Association Rules Groceries

Karim

2019-10-16

```
knitr::opts_chunk$set(echo = TRUE)
options(repos=structure(c(CRAN="http://cran.utstat.utoronto.ca/")))
```

R Markdown

```
install.packages("arules")
```

```
##
## The downloaded binary packages are in
##  /var/folders/l_/k6t9zw295sn46sr8s62l6cb80000gn/T//RtmpZ9d9ED/downloaded_packages
```

```
install.packages("arulesViz")
```

```
##
## The downloaded binary packages are in
##  /var/folders/l_/k6t9zw295sn46sr8s62l6cb80000gn/T//RtmpZ9d9ED/downloaded_packages
```

```
install.packages("RColorBrewer")
```

```
##
## The downloaded binary packages are in
##  /var/folders/l_/k6t9zw295sn46sr8s62l6cb80000gn/
T//RtmpZ9d9ED/downloaded_packages
```

```
library(arules)
```

```
## Warning: package 'arules' was built under R versio
n 3.5.2
```

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

```
## Warning: package 'Matrix' was built under R versio
n 3.5.2
```

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

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

```
library(arulesViz)
```

```
## Warning: package 'arulesViz' was built under R ver
sion 3.5.2
```

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

```
library(RColorBrewer)
```

```
data("Groceries")
summary(Groceries)
```

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns
soda
##             2513             1903             1809
1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10
11   12   13   14   15
## 2159 1643 1299 1005  855  645  545  438  350  246
182  117   78   77   55
##   16   17   18   19   20   21   22   23   24   26
27   28   29   32
##   46   29   14   14    9   11    4    6    1    1
1    1    3    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##        labels  level2           level1
## 1 frankfurter sausage meat and sausage
## 2     sausage sausage meat and sausage
## 3  liver loaf sausage meat and sausage
```

show the dimensions of the transactions object

```
print(dim(Groceries))
```

```
## [1] 9835  169
```

```
print(dim(Groceries)[1]) # 9835 market baskets for sh
opping trips
```

```
## [1] 9835
```

```
print(dim(Groceries)[2]) #169 initial store items
```

```
## [1] 169
```

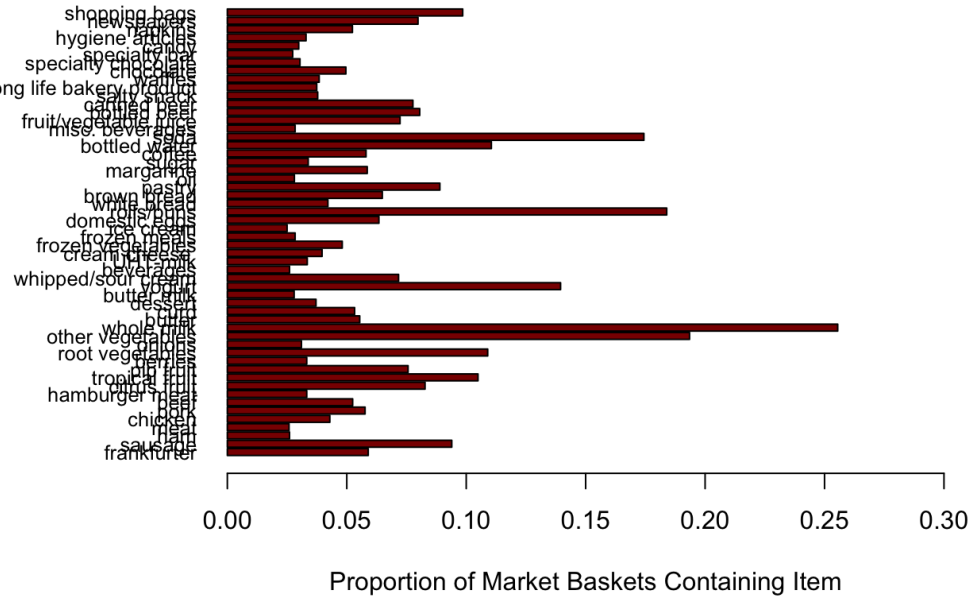
examine frequency for each item with support greater than 0.025

