ML\_Association\_Rules\_Exercise01

Daniel Ferreira Zanchetta and Lais Silva Almeida Zanchetta

18/02/2020

### Version de resumen de las Reglas de Asociación interesantes encontradad

1. {bottled beer,red/blush wine} => {liquor}
2. {tropical fruit,whole milk,yogurt} => {root vegetables}
3. {onions,whole milk} => {root vegetables}
4. {beef,rolls/buns} => {root vegetables}
5. {hamburger meat,yogurt,whipped/sour cream} => {butter}
6. {pip fruit,whole milk,yogurt,frozen meals} => {tropical fruit}
7. {root vegetables,whole milk,flour} => {whipped/sour cream}
8. {sausage} => {bread and backed goods}
9. {bread and backed goods,fruit} => {dairy produce}
10. {sausage*} => {bread and backed goods*}

### Versión más detallada de los pasos transcurridos

library (arules)

## Warning: package 'arules' was built under R version 3.6.2

## Loading required package: Matrix

##   
## Attaching package: 'arules'

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

library (arulesViz)

## Warning: package 'arulesViz' was built under R version 3.6.2

## Loading required package: grid

## Registered S3 method overwritten by 'seriation':  
## method from   
## reorder.hclust gclus

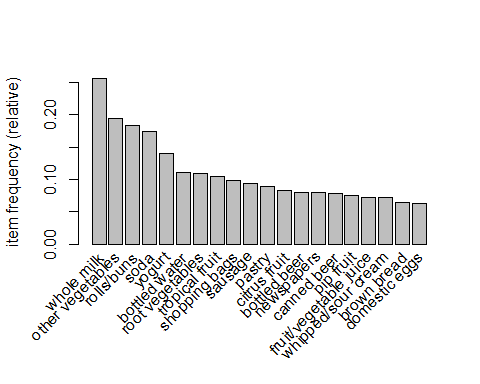
data ("Groceries")  
summary(Groceries)

## 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

Lo primero que hemos hecho despues de cargar la transaccion de Groceries, ha sido aplicar un Summary y alli descubrimos que los items que estaban siendo presentados con más frecuencia eran: -whole milk -other vegetables -rolls/buns -soda -yogurt

Para demostrar esta frecuancia de determinados items, hemos hecho un plot de frecuencia de los 20 items, que sale en orden descendente:

itemFrequencyPlot(Groceries,topN=20)



groceries.rules <- apriori(Groceries, parameter = list(support=0.005, confidence=0.3,minlen=2),control=list(verbose=F))  
groceries.rules.sortedlift <- sort(groceries.rules,by="lift")   
groceries.rules.sortedlift <- groceries.rules.sortedlift[!is.redundant(groceries.rules.sortedlift)]  
inspect(groceries.rules.sortedlift[1:10])

## lhs rhs support confidence lift count  
## [1] {citrus fruit,   
## other vegetables,   
## whole milk} => {root vegetables} 0.005795628 0.4453125 4.085493 57  
## [2] {herbs} => {root vegetables} 0.007015760 0.4312500 3.956477 69  
## [3] {citrus fruit,   
## pip fruit} => {tropical fruit} 0.005592272 0.4044118 3.854060 55  
## [4] {tropical fruit,   
## other vegetables,   
## whole milk} => {root vegetables} 0.007015760 0.4107143 3.768074 69  
## [5] {root vegetables,   
## whole milk,   
## yogurt} => {tropical fruit} 0.005693950 0.3916084 3.732043 56  
## [6] {pip fruit,   
## other vegetables,   
## whole milk} => {root vegetables} 0.005490595 0.4060150 3.724961 54  
## [7] {tropical fruit,   
## curd} => {yogurt} 0.005287239 0.5148515 3.690645 52  
## [8] {beef,   
## other vegetables} => {root vegetables} 0.007930859 0.4020619 3.688692 78  
## [9] {onions,   
## other vegetables} => {root vegetables} 0.005693950 0.4000000 3.669776 56  
## [10] {other vegetables,   
## whole milk,   
## fruit/vegetable juice} => {yogurt} 0.005083884 0.4854369 3.479790 50

