Tema8_Ejercicio_reglas_asociacion

Fran Camacho

2025-03-13

Tema 8 - Ejercicio Reglas de Asociación

Utilizando el dataset IncomeESL incluido con la librería arules, se pide generar reglas de asociación.

Para ello, previamente deberá depurar el dataset. En particular:

- Revisar que no haya valores omitidos.
- Transformar los factores en valores numéricos. ← no es necesario!!!
- Una vez depurado el dataset, crear la matriz de transacciones usando la función transactions.

A la hora de ejecutar el algoritmo para obtener las reglas, no olvide establecer los valores de los parámetros de la función apriori, justificando el motivo de su elección.

Por último, elabore un breve informe resumiendo las reglas obtenidas y analizando su significado.

Paso 1: Carga de los datos

```r
if (!require(arules)) install.packages('arules', dependencies = T)

library(arules)

# to plot rules we use package arulesViz
if (!require(arulesViz)) {
  # package graph -and other packages too- must be installed first
(and it is not available anymore in CRAN)
  if (!require("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")
    BiocManager::install("Rgraphviz")
    BiocManager::install("graph")
  }
  install.packages('arulesViz', dependencies = T)
}

data("IncomeESL")
#data("Income")
```

Descripción de las variables del dataset:

https://rdrr.io/cran/arules/man/Income.html

income: an ordered factor with levels: [0,10) < [10,15) < [15,20) < [20,25) < [25,30) < [30,40) < [40,50) < [50,75) < 75+

sex: a factor with levels: male female

marital status: a factor with levels: married cohabitation divorced widowed single

age: an ordered factor with levels: 14-17 < 18-24 < 25-34 < 35-44 < 45-54 < 55-64 < 65+

education: an ordered factor with levels: grade <9 < grades 9-11 < high school graduate < college (1-3 years) < college graduate < graduate study

occupation: a factor with levels: professional/managerial sales laborer clerical/service homemaker student military retired unemployed

years in bay area: an ordered factor with levels: <1 < 1-3 < 4-6 < 7-10 < >10

dual incomes: a factor with levels: not married yes no

number in household: an ordered factor with levels: 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9+

number of children: an ordered factor with levels: 0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9+

householder status: a factor with levels: own, rent, live with parents/family

type of home: a factor with levels: house condominium apartment mobile Home other

ethnic classification: a factor with levels: american indian, asian black east indian hispanic pacific islander white other

language in home: a factor with levels: english spanish other

Paso 2: Explorar y preparar los datos

```r
income_raw <- IncomeESL

#structure
str(income_raw)
```

```
## 'data.frame':    8993 obs. of  14 variables:
##  $ income               : Ord.factor w/ 9 levels "[0,10)"<"[10,15)"<..: 9 9 9 1 1 8 1 6 2
4 ...
##  $ sex                  : Factor w/ 2 levels "male","female": 2 1 2 2 2 1 1 1 1 1 ...
##  $ marital status       : Factor w/ 5 levels "married","cohabitation",..: 1 1 1 5 5 1 5 3 1
1 ...
##  $ age                  : Ord.factor w/ 7 levels "14-17"<"18-24"<..: 5 5 3 1 1 6 2 3 6 7 ...
##  $ education            : Ord.factor w/ 6 levels "grade <9"<"grades 9-11"<..: 4 5 5 2 2 4 3 4
3 4 ...
##  $ occupation           : Factor w/ 9 levels "professional/managerial",..: 5 5 1 6 6 8 9 3 8 8
...
##  $ years in bay area    : Ord.factor w/ 5 levels "<1"<"1-3"<"4-6"<..: 5 5 5 5 3 5 4 5 5 4 ...
##  $ dual incomes         : Factor w/ 3 levels "not married",..: 3 3 2 1 1 3 1 1 3 3 ...
##  $ number in household  : Ord.factor w/ 9 levels "1"<"2"<"3"<"4"<..: 3 5 3 4 4 2 3 1 3 2 ...
##  $ number of children   : Ord.factor w/ 10 levels "0"<"1"<"2"<"3"<..: 1 3 2 3 3 1 2 1 1 1 ...
##  $ householder status   : Factor w/ 3 levels "own","rent","live with parents/family": 1 1 2 3
3 1 2 2 2 2 ...
##  $ type of home         : Factor w/ 5 levels "house","condominium",..: 1 1 3 1 1 1 3 3 3 3 ...
##  $ ethnic classification: Factor w/ 8 levels "american indian",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ language in home     : Factor w/ 3 levels "english","spanish",..: NA 1 1 1 1 1 1 1 1 1 ...
```

Nos quedamos con las observaciones que estén completas:

```r
# only complete observations
income_complete <- income_raw[complete.cases(income_raw), ]
```

Y obtenemos las transacciones con ellas:

```r
# get transactions
income_transac <- transactions(income_complete)
```

Estructura y sumario de las transacciones:

```r
#See the structure
str(income_transac)
```

```
## Formal class 'transactions' [package "arules"] with 3 slots
##   ..@ data       :Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
##   .. .. ..@ i       : int [1:96264] 8 9 11 20 27 33 42 45 50 57 ...
##   .. .. ..@ p       : int [1:6877] 0 14 28 42 56 70 84 98 112 126 ...
##   .. .. ..@ Dim     : int [1:2] 84 6876
##   .. .. ..@ Dimnames:List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : NULL
##   .. .. ..@ factors : list()
##   ..@ itemInfo   :'data.frame':  84 obs. of  3 variables:
##   .. ..$ labels   : chr [1:84] "income=[0,10)" "income=[10,15)" "income=[15,20)"
"income=[20,25)" ...
##   .. ..$ variables: Factor w/ 14 levels "age","dual incomes",..: 6 6 6 6 6 6 6 6 6 12 ...
##   .. ..$ levels   : Factor w/ 73 levels "[0,10)","[10,15)",..: 1 2 3 4 5 6 7 8 29 54 ...
##   ..@ itemsetInfo:'data.frame':  6876 obs. of  1 variable:
##   .. ..$ transactionID: chr [1:6876] "2" "3" "4" "5" ...
```

Resumen estadístico

```r
#See the structure
summary(income_transac)
```

```
## transactions as itemMatrix in sparse format with
##  6876 rows (elements/itemsets/transactions) and
##  84 columns (items) and a density of 0.1666667 
## 
## most frequent items:
##    language in home=english ethnic classification=white 
##                        6277                        4605 
##       years in bay area=>10        number of children=0 
##                        4446                        4276 
##    dual incomes=not married                     (Other) 
##                        4114                       72546 
## 
## element (itemset/transaction) length distribution:
## sizes
##   14 
## 6876 
## 
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      14      14      14      14      14      14 
## 
## includes extended item information - examples:
##           labels variables  levels
## 1  income=[0,10)    income  [0,10)
## 2 income=[10,15)    income [10,15)
## 3 income=[15,20)    income [15,20)
## 
## includes extended transaction information - examples:
##   transactionID
## 1             2
## 2             3
## 3             4
```

La matriz dispersa tiene 6876 filas y 84 columnas. Al no tratarse de un problema como el de la cesta de la compra, y al tener las filas originales todas 14 valores por haber eliminado las filas que tenían valores nulos, todas las transacciones tienen 14 valores. La densidad es el 16.67%. (No soy sociólogo, pero los items más frecuentes, dan una idea bastante clara de la población de la muestra).

