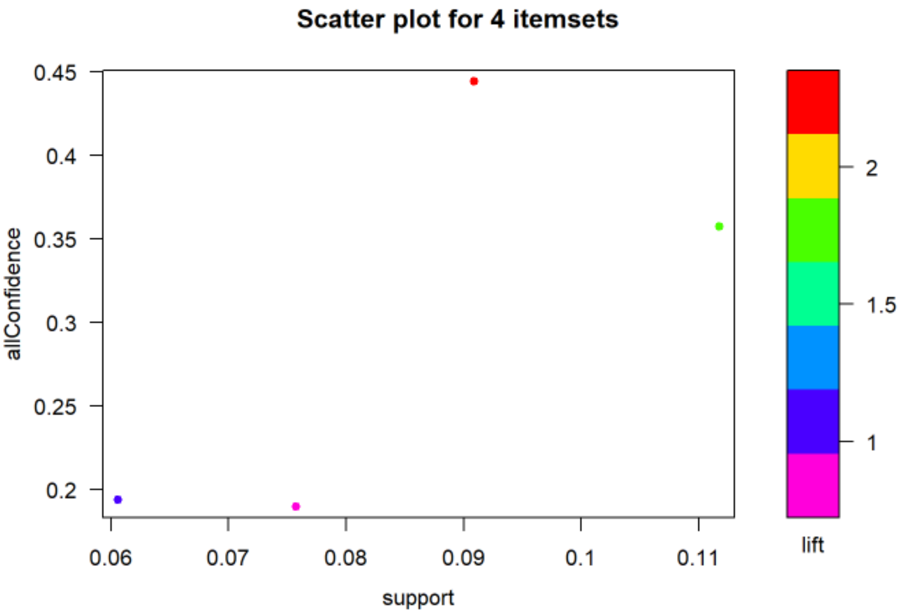


```
quality(itemsets) <- interestMeasure(itemsets, trans=med)
head(quality(itemsets))
```

```
##      support count allConfidence crossSupportRatio      lift
## 1 0.11174242    59     0.3575758         0.6545455 1.7481481
## 2 0.09090909    48     0.4444444         0.9722222 2.2349206
## 3 0.06060606    32     0.1939394         0.6363636 0.9752381
## 4 0.07575758    40     0.1895735         0.7819905 0.6066351
```



```
plot(itemsets, measure=c("support", "allConfidence"), shading="lift",control = list(col=rainbow(7)))
```



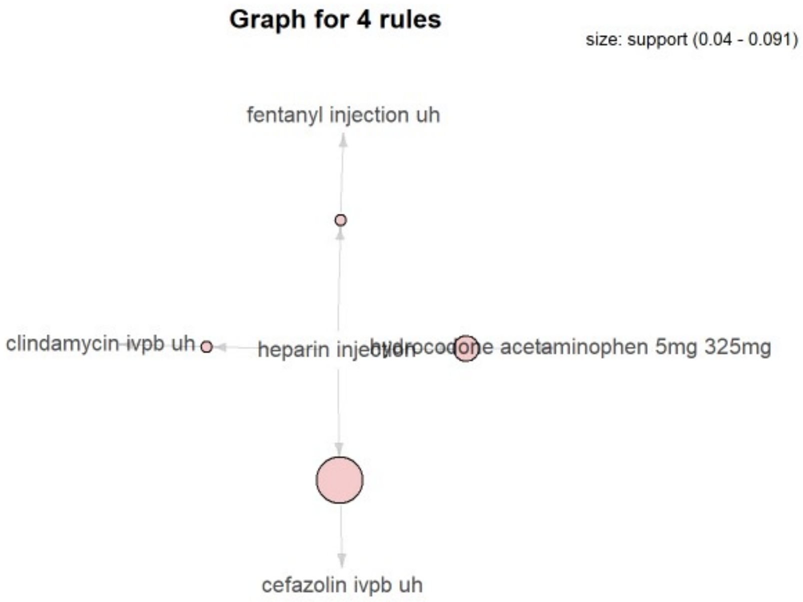
It represents the itemsets with various support and confidence values and with rainbow color using the lift parameter