En un primer intento para encontrar las reglas, hemos llegado a algunas reglas con un valor de lift bastante significativo, una que nos llamó la impresión por su lift de 35.71 ha sido de {bottled beer,red/blush wine} => {liquor}, con la cual añadimos a nuestra de reglas interesantes. Para llegar hasta ahi utilizamos support de 0.001 y confidence de 0.3. Sin embargo, en general, que no nos han interesado demasiado pues han parecido reglas un poco obvias. Mismo cuando cambiamos el support a 0.005. Un ejemplo en concreto es de las reglas [2] {herbs} => {root vegetables}. También nos ha dado la impresión que hay valores comunes que podrían ser quitados del algoritmo apriori (“other vegetables” y “shopping bags”), con la intención de intentar obtener reglas mas interesantes.

groceries.rules.filtered <- apriori(Groceries, parameter=list(support=0.001,confidence = 0.55,minlen=4,maxlen=5),appearance = list(none=c("other vegetables","shopping bags")),control=list(verbose=F))  
groceries.rules.filtered.sorted <- sort(groceries.rules.filtered,by="lift")  
groceries.rules.filtered.sorted <- groceries.rules.filtered.sorted[!is.redundant(groceries.rules.filtered.sorted)]  
inspect(groceries.rules.filtered.sorted[1:10])

## lhs rhs support confidence lift count  
## [1] {hamburger meat,   
## yogurt,   
## whipped/sour cream} => {butter} 0.001016777 0.6250000 11.278670 10  
## [2] {yogurt,   
## whipped/sour cream,   
## hard cheese} => {butter} 0.001016777 0.5882353 10.615219 10  
## [3] {tropical fruit,   
## whole milk,   
## yogurt,   
## sliced cheese} => {butter} 0.001016777 0.5555556 10.025484 10  
## [4] {whole milk,   
## butter,   
## hard cheese} => {whipped/sour cream} 0.001423488 0.6666667 9.300236 14  
## [5] {whole milk,   
## curd,   
## yogurt,   
## cream cheese } => {whipped/sour cream} 0.001118454 0.6470588 9.026700 11  
## [6] {citrus fruit,   
## whole milk,   
## cream cheese } => {domestic eggs} 0.001626843 0.5714286 9.006410 16  
## [7] {butter,   
## yogurt,   
## hard cheese} => {whipped/sour cream} 0.001016777 0.6250000 8.718972 10  
## [8] {curd,   
## yogurt,   
## sugar} => {whipped/sour cream} 0.001016777 0.6250000 8.718972 10  
## [9] {whole milk,   
## cream cheese ,   
## sugar} => {domestic eggs} 0.001118454 0.5500000 8.668670 11  
## [10] {citrus fruit,   
## tropical fruit,   
## grapes} => {fruit/vegetable juice} 0.001118454 0.6111111 8.453274 11

Por lo tanto, ha sido esto que hicimos, lo primero que intentamos ha sido jugar con un support de 0.001, confidence a 0.55, minimo de LHS a 4 y maximo a 5. Nos ha sorprendido la regla {hamburger meat,yogurt,whipped/sour cream} => {butter}, principalmente por su lift de 11.27.

Enseguida, cambiamos support a 0.005, confidence a 0.3, y minimo de LHS a 2. En el resultado ya empezaran a venir en repetidas combinaciones “whole milk” en los LHS, por lo tanto lo que nos parece un item más general. También hemos percibido RHS de “root vegetables” con un lift bastante elevado. Entonces, lo siguiente que hicimos ha sido centrarnos en los “root vegetables”:

root.vegetables.rule <- sort(subset(groceries.rules.filtered.sorted, rhs %in% "root vegetables"), by="lift")  
inspect(root.vegetables.rule)