Examinamos un par de transacciones con “inspect”:

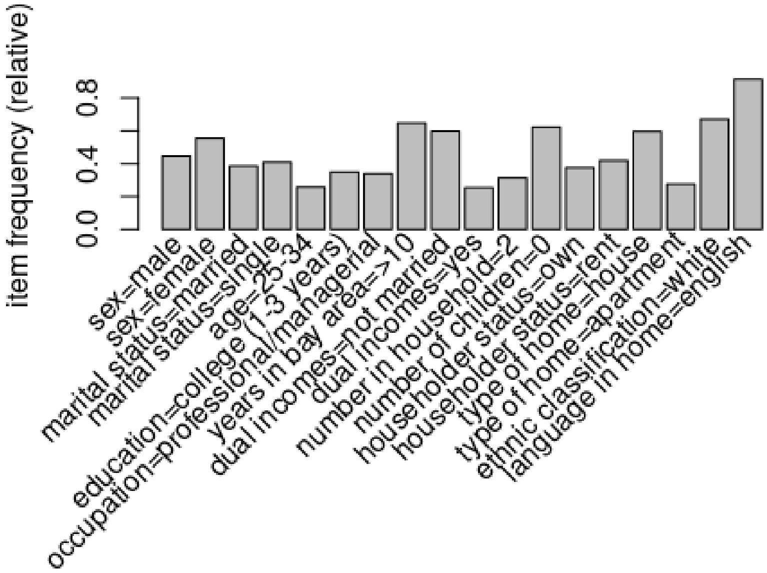
```
inspect(income_transac[1:2])
```

```
##     items                                 transactionID
## [1] {income=75+,                                       
##      sex=male,                                         
##      marital status=married,                           
##      age=45-54,                                        
##      education=college graduate,                       
##      occupation=homemaker,                             
##      years in bay area=>10,                            
##      dual incomes=no,                                  
##      number in household=5,                            
##      number of children=2,                             
##      householder status=own,                           
##      type of home=house,                               
##      ethnic classification=white,                      
##      language in home=english}                        2
## [2] {income=75+,
##      sex=female,
##      marital status=married,
##      age=25-34,
##      education=college graduate,
##      occupation=professional/managerial,
##      years in bay area=>10,
##      dual incomes=yes,
##      number in household=3,
##      number of children=1,
##      householder status=rent,
##      type of home=apartment,
##      ethnic classification=white,
##      language in home=english}                        3
```

Visualización de los datos

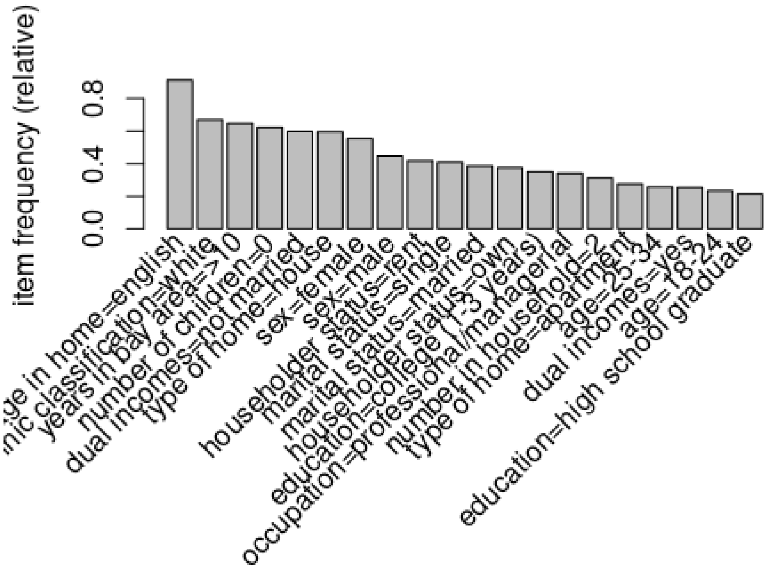
- Items con al menos un 25% de frecuencia

```
itemFrequencyPlot(income_transac, support = 0.25)    # at least 25
percent support
```



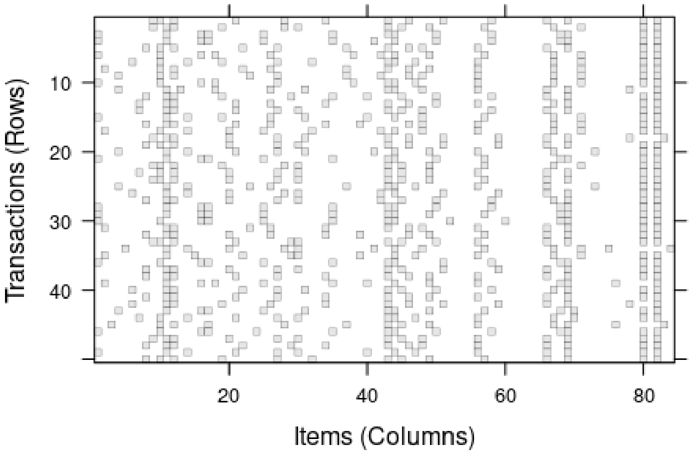
- Los 20 items más frecuentes

```
itemFrequencyPlot(income_transac, topN = 20) # top 20 items
```

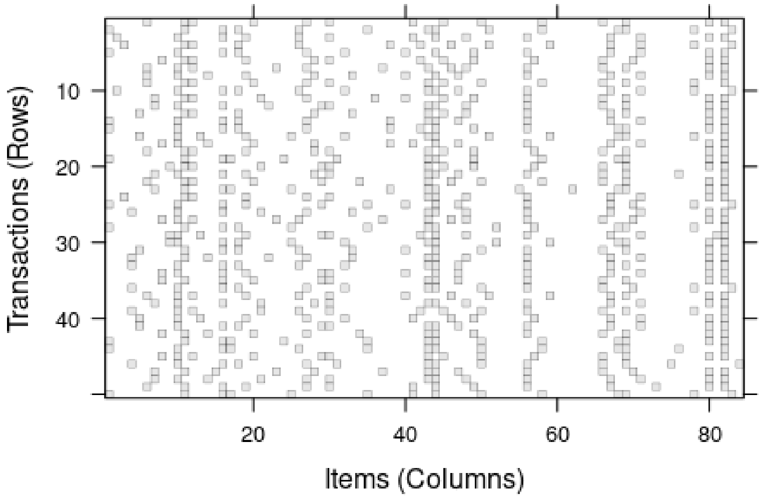


- Visualizamos también las matrices dispersas para algunos tramos de datos (por si vemos algo extraño)

```
image(income_transac[1:50])
```



```
#image(income_transac[1000:1050])
#image(income_transac[2000:2050])
#image(income_transac[5000:5050])
image(sample(income_transac, 50))
```



Se ve cómo predomina el idioma inglés. No veo nada más que sea digno de mención.

Paso 3: Entrenamiento del modelo

Vamos a probar primero con los valores por defecto:

```
system.time({
  income_rules <- apriori(income_transac)
})
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      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: 687
```

```
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[84 item(s), 6876 transaction(s)] done [0.01s].
## sorting and recoding items ... [42 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.02s].
## writing ... [1111 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

##    user  system elapsed
##   0.055   0.000   0.055
```

Con los parámetros por defecto (soporte 10%), se obtienen 1111 reglas.

