Лабораторная работа №5

Голуб Светлана

## Базовые понятия

**Ассоциативные правила** - описание связей между переменными в ннекотором списке транзакций. Они используются для нахождения абстрактных ассоциаций в датасете, которые используются в дальнейшем анализе данных.

**Support (поддержка)** - частота появления множества элементов во всех анализируемых транзакциях.

**Confidence (достоверность)** - показатель того, как часто правило сробатывает для всего датасета. Чем больше Confidence, тем интереснее правило.

**Lift** - показатель того, насколько элементы в множестве зависят друг от друга. Чем больше lift, тем больше элементы зависят друг от друга и тем сильнее правило.

## Ход работы

Для начала необходимо выгрузить данные из csv файла в виде объекта класса транзакций:

library(arules)

## Loading required package: Matrix

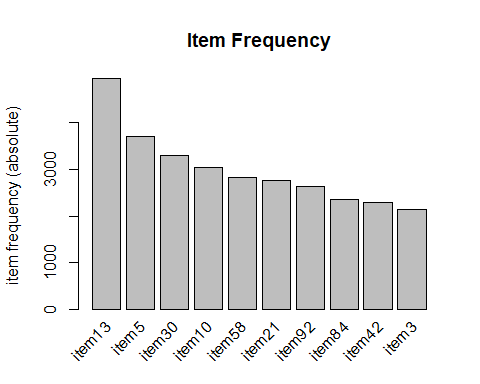
##   
## Attaching package: 'arules'

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

Associations <- read.transactions("AssociationRules.csv", header = FALSE)

1. Построим диаграмму частоты появления каждого элемента в транзакциях:

itemFrequencyPlot(Associations, topN=10, type="absolute", main="Item Frequency")



Наиболее часто встречается: **item13**

1. Выведем общий отчёт по нашим данным:

summary(Associations)

## transactions as itemMatrix in sparse format with  
## 10000 rows (elements/itemsets/transactions) and  
## 98 columns (items) and a density of 0.1000643   
##   
## most frequent items:  
## item13 item5 item30 item10 item58 (Other)   
## 4948 3699 3308 3035 2831 80242   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16   
## 17 88 176 319 490 660 858 1045 1132 1120 1079 859 675 520 398 249   
## 17 18 19 20 21 22 23 24 25   
## 133 97 41 22 7 9 2 1 3   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 7.000 10.000 9.806 12.000 25.000   
##   
## includes extended item information - examples:  
## labels  
## 1 item1  
## 2 item10  
## 3 item100

Из отчёта видим, что в наибольшей транзакции было **25 элементов**

Выводим правила с минимальной поддержкой - 1% и с минимальной достоверностью - 0%. Для этого используем алгоритм Априори (если множестово элементов встречается часто, то любое подмножество так же встречается часто):

rules = apriori(data = Associations, parameter = list(support = 0.01, confidence = 0))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0 0.1 1 none FALSE TRUE 5 0.01 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: 100   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[98 item(s), 10000 transaction(s)] done [0.01s].  
## sorting and recoding items ... [89 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 done [0.02s].  
## writing ... [11524 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

При таких параметрах мы получаем **11524** правил

Изменим минимальный процент Confidence на 50%:

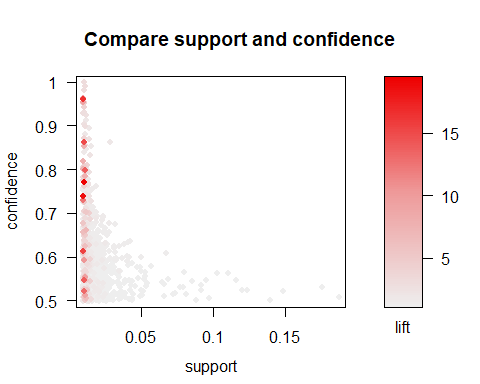
rules = apriori(data = Associations, parameter = list(support = 0.01, confidence = 0.5))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.5 0.1 1 none FALSE TRUE 5 0.01 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: 100   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[98 item(s), 10000 transaction(s)] done [0.00s].  
## sorting and recoding items ... [89 item(s)] done [0.00s].  
## creating transaction tree ... done [0.01s].  
## checking subsets of size 1 2 3 4 5 done [0.02s].  
## writing ... [1165 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

Получили **1165** правил.