## lhs rhs support confidence lift count  
## [1] {tropical fruit,   
## whole milk,   
## yogurt,   
## oil} => {root vegetables} 0.001118454 0.7857143 7.208489 11  
## [2] {whole milk,   
## yogurt,   
## rice} => {root vegetables} 0.001423488 0.7777778 7.135676 14  
## [3] {citrus fruit,   
## oil,   
## fruit/vegetable juice} => {root vegetables} 0.001016777 0.7692308 7.057262 10  
## [4] {sausage,   
## chicken,   
## citrus fruit} => {root vegetables} 0.001016777 0.7692308 7.057262 10  
## [5] {beef,   
## citrus fruit,   
## whipped/sour cream} => {root vegetables} 0.001016777 0.7692308 7.057262 10  
## [6] {sausage,   
## beef,   
## butter} => {root vegetables} 0.001016777 0.7142857 6.553172 10  
## [7] {citrus fruit,   
## tropical fruit,   
## frozen vegetables} => {root vegetables} 0.001118454 0.6875000 6.307428 11  
## [8] {citrus fruit,   
## herbs,   
## whole milk} => {root vegetables} 0.001321810 0.6842105 6.277249 13  
## [9] {sausage,   
## beef,   
## yogurt} => {root vegetables} 0.001321810 0.6842105 6.277249 13  
## [10] {whole milk,   
## whipped/sour cream,   
## flour} => {root vegetables} 0.001728521 0.6800000 6.238619 17  
## [11] {whole milk,   
## butter,   
## rice} => {root vegetables} 0.001016777 0.6666667 6.116294 10  
## [12] {herbs,   
## whole milk,   
## curd} => {root vegetables} 0.001220132 0.6666667 6.116294 12  
## [13] {pip fruit,   
## herbs,   
## whole milk} => {root vegetables} 0.001016777 0.6666667 6.116294 10  
## [14] {whole milk,   
## butter,   
## oil} => {root vegetables} 0.001220132 0.6666667 6.116294 12  
## [15] {pip fruit,   
## whole milk,   
## yogurt,   
## white bread} => {root vegetables} 0.001016777 0.6666667 6.116294 10  
## [16] {beef,   
## tropical fruit,   
## whole milk,   
## rolls/buns} => {root vegetables} 0.001423488 0.6666667 6.116294 14  
## [17] {tropical fruit,   
## herbs,   
## whole milk} => {root vegetables} 0.001525165 0.6521739 5.983331 15  
## [18] {beef,   
## whole milk,   
## butter,   
## rolls/buns} => {root vegetables} 0.001118454 0.6470588 5.936403 11  
## [19] {beef,   
## butter,   
## rolls/buns} => {root vegetables} 0.001423488 0.6363636 5.838280 14  
## [20] {yogurt,   
## whipped/sour cream,   
## sliced cheese} => {root vegetables} 0.001220132 0.6315789 5.794383 12  
## [21] {whole milk,   
## oil,   
## fruit/vegetable juice} => {root vegetables} 0.001220132 0.6315789 5.794383 12  
## [22] {sausage,   
## beef,   
## rolls/buns} => {root vegetables} 0.001220132 0.6315789 5.794383 12  
## [23] {beef,   
## tropical fruit,   
## rolls/buns} => {root vegetables} 0.001728521 0.6296296 5.776499 17  
## [24] {herbs,   
## whole milk,   
## butter} => {root vegetables} 0.001016777 0.6250000 5.734025 10  
## [25] {herbs,   
## whole milk,   
## rolls/buns} => {root vegetables} 0.001525165 0.6250000 5.734025 15  
## [26] {frankfurter,   
## whole milk,   
## sliced cheese} => {root vegetables} 0.001016777 0.6250000 5.734025 10  
## [27] {citrus fruit,   
## pip fruit,   
## frozen vegetables} => {root vegetables} 0.001016777 0.6250000 5.734025 10  
## [28] {sausage,   
## beef,   
## tropical fruit} => {root vegetables} 0.001016777 0.6250000 5.734025 10  
## [29] {whole milk,   
## butter,   
## whipped/sour cream,   
## rolls/buns} => {root vegetables} 0.001016777 0.6250000 5.734025 10  