Vamos a aumentar tanto el soporte como la confianza requeridas:

```
system.time({
income_rules <- apriori(income_transac, parameter = list(support =
0.5, 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     0.5      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: 3438
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[84 item(s), 6876 transaction(s)] done [0.01s].
## sorting and recoding items ... [7 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [12 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

##    user  system elapsed
##   0.017   0.000   0.017
```

Ahora obtenemos solo 12 reglas:

```
inspect(income_rules)
```

```
##      lhs                              rhs                              support confidence
coverage     lift count
## [1]  {sex=female}                  => {language in home=english}    0.5122164  0.9246521
0.5539558 1.012890  3522
## [2]  {language in home=english}    => {sex=female}                  0.5122164  0.5610961
0.9128854 1.012890  3522
## [3]  {type of home=house}          => {language in home=english}    0.5446481  0.9129693
0.5965678 1.000092  3745
```

```
## [4]  {language in home=english}    => {type of home=house}          0.5446481  0.5966226
0.9128854 1.000092  3745
## [5]  {dual incomes=not married}    => {language in home=english}    0.5426120  0.9069033
0.5983130 0.993447  3731
## [6]  {language in home=english}    => {dual incomes=not married}    0.5426120  0.5943922
0.9128854 0.993447  3731
## [7]  {number of children=0}        => {language in home=english}    0.5801338  0.9328812
0.6218732 1.021904  3989
## [8]  {language in home=english}    => {number of children=0}        0.5801338  0.6354947
0.9128854 1.021904  3989
## [9]  {years in bay area=>10}       => {language in home=english}    0.6013671  0.9300495
0.6465969 1.018802  4135
## [10] {language in home=english}    => {years in bay area=>10}       0.6013671  0.6587542
0.9128854 1.018802  4135
## [11] {ethnic classification=white} => {language in home=english}    0.6595404  0.9847991
0.6697208 1.078776  4535
## [12] {language in home=english}    => {ethnic classification=white} 0.6595404  0.7224789
0.9128854 1.078776  4535
```

Las 2 últimas reglas no aportan nada de información. Las 10 primeras, tampoco. (Considero todas triviales).

Vamos a buscar algo intermedio …

```
system.time({
  income_rules <- apriori(income_transac, parameter = list(support =
0.25, 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    0.25      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: 1719
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[84 item(s), 6876 transaction(s)] done [0.01s].
## sorting and recoding items ... [18 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [186 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##    user  system elapsed
##   0.016   0.000   0.016
```

```r
inspect(sort(income_rules, by = "lift")[1:20])
```

```
##      lhs                              rhs                           support confidence
coverage     lift count
## [1]  {type of home=apartment}      => {householder status=rent}  0.2507272  0.9097625
0.2755963 2.170551  1724
## [2]  {householder status=rent}     => {type of home=apartment}   0.2507272  0.5981957
0.4191390 2.170551  1724
## [3]  {householder status=own}      => {marital status=married}   0.2614892  0.6958204
0.3757999 1.804096  1798
## [4]  {marital status=married}      => {householder status=own}   0.2614892  0.6779789
0.3856894 1.804096  1798
## [5]  {marital status=single}       => {dual incomes=not married} 0.4091041  1.0000000
0.4091041 1.671366  2813
## [6]  {dual incomes=not married}    => {marital status=single}    0.4091041  0.6837628
0.5983130 1.671366  2813
## [7]  {marital status=single,
##       ethnic classification=white} => {dual incomes=not married} 0.2524724  1.0000000
0.2524724 1.671366  1736
## [8]  {marital status=single,
##       language in home=english}    => {dual incomes=not married} 0.3647469  1.0000000
0.3647469 1.671366  2508
## [9]  {dual incomes=not married,
##       language in home=english}    => {marital status=single}    0.3647469  0.6722058
0.5426120 1.643117  2508
## [10] {dual incomes=not married,
##       ethnic classification=white} => {marital status=single}    0.2524724  0.6608298
0.3820535 1.615310  1736
## [11] {number in household=2,
##       language in home=english}    => {number of children=0}     0.2725422  0.9186275
0.2966841 1.477194  1874
## [12] {number in household=2}       => {number of children=0}     0.2878127  0.9179035
0.3135544 1.476030  1979
## [13] {type of home=house,
##       language in home=english}    => {householder status=own}   0.2987202  0.5484646
0.5446481 1.459459  2054
## [14] {householder status=own,
##       language in home=english}    => {type of home=house}       0.2987202  0.8421484
0.3547120 1.411656  2054
## [15] {householder status=own}      => {type of home=house}       0.3161722  0.8413313
0.3757999 1.410286  2174
## [16] {type of home=house}          => {householder status=own}   0.3161722  0.5299854
0.5965678 1.410286  2174
## [17] {marital status=married,
##       language in home=english}    => {type of home=house}       0.2617801  0.7371007
0.3551483 1.235569  1800
## [18] {householder status=rent,
##       language in home=english}    => {number of children=0}     0.2920303  0.7608943
0.3837987 1.223552  2008
## [19] {marital status=married}      => {type of home=house}       0.2814136  0.7296380
0.3856894 1.223060  1935
## [20] {number of children=0,
##       language in home=english}    => {householder status=rent}  0.2920303  0.5033843
0.5801338 1.200996  2008
```

Creo que se pueden sacar conclusiones de estas reglas. Pero de todas maneras, considero que son demasiadas, y no veo el nivel de ingresos por ningún lado.

Voy a examinar con más detalle los ingresos. Voy a dividirlos en 3 niveles (bajo, medio, alto), y voy a intentar obtener reglas en las que estos 3 niveles aparezcan.

```r
# only complete observations
table(income_complete$income)
```

```
##
##  [0,10) [10,15) [15,20) [20,25) [25,30) [30,40) [40,50) [50,75)     75+
##    1255     529     505     618     527     846     784    1069     743
```

Vamos a dividirlos en tres rangos (bajos, medios, altos), y vamos a crear otra vez las transacciones con ese nuevo conjunto de datos.

```r
levels(income_complete[["income"]])
```

```
## [1] "[0,10)"  "[10,15)" "[15,20)" "[20,25)" "[25,30)" "[30,40)" "[40,50)"
## [8] "[50,75)" "75+"
```

```r
income_complete["income3levels"] <- income_complete["income"]
levels(income_complete[["income3levels"]]) <- c("0-20k$","0-20k$","0-20k$","20k-50k$","20k-50k$","20k-50k$","20k-50k$","50k+","50k+")
#test
income_complete[0:5,c("income","income3levels")]
```

```
##    income income3levels
## 2     75+          50k+
## 3     75+          50k+
## 4  [0,10)        0-20k$
## 5  [0,10)        0-20k$
## 6 [50,75)          50k+
```

```r
# remove income variable with 9 levels
income3L_complete <- income_complete[,-1]
```

Y obtenemos las transacciones con este dataframe

```r
# get transactions
income3L_transac <- transactions(income3L_complete)
```

Ahora aplicamos el algoritmo Apriori a estas nuevas transacciones:

```r
system.time({
income3L_rules <- apriori(income3L_transac, parameter = list(supp = 0.25, conf = 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    0.25      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: 1719
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[78 item(s), 6876 transaction(s)] done [0.00s].
## sorting and recoding items ... [21 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [199 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

##    user  system elapsed
##   0.018   0.000   0.018
```

Con el mismo nivel de soporte y confianza, obtenemos ahora 199 reglas, en lugar de 186. Presupongo que son las mismas o muy parecidas:

```
inspect(sort(income3L_rules, by = "lift")[1:20])
```

```
##      lhs                              rhs                           support confidence
coverage     lift count
## [1]  {type of home=apartment}      => {householder status=rent}  0.2507272  0.9097625
0.2755963 2.170551  1724
## [2]  {householder status=rent}     => {type of home=apartment}   0.2507272  0.5981957
0.4191390 2.170551  1724
## [3]  {householder status=own}      => {marital status=married}   0.2614892  0.6958204
0.3757999 1.804096  1798
## [4]  {marital status=married}      => {householder status=own}   0.2614892  0.6779789
0.3856894 1.804096  1798
## [5]  {marital status=single}       => {dual incomes=not married} 0.4091041  1.0000000
0.4091041 1.671366  2813
## [6]  {dual incomes=not married}    => {marital status=single}    0.4091041  0.6837628
0.5983130 1.671366  2813
## [7]  {marital status=single,

##       ethnic classification=white} => {dual incomes=not married} 0.2524724  1.0000000
0.2524724 1.671366  1736
## [8]  {marital status=single,

##       language in home=english}    => {dual incomes=not married} 0.3647469  1.0000000
0.3647469 1.671366  2508
## [9]  {dual incomes=not married,

##       language in home=english}    => {marital status=single}    0.3647469  0.6722058
0.5426120 1.643117  2508
## [10] {dual incomes=not married,

##       ethnic classification=white} => {marital status=single}    0.2524724  0.6608298
0.3820535 1.615310  1736
## [11] {number in household=2,

##       language in home=english}    => {number of children=0}     0.2725422  0.9186275
0.2966841 1.477194  1874
```

```
## [12] {number in household=2}       => {number of children=0}     0.2878127  0.9179035
0.3135544 1.476030  1979
## [13] {language in home=english,

##       income3levels=0-20k$}        => {dual incomes=not married} 0.2545084  0.8798391
0.2892670 1.470533  1750
## [14] {type of home=house,

##       language in home=english}    => {householder status=own}   0.2987202  0.5484646
0.5446481 1.459459  2054
## [15] {income3levels=0-20k$}        => {dual incomes=not married} 0.2882490  0.8658803
0.3328970 1.447203  1982
## [16] {householder status=own,

##       language in home=english}    => {type of home=house}       0.2987202  0.8421484
0.3547120 1.411656  2054
## [17] {householder status=own}      => {type of home=house}       0.3161722  0.8413313
0.3757999 1.410286  2174
## [18] {type of home=house}          => {householder status=own}   0.3161722  0.5299854
0.5965678 1.410286  2174
## [19] {marital status=married,

##       language in home=english}    => {type of home=house}       0.2617801  0.7371007
0.3551483 1.235569  1800
## [20] {householder status=rent,

##       language in home=english}    => {number of children=0}     0.2920303  0.7608943
0.3837987 1.223552  2008
```