```
pdf(file="fig_market_basket_initial_item_support.pdf"
,width = 8.5, height = 11)
```

```
itemFrequencyPlot(Groceries, support = 0.025, cex.nam
es=0.8, xlim = c(0,0.3),

                  type = "relative", horiz = TRUE, c
ol = "dark red", las = 1,

                  xlab = "Proportion of Market Baske
ts Containing Item")
```




```
dev.off()
```

```
## pdf
##   3
```

explore possibilities for combining similar items

```
print(head(itemInfo(Groceries)))
```

```
##               labels  level2           level1
## 1       frankfurter sausage meat and sausage
## 2           sausage sausage meat and sausage
## 3        liver loaf sausage meat and sausage
## 4               ham sausage meat and sausage
## 5              meat sausage meat and sausage
## 6 finished products sausage meat and sausage
```

```
print(levels(itemInfo(Groceries)[["level1"]])) # 10 l
evels... too few"
```

```
##  [1] "canned food"          "detergent"
"drinks"
##  [4] "fresh products"       "fruit and vegetables"
"meat and sausage"
##  [7] "non-food"             "perfumery"
"processed food"
## [10] "snacks and candies"
```

```
print(levels(itemInfo(Groceries)[["level2"]])) # 55 d
istinct levels
```

```
##  [1] "baby food"                           "bags"
##  [3] "bakery improver"                     "bathroom c
leaner"
##  [5] "beef"                                "beer"
##  [7] "bread and backed goods"              "candy"
##  [9] "canned fish"                         "canned fru
it/vegetables"
## [11] "cheese"                              "chewing gu
m"
## [13] "chocolate"                           "cleaner"
## [15] "coffee"                              "condiment
s"
## [17] "cosmetics"                           "dairy prod
uce"
## [19] "delicatessen"                        "dental car
e"
## [21] "detergent/softener"                  "eggs"
## [23] "fish"                                "frozen foo
ds"
## [25] "fruit"                               "games/book
s/hobby"
## [27] "garden"                              "hair care"
## [29] "hard drinks"                         "health foo
d"
## [31] "jam/sweet spreads"                   "long-life
bakery products"
## [33] "meat spreads"                        "non-alc. d
rinks"
## [35] "non-food house keeping products" "non-food k
itchen"
## [37] "packaged fruit/vegetables"           "perfumery"
## [39] "personal hygiene"                    "pet food/c
are"
## [41] "pork"                                "poultry"
## [43] "pudding powder"                      "sausage"
## [45] "seasonal products"                   "shelf-stab
le dairy"
## [47] "snacks"                              "soap"
## [49] "soups/sauces"                        "staple foo
ds"
```

```
## [51] "sweetener"                          "tea/cocoa
drinks"
## [53] "vegetables"                         "vinegar/oi
ls"
## [55] "wine"
```

aggregate items using the 55 level2 levels for food categories to create a more meaningful set of items