## [30] {citrus fruit,   
## tropical fruit,   
## whole milk,   
## fruit/vegetable juice} => {root vegetables} 0.001016777 0.6250000 5.734025 10  
## [31] {beef,   
## yogurt,   
## cream cheese } => {root vegetables} 0.001118454 0.6111111 5.606602 11  
## [32] {citrus fruit,   
## whipped/sour cream,   
## frozen vegetables} => {root vegetables} 0.001118454 0.6111111 5.606602 11  
## [33] {whole milk,   
## whipped/sour cream,   
## soft cheese} => {root vegetables} 0.001220132 0.6000000 5.504664 12  
## [34] {tropical fruit,   
## whole milk,   
## oil} => {root vegetables} 0.001525165 0.6000000 5.504664 15  
## [35] {beef,   
## citrus fruit,   
## tropical fruit} => {root vegetables} 0.001220132 0.6000000 5.504664 12  
## [36] {pip fruit,   
## whole milk,   
## yogurt,   
## soda} => {root vegetables} 0.001220132 0.6000000 5.504664 12  
## [37] {herbs,   
## whole milk,   
## bottled water} => {root vegetables} 0.001016777 0.5882353 5.396730 10  
## [38] {frankfurter,   
## chicken,   
## whole milk} => {root vegetables} 0.001016777 0.5882353 5.396730 10  
## [39] {sausage,   
## chicken,   
## yogurt} => {root vegetables} 0.001016777 0.5882353 5.396730 10  
## [40] {butter,   
## whipped/sour cream,   
## napkins} => {root vegetables} 0.001016777 0.5882353 5.396730 10  
## [41] {tropical fruit,   
## yogurt,   
## oil} => {root vegetables} 0.001118454 0.5789474 5.311518 11  
## [42] {herbs,   
## whole milk,   
## yogurt} => {root vegetables} 0.001220132 0.5714286 5.242537 12  
## [43] {whole milk,   
## hard cheese,   
## domestic eggs} => {root vegetables} 0.001220132 0.5714286 5.242537 12  
## [44] {beef,   
## whole milk,   
## cream cheese } => {root vegetables} 0.001220132 0.5714286 5.242537 12  
## [45] {frankfurter,   
## tropical fruit,   
## whole milk,   
## yogurt} => {root vegetables} 0.001220132 0.5714286 5.242537 12  
## [46] {onions,   
## whole milk,   
## butter} => {root vegetables} 0.001728521 0.5666667 5.198850 17  
## [47] {sausage,   
## beef,   
## whole milk} => {root vegetables} 0.001728521 0.5666667 5.198850 17  
## [48] {beef,   
## citrus fruit,   
## whole milk} => {root vegetables} 0.002236909 0.5641026 5.175325 22  
## [49] {whole milk,   
## domestic eggs,   
## oil} => {root vegetables} 0.001016777 0.5555556 5.096911 10  
## [50] {frankfurter,   
## beef,   
## rolls/buns} => {root vegetables} 0.001016777 0.5555556 5.096911 10  
## [51] {beef,   
## butter,   
## yogurt} => {root vegetables} 0.001525165 0.5555556 5.096911 15  
## [52] {beef,   
## whole milk,   
## butter} => {root vegetables} 0.002033554 0.5555556 5.096911 20  
## [53] {beef,   
## tropical fruit,   
## whole milk} => {root vegetables} 0.002541942 0.5555556 5.096911 25  
## [54] {citrus fruit,   
## bottled water,   
## napkins} => {root vegetables} 0.001016777 0.5555556 5.096911 10  
## [55] {tropical fruit,   
## whole milk,   
## yogurt,   
## sliced cheese} => {root vegetables} 0.001016777 0.5555556 5.096911 10  
## [56] {pork,   
## citrus fruit,   
## whole milk} => {root vegetables} 0.001626843 0.5517241 5.061760 16  
## [57] {whole milk,   
## butter,   
## sliced cheese} => {root vegetables} 0.001118454 0.5500000 5.045942 11  
## [58] {citrus fruit,   
## tropical fruit,   
## pip fruit,   
## whole milk} => {root vegetables} 0.001118454 0.5500000 5.045942 11