Lo que se pretende es encontrar las reglas que en el lado derecho tengan alguno de los 3 niveles de ingresos.

Probamos sin especificar nada en el lado derecho:

```
system.time({                                                  # supp =
0.33, conf = 0.5 -> 0 rules
                                                               # supp =
0.2, conf = 0.8 -> 146
                                                               # upp =
0.1, conf = 0.5 -> 3676
  income3L_rules <- apriori(income3L_transac, parameter = list(supp =
0.2, conf = 0.8, minlen = 2))
})
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.2      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: 1375
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[78 item(s), 6876 transaction(s)] done [0.01s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.01s].
## writing ... [146 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

##    user  system elapsed
##   0.021   0.000   0.021

#inspect(sort(income3L_rules, by = "lift"))
```

Probamos con los niveles de ingreso en el lado derecho, y ajustamos los valores de soporte y confianza:

```
system.time({                                     # supp = 0.33, conf = 0.5 -> 0 rules
                                                 # supp = 0.2, conf = 0.8 -> 0
                                                 # supp = 0.1, conf = 0.5 -> 101
                                                 # supp = 0.1, conf = 0.6 -> 29
income3L_rules_RHS <- apriori(income3L_transac, parameter = list(supp
= 0.1, conf = 0.55, minlen = 2),
                             appearance = list(rhs=c("income3levels=0-
20k$", "income3levels=20k-50k$", "income3levels=50k+")))
})

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.55    0.1    1 none FALSE            TRUE       5     0.1      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: 687
##
## set item appearances ...[3 item(s)] done [0.00s].
## set transactions ...[78 item(s), 6876 transaction(s)] done [0.00s].
## sorting and recoding items ... [40 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.02s].
## writing ... [57 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

##    user  system elapsed
##   0.039   0.000   0.039
```