1. С увиличением параметра Confidence количество правил уменьшается, так как повышается минимальная частота срабатывания правила на всём датасете.
2. Создадим диаграмму зависимости Support от Confidence с затемнением значений по шкале lift:

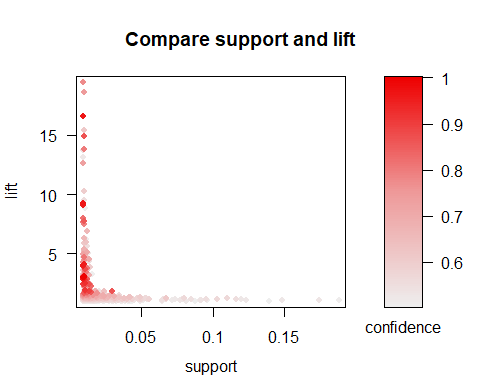
library(arulesViz)  
plot(rules, measure = c("support", "confidence"), shading = "lift", main = "Compare support and confidence")



Наиболее интересные правила расположены на границе Support и Confidence Чем больше Support, тем чаще оно всетречается в датасете. Чем больше Confidence правила, тем оно чаще срабатывает.

1. Создадим диаграмму зависимости Support от lift с затемнением значений по шкале Confidence:

plot(rules, measure = c("support", "lift"), shading = "confidence", main = "Compare support and lift")



1. Наиболее сильные правила расположены в левом верхнем углу диаграммы. Чем больще lift, тем сильнее элементы зависят друг от друга.
2. Главный недостаток алгоритма состоит в том, что генерируется большое количество неинтересных правил, которые занимают память и на которые тратится относительно много времени.
3. Для того, чтобы использовать интерактивный режим диаграммы, необходимо добавить параметр engine = “htmlwidget” или engine=‘interactive’:
4. **engine = “htmlwidget”**

plot(rules, measure = c("support", "lift"), shading = "confidence", main = "Compare support and lift", engine = "htmlwidget")

## Warning: plot: Too many rules supplied. Only plotting the best 1000 rules using  
## measure confidence (change parameter max if needed)

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

## Warning: `arrange\_()` is deprecated as of dplyr 0.7.0.  
## Please use `arrange()` instead.  
## See vignette('programming') for more help  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

1. **engine=‘interactive’**

plot(rules, measure = c(“support”, “lift”), shading = “confidence”, main = “Compare support and lift”, engine=‘interactive’)

Выделяем правила, Confidence которых дольше **10%**: - {item5} => {item13} - {item30} => {item13} - {item58} => {item13}

Рассмотрим правила, в которых Confidence больше 80%:

rules = apriori(data = Associations, parameter = list(support = 0.01, confidence = 0.8))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.01 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: 100   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[98 item(s), 10000 transaction(s)] done [0.01s].  
## sorting and recoding items ... [89 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 done [0.02s].  
## writing ... [40 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

Отсортируем по значению lift:

rules.sorted <- sort(rules, decreasing = TRUE, na.last = NA, by = "lift")  
inspect(rules.sorted)