```
groceries <- aggregate(Groceries, itemInfo(Groceries)
[["level2"]])
```

```
print(dim(groceries)[1]) # 9835 market baskets for sh
opping trips
```

```
## [1] 9835
```

```
print(dim(groceries)[2]) # 55 final store items (cate
gories)
```

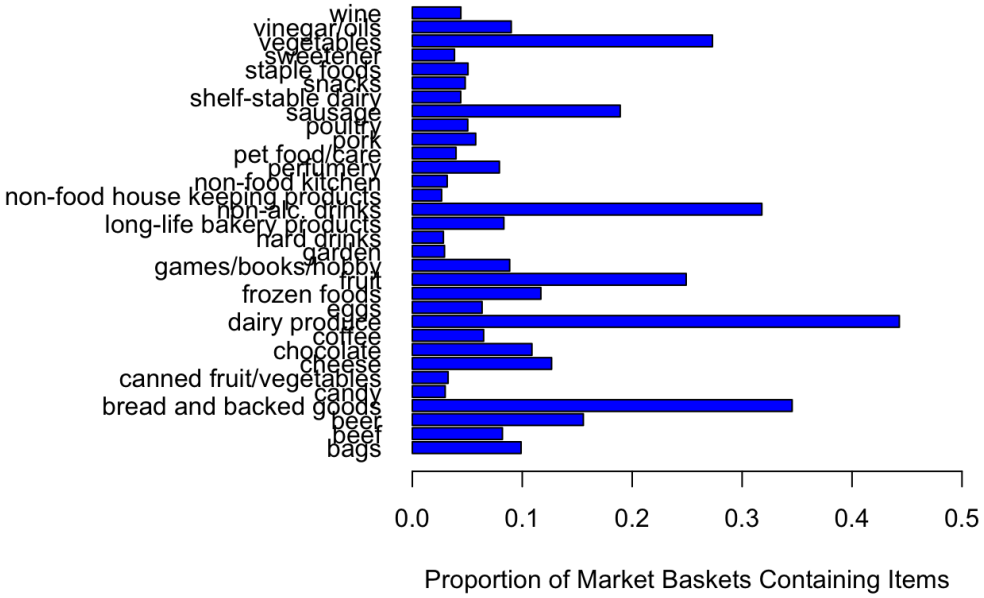
```
## [1] 55
```

```
pdf(file="fig_market_basket_initial_item_support.pdf"
,width = 8.5, height = 11)
```

```
itemFrequencyPlot(groceries, support = 0.025, cex.nam
es=1.0, xlim = c(0,0.5),

                  type = "relative", horiz = TRUE, c
ol = "blue", las = 1,

                  xlab = "Proportion of Market Baske
ts Containing Items")
```




```
dev.off()
```

```
## pdf
##   3
```

obtain large set of association rules for items by category and all shoppers this is done by setting very low criteria for support and confidence

```
first.rules <- apriori(groceries, parameter = list(su
pport = 0.001, confidence = 0.05))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport
maxtime support minlen
##         0.05    0.1    1 none FALSE            TRUE
5   0.001      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[55 item(s), 9835 transaction
(s)] done [0.00s].
## sorting and recoding items ... [54 item(s)] done
[0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.0
1s].
## writing ... [69921 rule(s)] done [0.01s].
## creating S4 object  ... done [0.02s].
```

```
print(summary(first.rules)) # yields 69,921 rules...
 too many
```

```
## set of 69921 rules
##
## rule length distribution (lhs + rhs):sizes
##     1     2     3     4     5     6     7     8
##    21  1205 10467 23895 22560  9888  1813    72
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   4.502   5.000   8.000
##
## summary of quality measures:
##     support           confidence          lift
count
##  Min.   :0.001017   Min.   :0.0500   Min.   : 0.44
75   Min.   :  10.00
##  1st Qu.:0.001118   1st Qu.:0.2110   1st Qu.: 1.83
15   1st Qu.:  11.00
##  Median :0.001525   Median :0.4231   Median : 2.25
73   Median :  15.00
##  Mean   :0.002488   Mean   :0.4364   Mean   : 2.53
82   Mean   :  24.47
##  3rd Qu.:0.002339   3rd Qu.:0.6269   3rd Qu.: 2.96
62   3rd Qu.:  23.00
##  Max.   :0.443010   Max.   :1.0000   Max.   :16.17
60   Max.   :4357.00
##
## mining info:
##       data ntransactions support confidence
##  groceries          9835   0.001       0.05
```

select association rules using thresholds for support and confidence