:Visulization of rules with Left hand side value as `heparin injection`

```
rules<-apriori(data=med, parameter=list(supp=0.001,conf = 0.15,minlen=2),
               appearance = list(default="rhs",lhs="heparin injection"),
               control = list(verbose=F))
rules<-sort(rules, decreasing=TRUE,by="confidence")
inspect(rules[1:3])
```

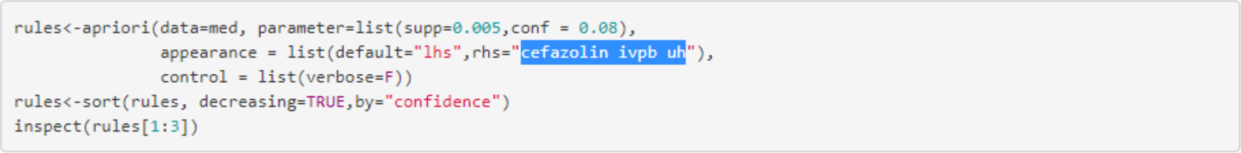
```
##     lhs                    rhs                                      support confidence   coverage      lift count
## [1] {heparin injection} => {cefazolin ivpb uh}                   0.09090909  0.4571429 0.1988636 2.2349206    48
## [2] {heparin injection} => {hydrocodone acetaminophen 5mg 325mg} 0.06060606  0.3047619 0.1988636 0.9752381    32
## [3] {heparin injection} => {clindamycin ivpb uh}                 0.03977273  0.2000000 0.1988636 2.2468085    21
```

```
library(arulesViz)
plot(rules,method="graph",engine = 'default',shading=NA)
```

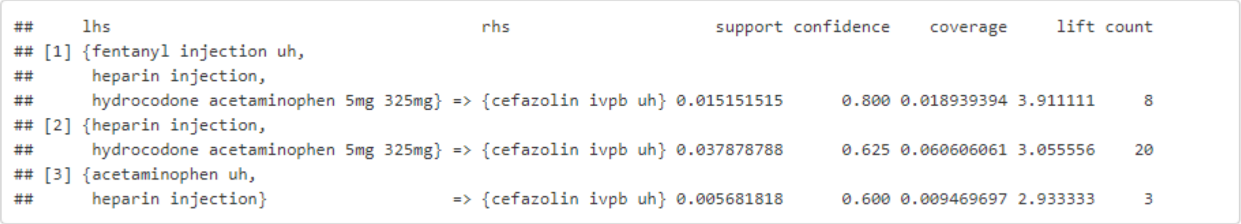


It is used find out the Customers who bought 'Whole Milk' also bought other item

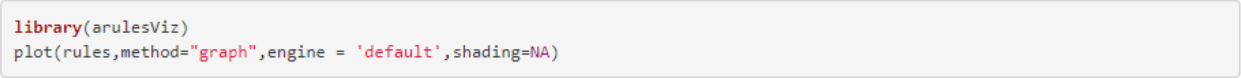
:Visualization of rules with Right Hand Side value as `cefazolin ivpb uh`