Con el resultado seleccionamos algunas reglas que han parecido interesantes: {tropical fruit,whole milk,yogurt} => {root vegetables} –> esta parece involucrar muchos items de quien va para hacer compras para una familia {onions,whole milk} => {root vegetables} {beef,rolls/buns} => {root vegetables}

Lo siguiente que hicimos ha sido de continuar en el focalizar en algunos RHS que parecian interesantes, como “tropical fruit”:

tropical.fruit.rule <- sort(subset(groceries.rules.filtered.sorted, rhs %in% "tropical fruit"), by="lift")  
inspect(head(tropical.fruit.rule,5))

## lhs rhs support confidence lift count  
## [1] {citrus fruit,   
## grapes,   
## fruit/vegetable juice} => {tropical fruit} 0.001118454 0.8461538 8.063879 11  
## [2] {root vegetables,   
## whole milk,   
## yogurt,   
## oil} => {tropical fruit} 0.001118454 0.7333333 6.988695 11  
## [3] {pip fruit,   
## whole milk,   
## yogurt,   
## frozen meals} => {tropical fruit} 0.001016777 0.7142857 6.807171 10  
## [4] {whole milk,   
## butter,   
## yogurt,   
## sliced cheese} => {tropical fruit} 0.001016777 0.7142857 6.807171 10  
## [5] {onions,   
## butter,   
## yogurt} => {tropical fruit} 0.001118454 0.6875000 6.551902 11

Del resultado consideramos: {pip fruit,whole milk,yogurt,frozen meals} => {tropical fruit} Ha sido ahora la vez de mirar el RHS yogurt, y vemos por primera vez en los LHS “pork”, no obstante no es una regla que nos ha interesado de todo: {pork,tropical fruit,fruit/vegetable juice} => {yogurt}

yogurt.rule <- sort(subset(groceries.rules.filtered.sorted, rhs %in% "yogurt"), by="lift")  
inspect(head(yogurt.rule,10))

## lhs rhs support confidence lift count  
## [1] {root vegetables,   
## butter,   
## cream cheese } => {yogurt} 0.001016777 0.9090909 6.516698 10  
## [2] {tropical fruit,   
## whole milk,   
## butter,   
## sliced cheese} => {yogurt} 0.001016777 0.9090909 6.516698 10  
## [3] {sausage,   
## pip fruit,   
## sliced cheese} => {yogurt} 0.001220132 0.8571429 6.144315 12  
## [4] {tropical fruit,   
## whole milk,   
## butter,   
## curd} => {yogurt} 0.001220132 0.8571429 6.144315 12  
## [5] {tropical fruit,   
## butter,   
## white bread} => {yogurt} 0.001118454 0.8461538 6.065542 11  
## [6] {tropical fruit,   
## butter,   
## margarine} => {yogurt} 0.001118454 0.8461538 6.065542 11  
## [7] {whole milk,   
## curd,   
## whipped/sour cream,   
## cream cheese } => {yogurt} 0.001118454 0.8461538 6.065542 11  
## [8] {whipped/sour cream,   
## cream cheese ,   
## margarine} => {yogurt} 0.001016777 0.8333333 5.973639 10  
## [9] {beef,   
## tropical fruit,   
## butter} => {yogurt} 0.001016777 0.8333333 5.973639 10  
## [10] {pork,   
## tropical fruit,   
## fruit/vegetable juice} => {yogurt} 0.001016777 0.8333333 5.973639 10

root.vegetables.rule <- sort(subset(groceries.rules.filtered.sorted, lhs %in% "root vegetables" & lift > 3), by="lift")  
inspect(root.vegetables.rule[1:10])