```
second.rules <- apriori(groceries, parameter = list(s
upport = 0.025, confidence = 0.05))
```

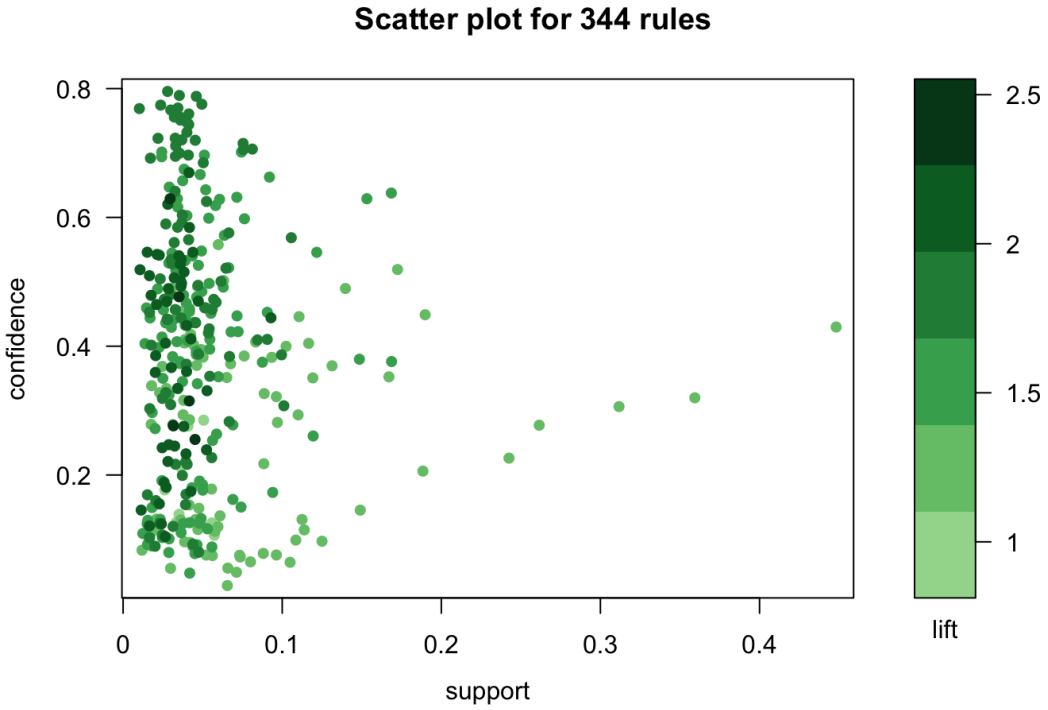
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport
maxtime support minlen
##         0.05    0.1    1 none FALSE            TRUE
5   0.025      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 245
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[55 item(s), 9835 transaction
(s)] done [0.00s].
## sorting and recoding items ... [32 item(s)] done
[0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [344 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
print(summary(second.rules)) # yields 344 rules
```