Vamos a examinar las 57 reglas obtenidas: (Estas son las reglas de asociación que estaba buscando).

```r
inspect(sort(income3L_rules_RHS, by = "lift"))
```

```
##      lhs                                                rhs                        support confidence
coverage     lift count
## [1]  {dual incomes=yes,

##       householder status=own}                        => {income3levels=50k+}     0.1023851  0.6622766
0.1545957 2.513142   704
## [2]  {marital status=married,

##       occupation=professional/managerial,

##       language in home=english}                      => {income3levels=50k+}     0.1012216  0.6590909
0.1535777 2.501054   696
## [3]  {occupation=professional/managerial,

##       householder status=own,

##       language in home=english}                      => {income3levels=50k+}     0.1090750  0.6590510
0.1655032 2.500902   750
## [4]  {occupation=professional/managerial,

##       householder status=own}                        => {income3levels=50k+}     0.1122746  0.6547922
0.1714660 2.484741   772
## [5]  {marital status=married,

##       occupation=professional/managerial}            => {income3levels=50k+}     0.1048575  0.6501353
0.1612856 2.467070   721
## [6]  {occupation=student,

##       dual incomes=not married,

##       language in home=english}                      => {income3levels=0-20k$}   0.1025305  0.7730263
0.1326353 2.322118   705
## [7]  {marital status=single,

##       occupation=student}                            => {income3levels=0-20k$}   0.1118383  0.7713139
0.1449971 2.316975   769
## [8]  {marital status=single,

##       occupation=student,

##       dual incomes=not married}                      => {income3levels=0-20k$}   0.1118383  0.7713139
0.1449971 2.316975   769
## [9]  {occupation=student,

##       dual incomes=not married}                      => {income3levels=0-20k$}   0.1182373  0.7698864
0.1535777 2.312686   813
## [10] {marital status=married,

##       householder status=own,

##       type of home=house,

##       ethnic classification=white,

##       language in home=english}                      => {income3levels=50k+}     0.1103839  0.6081731
0.1815009 2.307836   759
## [11] {marital status=married,

##       householder status=own,

##       type of home=house,

##       ethnic classification=white}                   => {income3levels=50k+}     0.1111111  0.6068308
0.1831006 2.302742   764
## [12] {marital status=married,

##       householder status=own,
##       ethnic classification=white,

##       language in home=english}                     => {income3levels=50k+}     0.1215823  0.5992832
0.2028796 2.274101   836
## [13] {marital status=married,

##       householder status=own,

##       ethnic classification=white}                  => {income3levels=50k+}     0.1224549  0.5975869
0.2049156 2.267664   842
## [14] {occupation=student,

##       language in home=english}                     => {income3levels=0-20k$}   0.1061664  0.7510288
0.1413613 2.256039   730
## [15] {occupation=student}                           => {income3levels=0-20k$}   0.1231821  0.7508865
0.1640489 2.255612   847
## [16] {marital status=married,

##       householder status=own,

##       type of home=house,

##       language in home=english}                     => {income3levels=50k+}     0.1307446  0.5922266
0.2207679 2.247324   899
## [17] {marital status=married,

##       householder status=own,

##       language in home=english}                     => {income3levels=50k+}     0.1445608  0.5847059
0.2472368 2.218785   994
## [18] {dual incomes=yes,

##       type of home=house}                           => {income3levels=50k+}     0.1009308  0.5827036
0.1732112 2.211187   694
## [19] {marital status=married,

##       householder status=own,

##       type of home=house}                           => {income3levels=50k+}     0.1351076  0.5799001
0.2329843 2.200548   929
## [20] {marital status=married,

##       householder status=own}                       => {income3levels=50k+}     0.1496510  0.5723026
0.2614892 2.171718  1029
## [21] {occupation=professional/managerial,

##       type of home=house,

##       language in home=english}                     => {income3levels=50k+}     0.1090750  0.5660377
0.1926992 2.147945   750
## [22] {marital status=married,

##       type of home=house,

##       ethnic classification=white,

##       language in home=english}                     => {income3levels=50k+}     0.1185282  0.5659722
0.2094241 2.147696   815
## [23] {marital status=married,

##       type of home=house,

##       ethnic classification=white}                  => {income3levels=50k+}     0.1194008  0.5654270
0.2111693 2.145627   821
## [24] {marital status=married,

##       years in bay area=>10,

##       householder status=own,

##       type of home=house}                           => {income3levels=50k+}     0.1022397  0.5597134
```

```
0.1826643 2.123945   703
## [25] {marital status=married,

##       years in bay area=>10,

##       householder status=own,

##       language in home=english}                      => {income3levels=50k+}     0.1070390  0.5579985
0.1918266 2.117438   736
## [26] {occupation=professional/managerial,

##       type of home=house}                            => {income3levels=50k+}     0.1128563  0.5578720
0.2022978 2.116958   776
## [27] {marital status=single,

##       householder status=live with parents/family}  => {income3levels=0-20k$}   0.1316172  0.6983025
0.1884817 2.097653   905
## [28] {marital status=single,

##       dual incomes=not married,

##       householder status=live with parents/family}  => {income3levels=0-20k$}   0.1316172  0.6983025
0.1884817 2.097653   905
## [29] {dual incomes=not married,

##       householder status=live with parents/family}  => {income3levels=0-20k$}   0.1394706  0.6979622
0.1998255 2.096631   959
## [30] {marital status=single,

##       householder status=live with parents/family,

##       language in home=english}                      => {income3levels=0-20k$}   0.1122746  0.6961226
0.1612856 2.091105   772
## [31] {marital status=single,

##       dual incomes=not married,

##       householder status=live with parents/family,

##       language in home=english}                      => {income3levels=0-20k$}   0.1122746  0.6961226
0.1612856 2.091105   772
## [32] {householder status=live with parents/family}  => {income3levels=0-20k$}   0.1423793  0.6943262
0.2050611 2.085709   979
## [33] {dual incomes=not married,

##       householder status=live with parents/family,

##       language in home=english}                      => {income3levels=0-20k$}   0.1185282  0.6942078
0.1707388 2.085353   815
## [34] {householder status=live with parents/family,

##       language in home=english}                      => {income3levels=0-20k$}   0.1201280  0.6889074
0.1743746 2.069431   826
## [35] {marital status=single,

##       householder status=live with parents/family,

##       type of home=house}                            => {income3levels=0-20k$}   0.1060209  0.6800373
0.1559046 2.042786   729
## [36] {marital status=single,

##       dual incomes=not married,

##       householder status=live with parents/family,

##       type of home=house}                            => {income3levels=0-20k$}   0.1060209  0.6800373
0.1559046 2.042786   729
## [37] {dual incomes=not married,

##       householder status=live with parents/family,
##       type of home=house}                           => {income3levels=0-20k$}   0.1119837  0.6790123
0.1649215 2.039707   770
## [38] {householder status=live with parents/family,

##       type of home=house}                           => {income3levels=0-20k$}   0.1140198  0.6746988
0.1689936 2.026749   784
## [39] {sex=female,

##       marital status=single}                        => {income3levels=0-20k$}   0.1237638  0.6184593
0.2001163 1.857810   851
## [40] {sex=female,

##       marital status=single,

##       dual incomes=not married}                     => {income3levels=0-20k$}   0.1237638  0.6184593
0.2001163 1.857810   851
## [41] {sex=female,

##       marital status=single,

##       language in home=english}                     => {income3levels=0-20k$}   0.1099476  0.6106624
0.1800465 1.834388   756
## [42] {sex=female,

##       marital status=single,

##       dual incomes=not married,

##       language in home=english}                     => {income3levels=0-20k$}   0.1099476  0.6106624
0.1800465 1.834388   756
## [43] {marital status=single,

##       type of home=house}                           => {income3levels=0-20k$}   0.1327807  0.5830140
0.2277487 1.751334   913
## [44] {marital status=single,

##       dual incomes=not married,

##       type of home=house}                           => {income3levels=0-20k$}   0.1327807  0.5830140
0.2277487 1.751334   913
## [45] {marital status=single,

##       type of home=house,

##       language in home=english}                     => {income3levels=0-20k$}   0.1163467  0.5813953
0.2001163 1.746472   800
## [46] {marital status=single,

##       dual incomes=not married,

##       type of home=house,

##       language in home=english}                     => {income3levels=0-20k$}   0.1163467  0.5813953
0.2001163 1.746472   800
## [47] {age=18-24,

##       dual incomes=not married,

##       language in home=english}                     => {income3levels=0-20k$}   0.1022397  0.5738776
0.1781559 1.723889   703
## [48] {age=18-24,

##       dual incomes=not married}                     => {income3levels=0-20k$}   0.1156195  0.5731795
0.2017161 1.721792   795
## [49] {marital status=single,

##       age=18-24}                                    => {income3levels=0-20k$}   0.1015125  0.5670187
0.1790285 1.703285   698
## [50] {marital status=single,

##       age=18-24,
##       dual incomes=not married}                     => {income3levels=0-20k$}   0.1015125  0.5670187
0.1790285 1.703285   698
## [51] {marital status=single}                        => {income3levels=0-20k$}   0.2268761  0.5545681
0.4091041 1.665885  1560
## [52] {marital status=single,

##       dual incomes=not married}                     => {income3levels=0-20k$}   0.2268761  0.5545681
0.4091041 1.665885  1560
## [53] {age=18-24}                                    => {income3levels=0-20k$}   0.1295812  0.5534161
0.2341478 1.662424   891
## [54] {age=18-24,

##       language in home=english}                     => {income3levels=0-20k$}   0.1135835  0.5503876
0.2063700 1.653327   781
## [55] {occupation=professional/managerial,

##       dual incomes=not married}                     => {income3levels=20k-50k$} 0.1016579  0.5964164
0.1704479 1.477823   699
## [56] {age=25-34,

##       householder status=rent}                      => {income3levels=20k-50k$} 0.1009308  0.5866441
0.1720477 1.453609   694
## [57] {sex=male,

##       householder status=rent,

##       language in home=english}                     => {income3levels=20k-50k$} 0.1018034  0.5524862
0.1842641 1.368971   700
```

Vamos a intentar encontrar ahora las reglas redundantes (con la función **is.subset()**), eliminarlas y simplicar así el conjunto de reglas:

```
income3L_rules_RHS_sortedByLift <- sort(income3L_rules_RHS, by =
"lift")
#inspect(income3L_rules_RHS_sortedByLift)

subset_matrix <-
is.subse
t(income3L_rules_RHS_sortedByLift,income3L_rules_RHS_sortedByLift)
#subset_matrix
```

[

57 x 57 sparse Matrix of class “ngCMatrix” [[suppressing 57 column names ‘{dual incomes=yes,householder status=own,income3levels=50k+}’, ‘{marital status=married,occupation=professional/managerial,language in home=english,income3levels=50k+}’, ‘{occupation=professional/managerial,householder status=own,language in home=english,income3levels=50k+}’ …]] [[suppressing 57 column names ‘{dual incomes=yes,householder status=own,income3levels=50k+}’, ‘{marital status=married,occupation=professional/managerial,language in home=english,income3levels=50k+}’, ‘{occupation=professional/managerial,householder status=own,language in home=english,income3levels=50k+}’ …]]

{dual incomes=yes,householder status=own,income3levels=50k+} | . . . . {marital status=married,occupation=professional/managerial,language in home=english,income3levels=50k+} . | . . . . {occupation=professional/managerial,householder status=own,language in home=english,income3levels=50k+} . . | . . . . {occupation=professional/managerial,householder status=own,income3levels=50k+} . . | | . . . . {marital status=married,occupation=professional/managerial,income3levels=50k+} . | . . | . . . . {occupation=student,dual incomes=not married,language in home=english,income3levels=0-20k$} . . . . . | . . . . {marital status=single,occupation=student,income3levels=0-20k$} . . . . . . | | . . . .

. . . . {marital status=single,occupation=student,dual incomes=not married,income3levels=0-20k$} . . . . | . . . . {occupation=student,dual incomes=not married,income3levels=0-20k$} . . . . . | . | | . . . .