## lhs rhs support confidence lift count  
## [1] {root vegetables,   
## whole milk,   
## flour} => {whipped/sour cream} 0.001728521 0.5862069 8.177794 17  
## [2] {root vegetables,   
## whole milk,   
## yogurt,   
## white bread} => {pip fruit} 0.001016777 0.5555556 7.343937 10  
## [3] {root vegetables,   
## whole milk,   
## yogurt,   
## oil} => {tropical fruit} 0.001118454 0.7333333 6.988695 11  
## [4] {root vegetables,   
## oil,   
## fruit/vegetable juice} => {citrus fruit} 0.001016777 0.5555556 6.712394 10  
## [5] {tropical fruit,   
## root vegetables,   
## whole milk,   
## fruit/vegetable juice} => {citrus fruit} 0.001016777 0.5555556 6.712394 10  
## [6] {root vegetables,   
## butter,   
## cream cheese } => {yogurt} 0.001016777 0.9090909 6.516698 10  
## [7] {root vegetables,   
## whole milk,   
## yogurt,   
## sliced cheese} => {tropical fruit} 0.001016777 0.6666667 6.353359 10  
## [8] {root vegetables,   
## whole milk,   
## yogurt,   
## bottled water} => {tropical fruit} 0.001525165 0.6521739 6.215243 15  
## [9] {tropical fruit,   
## root vegetables,   
## whole milk,   
## margarine} => {yogurt} 0.001016777 0.8333333 5.973639 10  
## [10] {root vegetables,   
## yogurt,   
## whipped/sour cream,   
## rolls/buns} => {tropical fruit} 0.001118454 0.6111111 5.823913 11

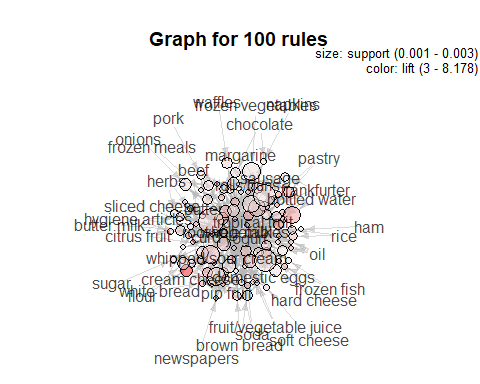
El siguiente que hicimos ha sido de regenerar reglas, pero ahora considerando un LHS que habia salido en otras reglas: “root vegetables”. Quedamos con esta regla de interesante: {root vegetables,whole milk,flour} => {whipped/sour cream}.

Abajo analizamos algunos graficos con algunos items que vimos durante la analisis.

#plot for root vegetables  
plot(root.vegetables.rule,method="graph",interactive=FALSE,shading="lift")

## Warning in plot.rules(root.vegetables.rule, method = "graph", interactive =  
## FALSE, : The parameter interactive is deprecated. Use engine='interactive'  
## instead.

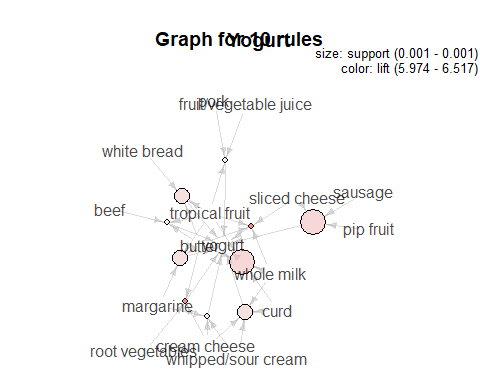
## Warning: plot: Too many rules supplied. Only plotting the best 100 rules  
## using 'support' (change control parameter max if needed)



#plot for yogurt - otra manera de hacer el plot  
plot(yogurt.rule[1:10],method="graph",interactive=FALSE,shading="lift")

## Warning in plot.rules(yogurt.rule[1:10], method = "graph", interactive =  
## FALSE, : The parameter interactive is deprecated. Use engine='interactive'  
## instead.

title(main = "Yogurt")



También hemos agregado los items de groceries por level2, que nos es dado con grupos como por ejemplo: -sausage -vegetables -poultry -beef -fruit -fish -cheese -dairy produce