## lhs rhs support confidence coverage lift   
## [1] {item15,item30,item49} => {item56} 0.0101 0.9619048 0.0105 16.584565  
## [2] {item15,item49} => {item56} 0.0101 0.8632479 0.0117 14.883584  
## [3] {item30,item49,item84} => {item56} 0.0100 0.8000000 0.0125 13.793103  
## [4] {item30,item49,item56} => {item15} 0.0101 0.9619048 0.0105 9.240199  
## [5] {item49,item56} => {item15} 0.0101 0.9528302 0.0106 9.153028  
## [6] {item30,item56,item77} => {item15} 0.0100 0.8196721 0.0122 7.873892  
## [7] {item55} => {item34} 0.0100 0.8547009 0.0117 7.693077  
## [8] {item30,item49,item84} => {item15} 0.0100 0.8000000 0.0125 7.684918  
## [9] {item30,item49,item84} => {item77} 0.0101 0.8080000 0.0125 4.651698  
## [10] {item30,item49,item56} => {item84} 0.0100 0.9523810 0.0105 4.026981  
## [11] {item15,item30,item49} => {item84} 0.0100 0.9523810 0.0105 4.026981  
## [12] {item49,item56} => {item84} 0.0100 0.9433962 0.0106 3.988990  
## [13] {item10,item22,item41} => {item3} 0.0118 0.8082192 0.0146 3.789119  
## [14] {item15,item49} => {item84} 0.0102 0.8717949 0.0117 3.686236  
## [15] {item15,item30,item56} => {item84} 0.0106 0.8091603 0.0131 3.421397  
## [16] {item20,item25,item41} => {item92} 0.0100 0.8064516 0.0124 3.066356  
## [17] {item15,item49,item56} => {item30} 0.0101 1.0000000 0.0101 3.022975  
## [18] {item49,item56,item84} => {item30} 0.0100 1.0000000 0.0100 3.022975  
## [19] {item49,item56} => {item30} 0.0105 0.9905660 0.0106 2.994456  
## [20] {item15,item49,item84} => {item30} 0.0100 0.9803922 0.0102 2.963701  
## [21] {item15,item56,item77} => {item30} 0.0100 0.9523810 0.0105 2.879023  
## [22] {item22,item3,item41} => {item10} 0.0118 0.8550725 0.0138 2.817372  
## [23] {item15,item56,item84} => {item30} 0.0106 0.9298246 0.0114 2.810836  
## [24] {item49,item77,item84} => {item30} 0.0101 0.9266055 0.0109 2.801105  
## [25] {item15,item49} => {item30} 0.0105 0.8974359 0.0117 2.712926  
## [26] {item16,item61,item77} => {item5} 0.0108 0.9230769 0.0117 2.495477  
## [27] {item16,item34,item77} => {item5} 0.0102 0.9026549 0.0113 2.440267  
## [28] {item13,item82,item99} => {item5} 0.0134 0.8701299 0.0154 2.352338  
## [29] {item25,item34,item77} => {item5} 0.0103 0.8583333 0.0120 2.320447  
## [30] {item82,item99} => {item5} 0.0150 0.8333333 0.0180 2.252861  
## [31] {item16,item25,item77} => {item5} 0.0104 0.8062016 0.0129 2.179512  
## [32] {item20,item23} => {item13} 0.0114 0.9120000 0.0125 1.843169  
## [33] {item5,item82,item99} => {item13} 0.0134 0.8933333 0.0150 1.805443  
## [34] {item3,item84,item95} => {item13} 0.0108 0.8780488 0.0123 1.774553  
## [35] {item23} => {item13} 0.0292 0.8613569 0.0339 1.740818  
## [36] {item82,item99} => {item13} 0.0154 0.8555556 0.0180 1.729094  
## [37] {item10,item44} => {item13} 0.0101 0.8487395 0.0119 1.715318  
## [38] {item83} => {item13} 0.0119 0.8439716 0.0141 1.705682  
## [39] {item23,item5} => {item13} 0.0105 0.8400000 0.0125 1.697656  
## [40] {item30,item95,item96} => {item13} 0.0118 0.8027211 0.0147 1.622314  
## count  
## [1] 101   
## [2] 101   
## [3] 100   
## [4] 101   
## [5] 101   
## [6] 100   
## [7] 100   
## [8] 100   
## [9] 101   
## [10] 100   
## [11] 100   
## [12] 100   
## [13] 118   
## [14] 102   
## [15] 106   
## [16] 100   
## [17] 101   
## [18] 100   
## [19] 105   
## [20] 100   
## [21] 100   
## [22] 118   
## [23] 106   
## [24] 101   
## [25] 105   
## [26] 108   
## [27] 102   
## [28] 134   
## [29] 103   
## [30] 150   
## [31] 104   
## [32] 114   
## [33] 134   
## [34] 108   
## [35] 292   
## [36] 154   
## [37] 101   
## [38] 119   
## [39] 105   
## [40] 118

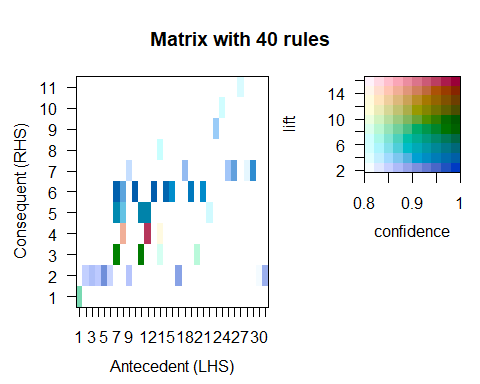
Также работать с правилами (сортировка/фильтрация) в интерактивном формате можно с помощью функции inspectDT().