]

Eliminamos las ocurrencias que se producen en la diagonal principal y por debajo de ella:

```r
subset_matrix[lower.tri(subset_matrix, diag = T)] <- F
#subset_matrix
```

[

output:

57 x 57 sparse Matrix of class “ngCMatrix” [[suppressing 57 column names ‘{dual incomes=yes,householder status=own,income3levels=50k+}’, ‘{marital status=married,occupation=professional/managerial,language in home=english,income3levels=50k+}’, ‘{occupation=professional/managerial,householder status=own,language in home=english,income3levels=50k+}’ …]] [[suppressing 57 column names ‘{dual incomes=yes,householder status=own,income3levels=50k+}’, ‘{marital status=married,occupation=professional/managerial,language in home=english,income3levels=50k+}’, ‘{occupation=professional/managerial,householder status=own,language in home=english,income3levels=50k+}’ …]]

{dual incomes=yes,householder status=own,income3levels=50k+} . . . . {marital status=married,occupation=professional/managerial,language in home=english,income3levels=50k+} . . . . {occupation=professional/managerial,householder status=own,language in home=english,income3levels=50k+} . . . . {occupation=professional/managerial,householder status=own,income3levels=50k+} . . . . {marital status=married,occupation=professional/managerial,income3levels=50k+} . . . . {occupation=student,dual incomes=not married,language in home=english,income3levels=0-20k$} . . . . {marital status=single,occupation=student,income3levels=0-20k$} . . . . | . . . . {marital status=single,occupation=student,dual incomes=not married,income3levels=0-20k$} . . . . {occupation=student,dual incomes=not married,income3levels=0-20k$} . . . .

]

Ahora podemos extraer las reglas de asociación redundantes, que serán aquellas que tengan una columna cuya suma sea igual o superior a 1:

```r
redundant <- colSums(subset_matrix, na.rm = T) >= 1
#redundant
```

[

output:

```
{dual incomes=yes,householder status=own,income3levels=50k+}

FALSE
                                {marital
status=married,occupation=professional/managerial,language in home=english,income3levels=50k+}

FALSE
                                {occupation=professional/managerial,householder
status=own,language in home=english,income3levels=50k+}

FALSE

{occupation=professional/managerial,householder status=own,income3levels=50k+}

FALSE
                                                        {marital
status=married,occupation=professional/managerial,income3levels=50k+}

FALSE
                                           {occupation=student,dual incomes=not married,language
in home=english,income3levels=0-20k$}

FALSE
                                                                       {marital
status=single,occupation=student,income3levels=0-20k$}

FALSE
                                              {marital status=single,occupation=student,dual
incomes=not married,income3levels=0-20k$}

TRUE
                                                                    {occupation=student,dual
incomes=not married,income3levels=0-20k$}

FALSE
```

…

]

Las reglas marcadas como TRUE son consideradas redundantes por lo que podemos eliminarlas:

```
rules_pruned <- income3L_rules_RHS_sortedByLift[!redundant]
inspect(rules_pruned)
```

```
##      lhs                                              rhs                        support
confidence  coverage     lift count
## [1]  {dual incomes=yes,

##       householder status=own}                      => {income3levels=50k+}     0.1023851
0.6622766 0.1545957 2.513142   704
## [2]  {marital status=married,

##       occupation=professional/managerial,

##       language in home=english}                    => {income3levels=50k+}     0.1012216
0.6590909 0.1535777 2.501054   696
## [3]  {occupation=professional/managerial,

##       householder status=own,

##       language in home=english}                    => {income3levels=50k+}     0.1090750
0.6590510 0.1655032 2.500902   750
## [4]  {occupation=professional/managerial,

##       householder status=own}                      => {income3levels=50k+}     0.1122746
```

```
0.6547922 0.1714660 2.484741   772
## [5]  {marital status=married,

##       occupation=professional/managerial}          => {income3levels=50k+}     0.1048575
0.6501353 0.1612856 2.467070   721
## [6]  {occupation=student,

##       dual incomes=not married,

##       language in home=english}                    => {income3levels=0-20k$}   0.1025305
0.7730263 0.1326353 2.322118   705
## [7]  {marital status=single,

##       occupation=student}                          => {income3levels=0-20k$}   0.1118383
0.7713139 0.1449971 2.316975   769
## [8]  {occupation=student,

##       dual incomes=not married}                    => {income3levels=0-20k$}   0.1182373
0.7698864 0.1535777 2.312686   813
## [9]  {marital status=married,

##       householder status=own,

##       type of home=house,

##       ethnic classification=white,

##       language in home=english}                    => {income3levels=50k+}     0.1103839
0.6081731 0.1815009 2.307836   759
## [10] {marital status=married,

##       householder status=own,

##       type of home=house,

##       ethnic classification=white}                 => {income3levels=50k+}     0.1111111
0.6068308 0.1831006 2.302742   764
## [11] {marital status=married,

##       householder status=own,

##       ethnic classification=white,

##       language in home=english}                    => {income3levels=50k+}     0.1215823
0.5992832 0.2028796 2.274101   836
## [12] {marital status=married,

##       householder status=own,

##       ethnic classification=white}                 => {income3levels=50k+}     0.1224549
0.5975869 0.2049156 2.267664   842
## [13] {occupation=student,

##       language in home=english}                    => {income3levels=0-20k$}   0.1061664
0.7510288 0.1413613 2.256039   730
## [14] {occupation=student}                          => {income3levels=0-20k$}   0.1231821
0.7508865 0.1640489 2.255612   847
## [15] {marital status=married,

##       householder status=own,

##       type of home=house,
##       language in home=english}                     => {income3levels=50k+}     0.1307446
0.5922266 0.2207679 2.247324   899
## [16] {marital status=married,

##       householder status=own,

##       language in home=english}                     => {income3levels=50k+}     0.1445608
0.5847059 0.2472368 2.218785   994
## [17] {dual incomes=yes,

##       type of home=house}                           => {income3levels=50k+}     0.1009308
0.5827036 0.1732112 2.211187   694
## [18] {marital status=married,

##       householder status=own,

##       type of home=house}                           => {income3levels=50k+}     0.1351076
0.5799001 0.2329843 2.200548   929
## [19] {marital status=married,

##       householder status=own}                       => {income3levels=50k+}     0.1496510
0.5723026 0.2614892 2.171718  1029
## [20] {occupation=professional/managerial,

##       type of home=house,

##       language in home=english}                     => {income3levels=50k+}     0.1090750
0.5660377 0.1926992 2.147945   750
## [21] {marital status=married,

##       type of home=house,

##       ethnic classification=white,

##       language in home=english}                     => {income3levels=50k+}     0.1185282
0.5659722 0.2094241 2.147696   815
## [22] {marital status=married,

##       type of home=house,

##       ethnic classification=white}                  => {income3levels=50k+}     0.1194008
0.5654270 0.2111693 2.145627   821
## [23] {occupation=professional/managerial,

##       type of home=house}                           => {income3levels=50k+}     0.1128563
0.5578720 0.2022978 2.116958   776
## [24] {marital status=single,

##       householder status=live with parents/family} => {income3levels=0-20k$}   0.1316172
0.6983025 0.1884817 2.097653   905
## [25] {dual incomes=not married,

##       householder status=live with parents/family} => {income3levels=0-20k$}   0.1394706
0.6979622 0.1998255 2.096631   959
## [26] {householder status=live with parents/family} => {income3levels=0-20k$}   0.1423793
0.6943262 0.2050611 2.085709   979
## [27] {sex=female,

##       marital status=single}                        => {income3levels=0-20k$}   0.1237638
0.6184593 0.2001163 1.857810   851
## [28] {marital status=single,
##       type of home=house}                             => {income3levels=0-20k$}   0.1327807
0.5830140 0.2277487 1.751334   913
## [29] {age=18-24,

##       dual incomes=not married,

##       language in home=english}                       => {income3levels=0-20k$}   0.1022397
0.5738776 0.1781559 1.723889   703
## [30] {age=18-24,

##       dual incomes=not married}                       => {income3levels=0-20k$}   0.1156195
0.5731795 0.2017161 1.721792   795
## [31] {marital status=single,

##       age=18-24}                                      => {income3levels=0-20k$}   0.1015125
0.5670187 0.1790285 1.703285   698
## [32] {marital status=single}                          => {income3levels=0-20k$}   0.2268761
0.5545681 0.4091041 1.665885  1560
## [33] {age=18-24}                                      => {income3levels=0-20k$}   0.1295812
0.5534161 0.2341478 1.662424   891
## [34] {occupation=professional/managerial,

##       dual incomes=not married}                       => {income3levels=20k-50k$} 0.1016579
0.5964164 0.1704479 1.477823   699
## [35] {age=25-34,

##       householder status=rent}                        => {income3levels=20k-50k$} 0.1009308
0.5866441 0.1720477 1.453609   694
## [36] {sex=male,

##       householder status=rent,

##       language in home=english}                       => {income3levels=20k-50k$} 0.1018034
0.5524862 0.1842641 1.368971   700
```

Nos quedamos al final con 36 reglas.