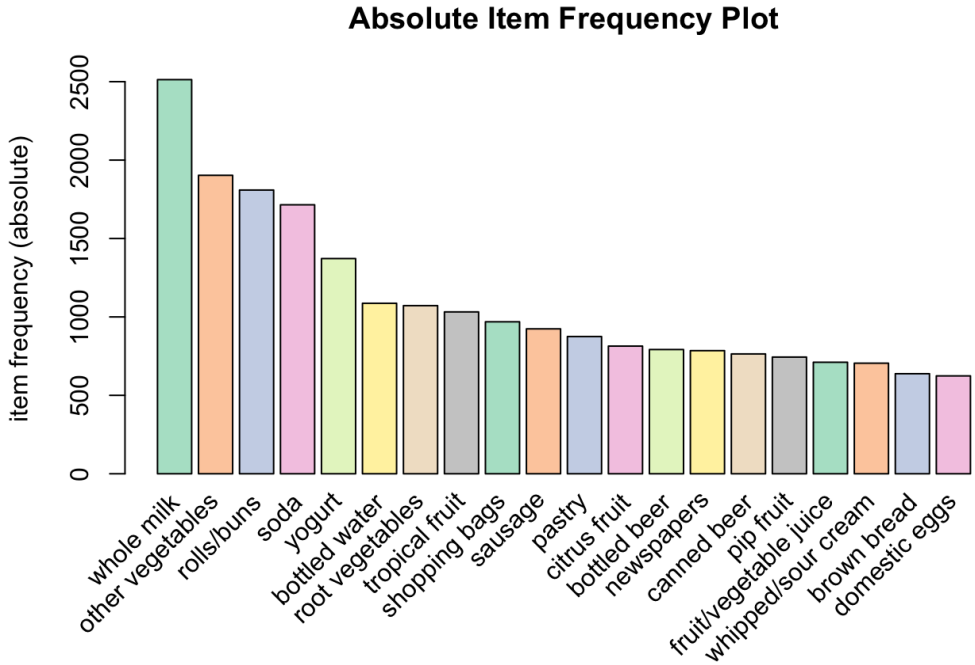
```
## set of 344 rules
##
## rule length distribution (lhs + rhs):sizes
##   1   2   3   4
##  21 162 129  32
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     2.0     2.0     2.5     3.0     4.0
##
## summary of quality measures:
##     support          confidence          lift
count
##  Min.   :0.02542   Min.   :0.05043   Min.   :0.666
9   Min.   : 250.0
##  1st Qu.:0.03030   1st Qu.:0.18202   1st Qu.:1.249
8   1st Qu.: 298.0
##  Median :0.03854   Median :0.39522   Median :1.477
0   Median : 379.0
##  Mean   :0.05276   Mean   :0.37658   Mean   :1.483
1   Mean   : 518.9
##  3rd Qu.:0.05236   3rd Qu.:0.51271   3rd Qu.:1.709
4   3rd Qu.: 515.0
##  Max.   :0.44301   Max.   :0.79841   Max.   :2.407
3   Max.   :4357.0
##
## mining info:
##       data ntransactions support confidence
##  groceries          9835   0.025       0.05
```

data visualization of association rules in scatter plot

```
plot(second.rules, control=list(jitter=2, col = rev(b
rewer.pal(9, "Greens")[4:9])), shading = "lift") # no
t working
```