#Aggregating by level2, found in itemInfo  
#itemInfo(Groceries)  
groceries\_level\_2 <- aggregate(Groceries, by="level2")  
groceries.rules.level2 <- apriori(groceries\_level\_2,parameter=list(support=0.07, confidence=0.5,minlen=2))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.5 0.1 1 none FALSE TRUE 5 0.07 2  
## 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: 688   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[55 item(s), 9835 transaction(s)] done [0.02s].  
## sorting and recoding items ... [16 item(s)] done [0.00s].  
## creating transaction tree ... done [0.01s].  
## checking subsets of size 1 2 3 done [0.00s].  
## writing ... [11 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

inspect(groceries.rules.level2)

## lhs rhs support confidence lift count  
## [1] {cheese} => {dairy produce} 0.08459583 0.6677368 1.507274 832  
## [2] {sausage} => {bread and backed goods} 0.10360956 0.5478495 1.585668 1019  
## [3] {sausage} => {dairy produce} 0.10737163 0.5677419 1.281557 1056  
## [4] {fruit} => {dairy produce} 0.15638027 0.6277551 1.417024 1538  
## [5] {vegetables} => {dairy produce} 0.17041179 0.6242086 1.409018 1676  
## [6] {bread and backed goods} => {dairy produce} 0.18769700 0.5432607 1.226295 1846  
## [7] {fruit,   
## vegetables} => {dairy produce} 0.07869853 0.7350427 1.659203 774  
## [8] {dairy produce,   
## fruit} => {vegetables} 0.07869853 0.5032510 1.843379 774  
## [9] {bread and backed goods,   
## fruit} => {dairy produce} 0.07727504 0.7183365 1.621492 760  
## [10] {bread and backed goods,   
## non-alc. drinks} => {dairy produce} 0.07229283 0.5818331 1.313364 711  
## [11] {bread and backed goods,   
## vegetables} => {dairy produce} 0.08195221 0.7051619 1.591753 806

groceries\_level\_3 <- addAggregate(Groceries, by="level2")  
groceries.rules.level3 <- apriori(groceries\_level\_3,parameter=list(support=0.1, confidence=0.3,minlen=2))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.3 0.1 1 none FALSE TRUE 5 0.1 2  
## 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: 983   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[224 item(s), 9835 transaction(s)] done [0.04s].  
## sorting and recoding items ... [18 item(s)] done [0.00s].  
## creating transaction tree ... done [0.02s].  
## checking subsets of size 1 2 3 done [0.01s].  
## writing ... [48 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

multirules <- filterAggregate(groceries.rules.level3)  
inspect(head(multirules, by = "lift"))

## lhs rhs support confidence  
## [1] {sausage\*} => {bread and backed goods\*} 0.1036096 0.5478495   
## [2] {fruit\*} => {vegetables\*} 0.1070666 0.4297959   
## [3] {vegetables\*} => {fruit\*} 0.1070666 0.3921788   
## [4] {whole milk} => {vegetables\*} 0.1062532 0.4158376   
## [5] {vegetables\*} => {whole milk} 0.1062532 0.3891993   
## [6] {other vegetables} => {dairy produce\*} 0.1220132 0.6305833   
## lift count  
## [1] 1.585668 1019   
## [2] 1.574318 1053   
## [3] 1.574318 1053   
## [4] 1.523189 1045   
## [5] 1.523189 1045   
## [6] 1.423408 1200

La regla {sausage} => {bread and backed goods} nos ha parecido interesante. Hemos notado que con un support de 0.07 y confidence de 0.5 los valores de lift van más proximos a 1 en relación a los valores que habiamos encontrados en las ejecuciones anteriores. Aun sobre los mismos parametros, la siguiente regla pueden ser de personas que les gusten a desayunos variados: {bread and backed goods,fruit} => {dairy produce}

Por fin, intentamos a través de la analisis Multi-level, para analizar la relacón de items individuales con items de grupo, y considerando un support de 0.1 y confidence de 0.3. La regla que nos ha parecido interesado ha sido la {sausage*} => {bread and backed goods*}.