Las reglas con más elevación son las que incluyen los sueldos altos. Las que menos, las que incluyen los sueldos intermedios.

Las reglas de asociación de este grupo de población con sueldos intermedios son las 3 últimas (34,35 y 36). La 35 por ejemplo indica que una edad entre 25 y 34, si vive de alquiler, es porque probablemente tendrá un sueldo que le permite vivir de manera independiente (ni en el domicilio de los padres .. ni tampoco adquirir una propiedad).

La regla 31 indica que las personas solteras, y con una edad entre 18 y 24, tendrán probablemente un sueldo inferior a 20k $. (Igual que la regla 29 y 30 ..) Si la ocupación es estudiante, entonces la regla de asociación con un sueldo bajo tiene una elevación y una confianza todavía mayores (regla 14). (Y mayores todavía si la regla incluye en la parte izquierda que se está soltero y se es estudiante, regla 7).

En cuanto a los sueldos más altos: Si hay dos sueldos en la unidad familiar, y se tiene la propiedad de la vivienda (regla 1), eso indica que es bastante probable que se trate de sueldos altos. (También si la categoría profesional es alta, y se tiene en propiedad la casa -y más si se habla inglés-, reglas 2 y 3).

Métricas:

Se pueden generar diferentes métricas con la función interestMeasure() disponible en la librería arules.

```r
#interestMeasure(rules_pruned, c("support", "chiSquare", "confidence",
"conviction", "cosine", "coverage", "leverage", "lift", "oddsRatio"),
income3L_complete)
interestMeasure(rules_pruned[1:10], c("support",  "confidence",
"lift", "chiSquare",  "coverage"), income3L_complete)
```

```
##      support confidence     lift chiSquared  coverage
## 1  0.1023851  0.6622766 2.513142  1030.1318 0.1545957
## 2  0.1012216  0.6590909 2.501054  1005.8511 0.1535777
## 3  0.1090750  0.6590510 2.500902  1099.2254 0.1655032
## 4  0.1122746  0.6547922 2.484741  1122.4564 0.1714660
## 5  0.1048575  0.6501353 2.467070  1018.3179 0.1612856
## 6  0.1025305  0.7730263 2.322118   917.1720 0.1326353
## 7  0.1118383  0.7713139 2.316975  1009.2514 0.1449971
## 8  0.1182373  0.7698864 2.312686  1072.7924 0.1535777
## 9  0.1103839  0.6081731 2.307836   933.1837 0.1815009
## 10 0.1111111  0.6068308 2.302742   935.9197 0.1831006
```

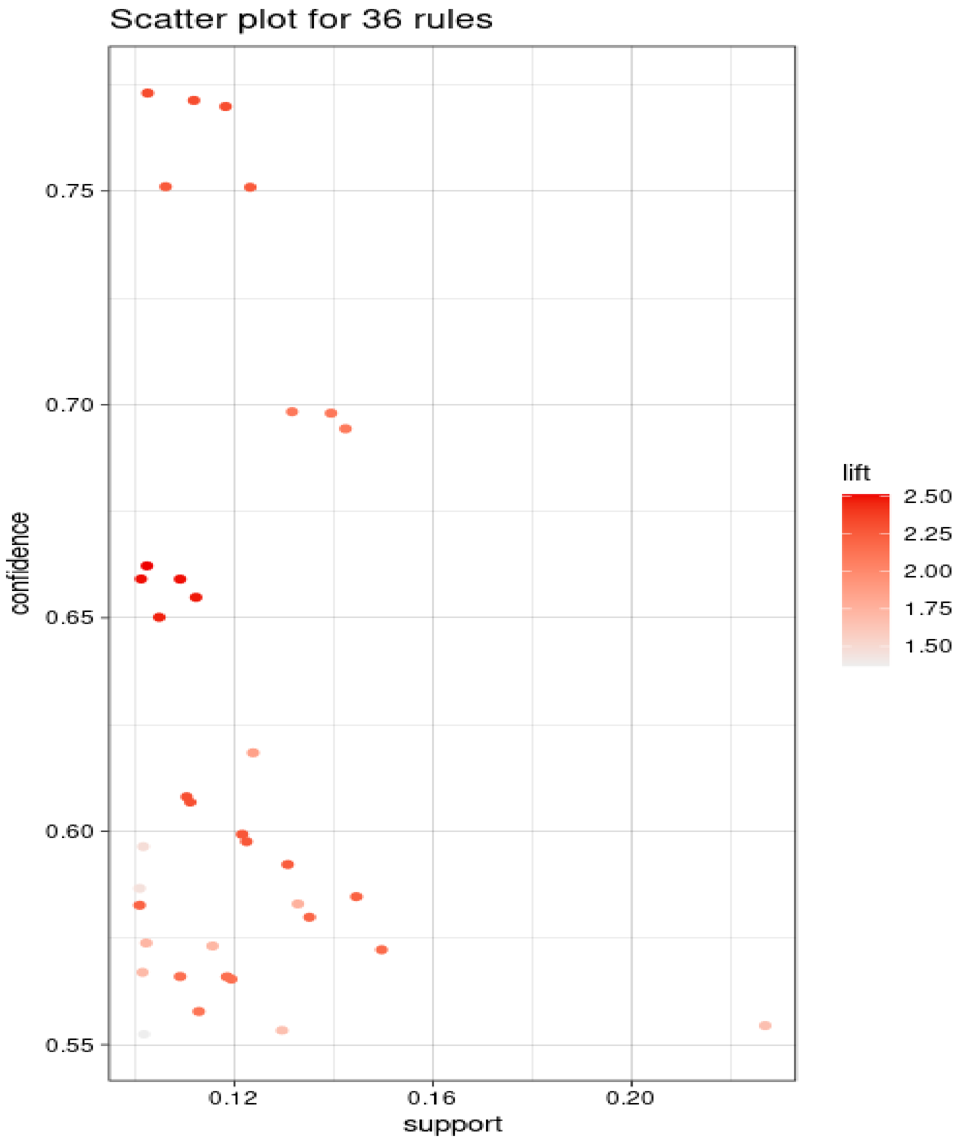
Los valores por ejemplo de “chi squared” para las 26 primeras reglas son muy altos (también son bastante altos para las últimas 10). Esto indica una relación muy fuerte entre la parte izquierda y derecha de las reglas. Entiendo por tanto que las reglas de asociación obtenidas tienen validez (no son triviales).

[Nota: he tenido que reducir mucho las salidas del código. Si no el fichero word generado por el RStudio salía tan grande que el LibreOffice no era capaz de abrirlo.]

Visualización de las reglas

Con la ayuda de la librería arulesViz, vamos a visualizar las reglas obtenidas, para ver si podemos deducir algo más:

```
plot(rules_pruned)
```



Se puede ver gráficamente que el grado de confianza supera el 55% (lógico, ese fue el parámetro que elegí), y el soporte no es muy alto. Tuve que disminuirlo hasta ese nivel para poder encontrar reglas de asociación relacionadas con los niveles de ingreso.