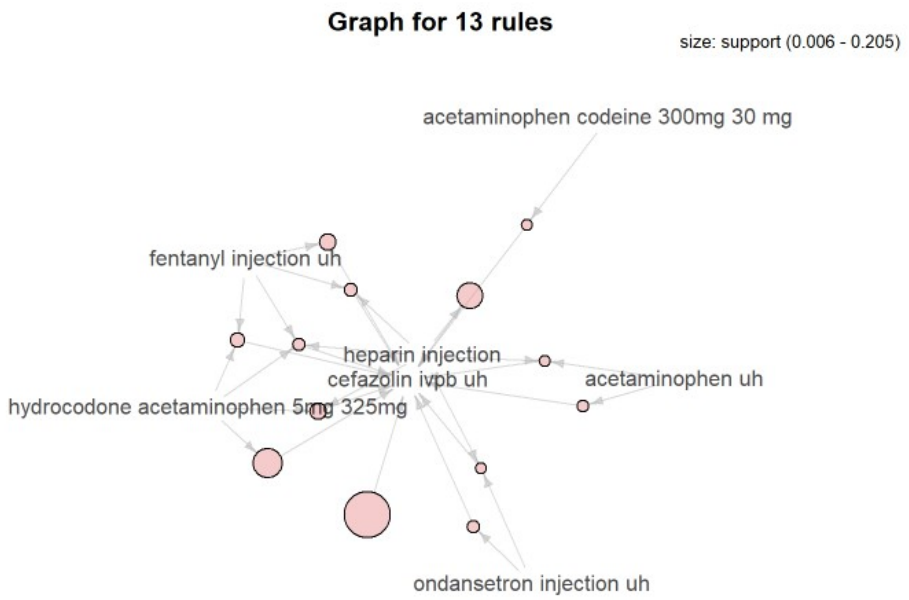
```
rules<-apriori(data=med, parameter=list(supp=0.005,conf = 0.08),
               appearance = list(default="lhs",rhs="cefazolin ivpb uh"),
               control = list(verbose=F))
rules<-sort(rules, decreasing=TRUE,by="confidence")
inspect(rules[1:3])
```



```
##     lhs                                    rhs                     support confidence    coverage     lift count
## [1] {fentanyl injection uh,
##      heparin injection,
##      hydrocodone acetaminophen 5mg 325mg} => {cefazolin ivpb uh} 0.015151515      0.800 0.018939394 3.911111     8
## [2] {heparin injection,
##      hydrocodone acetaminophen 5mg 325mg} => {cefazolin ivpb uh} 0.037878788      0.625 0.060606061 3.055556    20
## [3] {acetaminophen uh,
##      heparin injection}                   => {cefazolin ivpb uh} 0.005681818      0.600 0.009469697 2.933333     3
```



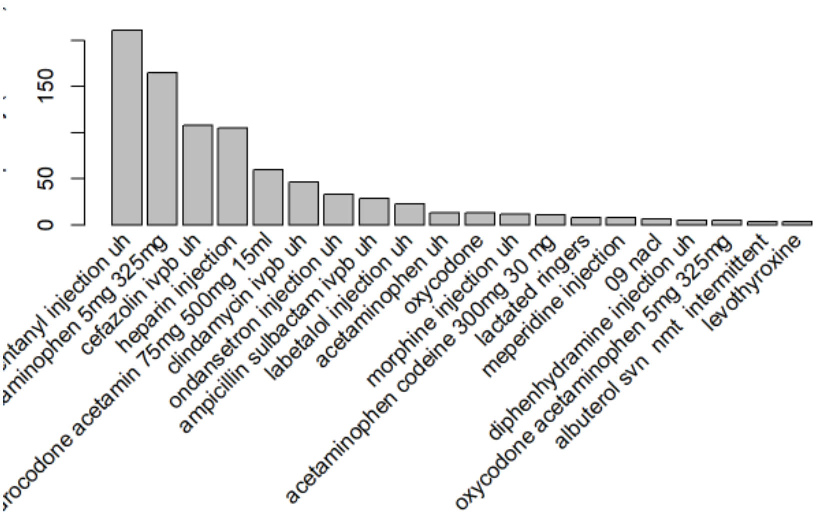
```
library(arulesViz)
plot(rules,method="graph",engine = 'default',shading=NA)
```



It represents the what customers had purchased before buying ‘`cefazolin ivpb uh`’. This will help you understand the patterns that led to the purchase of ‘`cefazolin ivpb uh`’

:Create an item frequency plot for top 20 items

```
itemFrequencyPlot(med,topN=20,type="absolute")
```


ntanyl injection uh
aminophen 5mg 325mg
cefazolin ivpb uh
heparin injection
rocodone acetamin 75mg 500mg 15ml
clindamycin ivpb uh
ondansetron injection uh
ampicillin sulbactam ivpb uh
labetalol injection uh
acetaminophen uh
oxycodone
morphine injection uh
acetaminophen codeine 300mg 30 mg
lactated ringers
meperidine injection
09 nacl
diphenhydramine injection uh
oxycodone acetaminophen 5mg 325mg
albuterol svn nmt intermittent
levothyroxine