Правила с небольшим значением параметра lift являются достаточно случайными, то есть элементы в этом правиле совпали случайно, потому что просто часто встречаются сами по себе.

1. Построим диаграмму в матричном виде на основе confidence и lift:

plot(rules, method="matrix", measure=c("lift", "support"), shading = c("lift", "confidence"), reorder=FALSE)

## Itemsets in Antecedent (LHS)  
## [1] "{item55}" "{item83}" "{item23}"   
## [4] "{item10,item44}" "{item20,item23}" "{item23,item5}"   
## [7] "{item49,item56}" "{item15,item49}" "{item82,item99}"   
## [10] "{item15,item49,item56}" "{item30,item49,item56}" "{item15,item30,item49}"  
## [13] "{item49,item56,item84}" "{item30,item49,item84}" "{item15,item49,item84}"  
## [16] "{item49,item77,item84}" "{item5,item82,item99}" "{item13,item82,item99}"  
## [19] "{item15,item56,item77}" "{item30,item56,item77}" "{item15,item56,item84}"  
## [22] "{item15,item30,item56}" "{item22,item3,item41}" "{item10,item22,item41}"  
## [25] "{item25,item34,item77}" "{item16,item34,item77}" "{item20,item25,item41}"  
## [28] "{item16,item25,item77}" "{item16,item61,item77}" "{item30,item95,item96}"  
## [31] "{item3,item84,item95}"   
## Itemsets in Consequent (RHS)  
## [1] "{item34}" "{item13}" "{item15}" "{item56}" "{item84}" "{item30}"  
## [7] "{item5}" "{item77}" "{item10}" "{item3}" "{item92}"



Наиболее интересными для нас правилами являются те, цвет которых тёмно-зелёный или тёмно-красный. Эти цвета значат, что эти правила встречаются достаточно часто и являются достаточно независимыми друг от друга.

1. Правила тёмно-синего цвета встречаются часто, но элементы в этом правиле сильно зависят друг от друга. То есть они появляются вместе из-за того, что часто встречаются в транзакциях сами по себе.
2. Найдём 3 правила с наибольшим значением параметра lift:

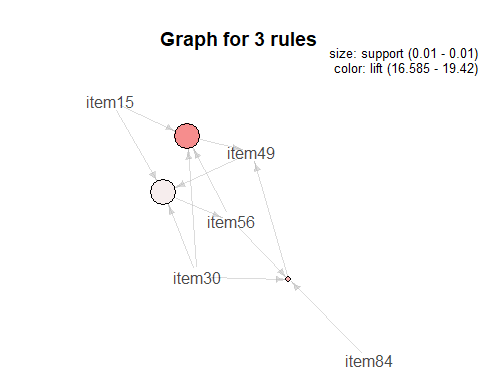
rules = apriori(data = Associations, parameter = list(support = 0.01, confidence = 0))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0 0.1 1 none FALSE TRUE 5 0.01 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: 100   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[98 item(s), 10000 transaction(s)] done [0.01s].  
## sorting and recoding items ... [89 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 done [0.03s].  
## writing ... [11524 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules.filtrated <- head(rules, n = 3, by = "lift")

1. Визуализируем эти правила с помощью графов:

plot(rules.filtrated, method = "graph")



1. В процессе анализа данных стало ясно, что при увиличении количества элементов в правиле значение параметров Support и Confidence уменьшается. Это связано с тем, что элементы вместе встречаюся реже, чем сами по себе.

Создадим обучающий набор из первых 8000 транзакций и тестовый набор на последних 2000:

Associations.training <- head(Associations, n = 8000)  
Associations.testing <- tail(Associations, n = 2000)

Запустим алгоритм на каждом датасете:

1. Обучающий набор:

rules.training <- apriori(data = Associations.training, parameter = list(support = 0.01, confidence = 0, minlen = 2))