Gráfico de grupos de reglas:

```
plot(rules_pruned, method="grouped")
```

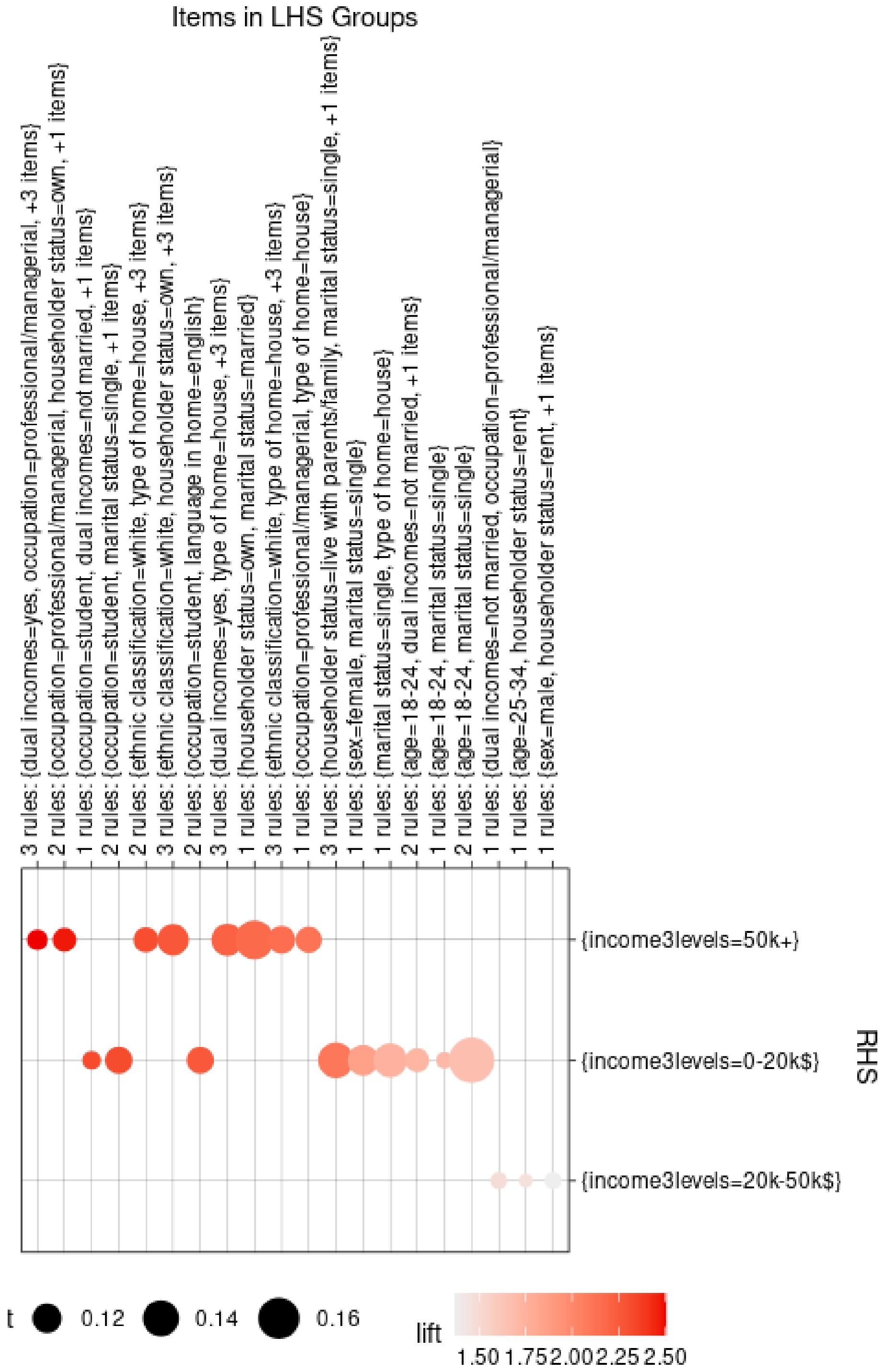
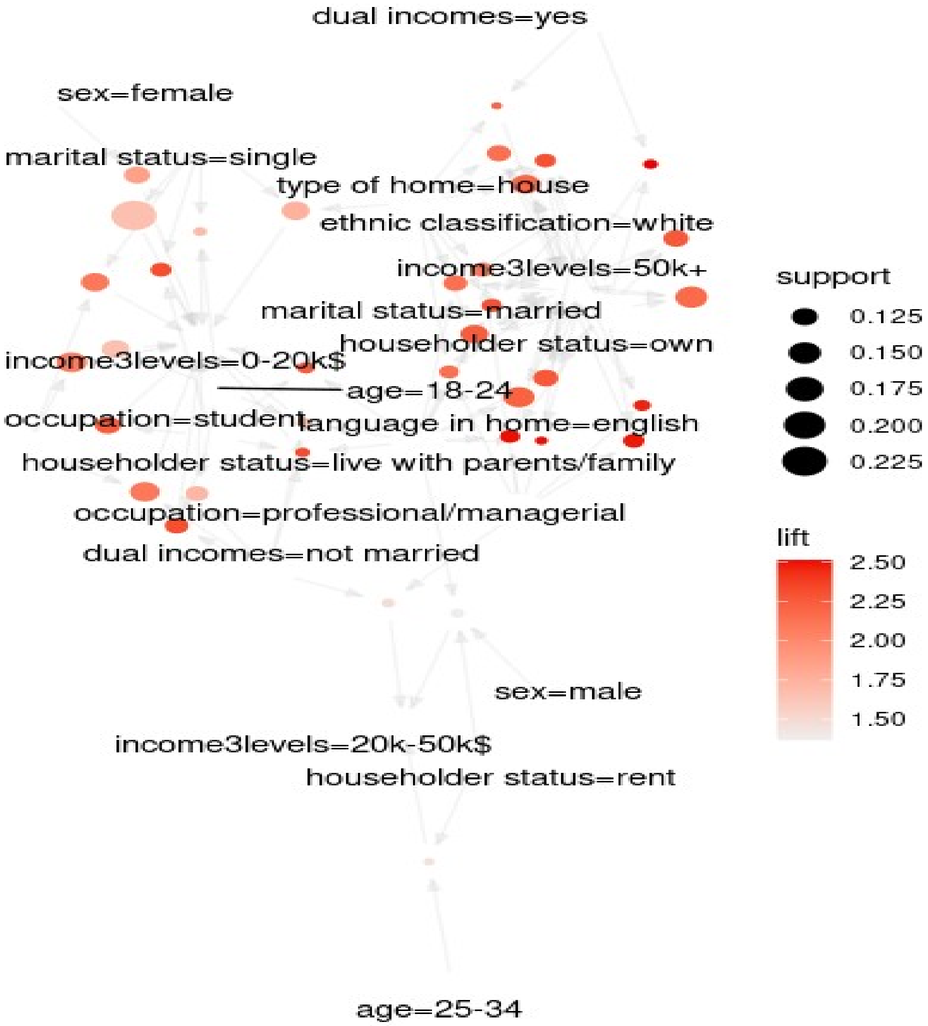


Gráfico de conexiones entre reglas:

```r
plot(rules_pruned, method="graph")
```



En este gráfico pueden apreciar 2 "clusters", y algunos puntos sueltos en la parte inferior. En el cluster de la parte izquierda se puede ver "estudiante", "sueldo inferior a 20k $", "edad 18 a 24", "vive con los padres" .. En el cluster de la parte derecha "casado", "sueldo superior a 50k $", "vivienda en propiedad" … Y en la parte inferior, lo también descrito: las personas con edades entre 25 y 34, que viven de alquiler, y tienen un sueldo intermedio entre 20k y 50k $.