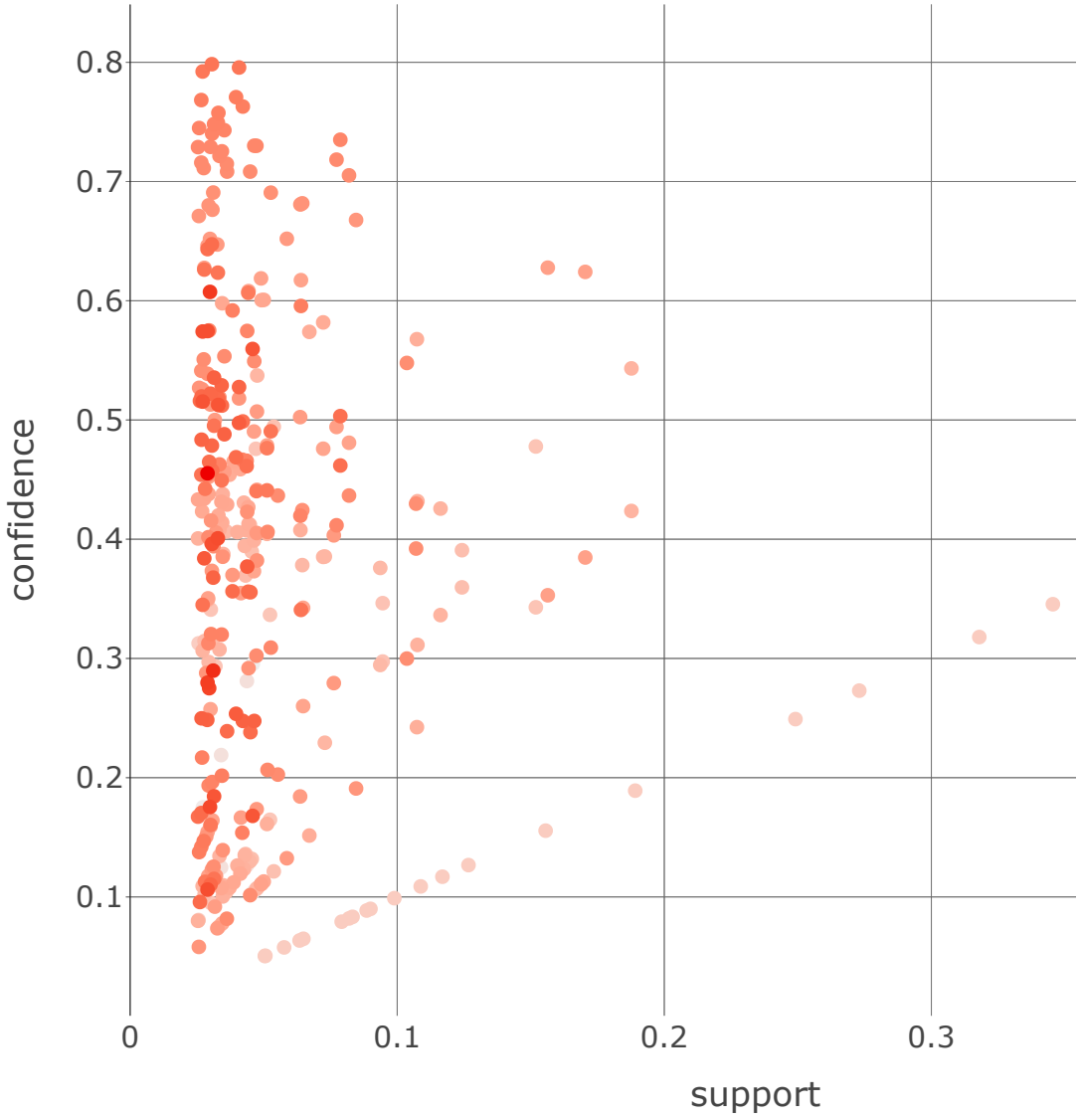

```
itemFrequencyPlot(Groceries,topN=20,type="absolute",col=brewer.pal(8,'Pastel2'), main="Absolute Item Frequency Plot")
```



```
plotly_arules(second.rules)
```

```
## Warning: 'plotly_arules' is deprecated.
## Use 'plot' instead.
## See help("Deprecated")
```

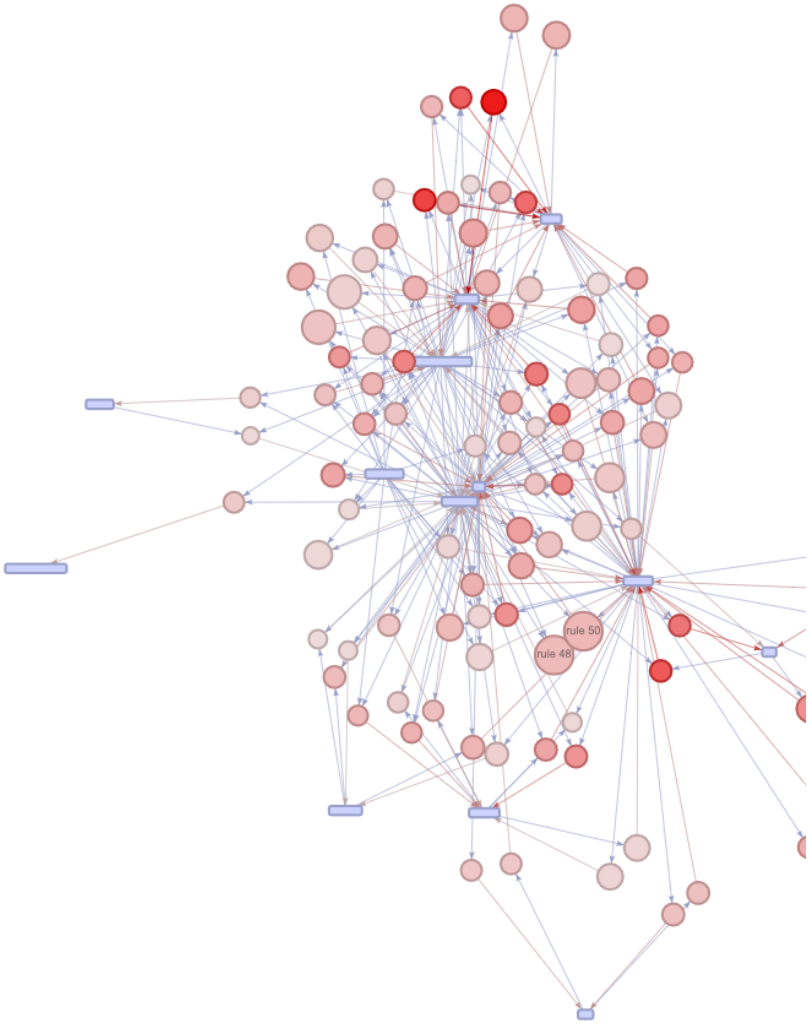
```
## To reduce overplotting, jitter is added! Use jitte
r = 0 to prevent jitter.
```