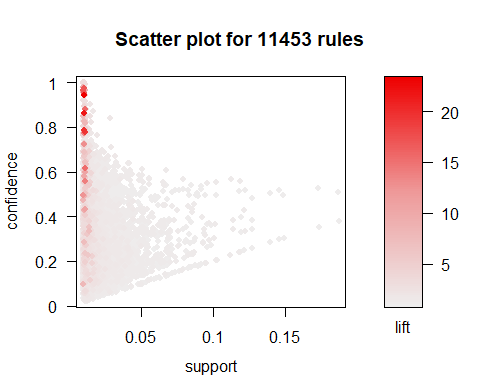
## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0 0.1 1 none FALSE TRUE 5 0.01 2  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 80   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[98 item(s), 8000 transaction(s)] done [0.00s].  
## sorting and recoding items ... [89 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 done [0.02s].  
## writing ... [11453 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(rules.training)

## set of 11453 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5   
## 2940 7179 1324 10   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.000 2.000 3.000 2.861 3.000 5.000   
##   
## summary of quality measures:  
## support confidence coverage lift   
## Min. :0.01000 Min. :0.02081 Min. :0.01000 Min. : 0.6486   
## 1st Qu.:0.01150 1st Qu.:0.17057 1st Qu.:0.03900 1st Qu.: 1.0133   
## Median :0.01400 Median :0.25150 Median :0.06275 Median : 1.1325   
## Mean :0.01819 Mean :0.28287 Mean :0.08597 Mean : 1.2520   
## 3rd Qu.:0.01975 3rd Qu.:0.36638 3rd Qu.:0.10287 3rd Qu.: 1.2982   
## Max. :0.18713 Max. :1.00000 Max. :0.49250 Max. :23.3109   
## count   
## Min. : 80.0   
## 1st Qu.: 92.0   
## Median : 112.0   
## Mean : 145.6   
## 3rd Qu.: 158.0   
## Max. :1497.0   
##   
## mining info:  
## data ntransactions support confidence  
## Associations.training 8000 0.01 0

plot(rules.training)

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



1. Тестовый набор:

rules.testing <- apriori(data = Associations.testing, parameter = list(support = 0.01, confidence = 0, minlen = 2))

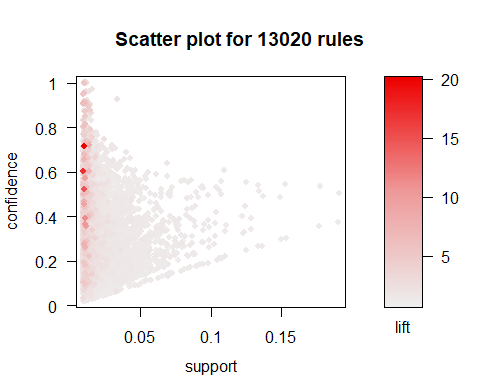
## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0 0.1 1 none FALSE TRUE 5 0.01 2  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 20   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[98 item(s), 2000 transaction(s)] done [0.00s].  
## sorting and recoding items ... [89 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 done [0.01s].  
## writing ... [13020 rule(s)] done [0.00s].  
## creating S4 object ... done [0.01s].

summary(rules.testing)

## set of 13020 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5   
## 3026 8154 1820 20   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 3.00 3.00 2.91 3.00 5.00   
##   
## summary of quality measures:  
## support confidence coverage lift   
## Min. :0.01000 Min. :0.01984 Min. :0.01000 Min. : 0.5762   
## 1st Qu.:0.01100 1st Qu.:0.17647 1st Qu.:0.03600 1st Qu.: 1.0200   
## Median :0.01350 Median :0.26316 Median :0.05650 Median : 1.1759   
## Mean :0.01762 Mean :0.29422 Mean :0.08124 Mean : 1.2919   
## 3rd Qu.:0.01900 3rd Qu.:0.38114 3rd Qu.:0.09850 3rd Qu.: 1.3957   
## Max. :0.19000 Max. :1.00000 Max. :0.50400 Max. :20.1207   
## count   
## Min. : 20.00   
## 1st Qu.: 22.00   
## Median : 27.00   
## Mean : 35.25   
## 3rd Qu.: 38.00   
## Max. :380.00   
##   
## mining info:  
## data ntransactions support confidence  
## Associations.testing 2000 0.01 0

plot(rules.testing)

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



1. Краткий отчёт по получившимся правилам и диаграммы показали, что большинство правил, присуьствующих в тренировочном наборе есть и в тестовом.
2. Можно сделать вывод, что правила, которые работают для тестового набора, работают и для всего датасета.