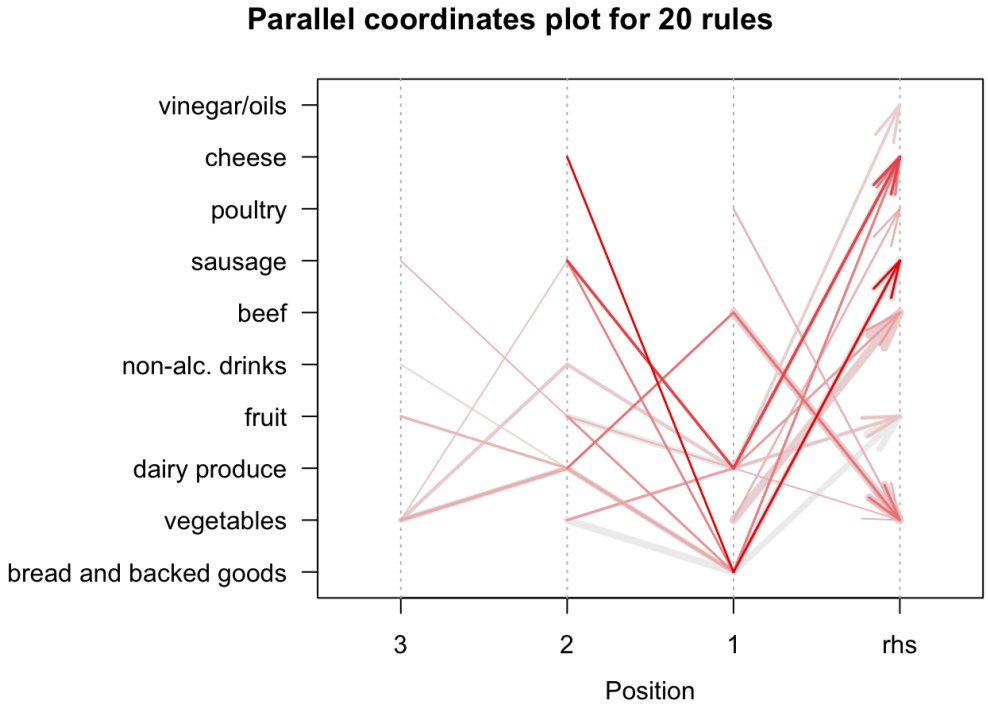
```
plot(second.rules, method = "graph",  engine = "htmlwidget")
```

```
## Warning: Too many rules supplied. Only plotting the best 100 rules using
## lift (change control parameter max if needed)
```

Select by id




```
second.rules2<-head(second.rules, n=20, by="lift")
plot(second.rules2, method="paracoord")
```


Parallel coordinates plot for 20 rules
vinegar/oils
cheese
poultry
sausage
beef
non-alc. drinks
fruit
dairy produce
vegetables
bread and backed goods
3
2
1
rhs
Position