Trabajo final de Minería de datos (aprendizaje no supervisado y detección de anomalías): REGLAS DE ASOCIACIÓN

Dataset: Customer Personality Analysis

Lucía Almorox Antón

Contents

0. Configuración del documento y carga de paquetes.

```
rm(list=ls())
set.seed(333)
```

```
list.of.packages <- c("formatR","tidyverse","ggplot2","Amelia","readr","lubridate",
                      "arules", "arulesViz","dlookr")

new.packages <- list.of.packages[!(list.of.packages %in%
                                   installed.packages()[,"Package"])]

if(length(new.packages)) install.packages(new.packages)
lapply(list.of.packages, library, character.only = TRUE)

knitr::opts_chunk$set(echo = TRUE, tidy=TRUE,
    tidy.opts=list(width.cutoff=40), fig.align = 'center',cache=TRUE)
ggplot2::theme_set(theme_bw())
```

1. Descripción del dataset.

El dataset sobre el que se pretende extraer conocimiento mediante reglas de asociación se llama "Customer Personality Analysis". Se encuentra en Kaggle (https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis?datasetId=1546318) y contiene un análisis detallado de los clientes de una empresa, incluyendo tanto características personales del cliente (fecha de nacimiento, estado civil, etc.) como información sobre su interacción con la empresa (compras, quejas, etc.).

El análisis de la personalidad del cliente ayuda a una empresa a comprender mejor a sus clientes y a identificar distintos tipos de ellos, facilitando la modificación de productos de acuerdo con las necesidades, comportamientos e inquietudes específicos de cada tipo.

Mediante la extracción de reglas de asociación de esta base de datos se pretende identificar relaciones no triviales entre las características de los clientes y las preferencias de compra o su forma de interacción con la empresa.

1.1. Variables del dataset.

El dataset se compone de 2240 clientes (observaciones/filas) sobre los que se detallan 29 características (columnas).

```
Clientes <- read.table("marketing_campaign.csv",
    sep = "\t", header = T)
head(Clientes)
```

```
##     ID Year_Birth  Education Marital_Status Income Kidhome Teenhome Dt_Customer
## 1 5524       1957 Graduation         Single  58138       0        0  04-09-2012
## 2 2174       1954 Graduation         Single  46344       1        1  08-03-2014
## 3 4141       1965 Graduation       Together  71613       0        0  21-08-2013
## 4 6182       1984 Graduation       Together  26646       1        0  10-02-2014
## 5 5324       1981        PhD        Married  58293       1        0  19-01-2014
## 6 7446       1967     Master       Together  62513       0        1  09-09-2013
##   Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1      58      635        88             546             172               88
## 2      38       11         1               6               2                1
## 3      26      426        49             127             111               21
## 4      26       11         4              20              10                3
## 5      94      173        43             118              46               27
```

```
## 6      16      520        42              98               0               42
##   MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases
## 1           88                 3               8                  10
## 2            6                 2               1                   1
## 3           42                 1               8                   2
## 4            5                 2               2                   0
## 5           15                 5               5                   3
## 6           14                 2               6                   4
##   NumStorePurchases NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
## 1                 4                 7            0            0            0
## 2                 2                 5            0            0            0
## 3                10                 4            0            0            0
## 4                 4                 6            0            0            0
## 5                 6                 5            0            0            0
## 6                10                 6            0            0            0
##   AcceptedCmp1 AcceptedCmp2 Complain Z_CostContact Z_Revenue Response
## 1            0            0        0             3        11        1
## 2            0            0        0             3        11        0
## 3            0            0        0             3        11        0
## 4            0            0        0             3        11        0
## 5            0            0        0             3        11        0
## 6            0            0        0             3        11        0
```

```
cat("Dimensión del dataset:", dim(Clientes),
    "\n")
```

```
## Dimensión del dataset: 2240 29
```

```
cat("Nombres de las variables del dataset:\n",
    colnames(Clientes), sep = "  ")
```

```
## Nombres de las variables del dataset:
##   ID  Year_Birth  Education  Marital_Status  Income  Kidhome  Teenhome  Dt_Customer  Recency  MntWin
```

```
Clientes_o <- Clientes
```

En Kaggle se describe el significado de cada variable de la siguiente forma:

Personas

ID: Identificador único del cliente

Year_Birth: Año de nacimiento del cliente

Education: Nivel de educación del cliente

Marital_Status: Estado civil del cliente

Income: Ingreso familiar anual del cliente

Kidhome: Número de niños en el hogar del cliente

Teenhome: Número de adolescentes en el hogar del cliente

Dt_Customer: Fecha de alta del cliente en la empresa

Recency: Número de días desde la última compra del cliente

Complain: 1 si el cliente se ha quejado en los dos últimos años, 0 si no

Productos

MntWines: Cantidad gastada en vino en los dos últimos años

MntFruits: Cantidad gastada en fruta en los dos últimos años

MntMeatProducts: Cantidad gastada en carne en los dos últimos años

MntFishProducts: Cantidad gastada en pescado en los dos últimos años

MntSweetProducts: Cantidad gastada en dulces en los dos últimos años

MntGoldProds: Cantidad gastada en oro en los dos últimos años

Promoción

NumDealsPurchases: Número de compras realizadas con descuento.

AcceptedCmp1: 1 si el cliente aceptó la oferta en la primera campaña, 0 si no

AcceptedCmp2: 1 si el cliente aceptó la oferta en la segunda campaña, 0 si no

AcceptedCmp3: 1 si el cliente aceptó la oferta en la tercera campaña, 0 si no

AcceptedCmp4: 1 si el cliente aceptó la oferta en la cuarta campaña, 0 si no

AcceptedCmp5: 1 si el cliente aceptó la oferta en la quinta campaña, 0 si no

Response: 1 si el cliente aceptó la oferta en la última campaña, 0 si no

Lugar

NumWebPurchases: Número de compras realizadas a través de la página web de la empresa

NumStorePurchases: Número de compras realizadas directamente en tiendas

NumCatalogPurchases: Número de compras realizadas utilizando el catálogo *(entiendo que estas compras se realizan igualmente en tienda o través de la página web).*

NumWebVisitsMonth: Número de visitas a la página web de la empresa en el último mes

```
str(Clientes)
```

```
## 'data.frame':    2240 obs. of  29 variables:
##  $ ID                 : int  5524 2174 4141 6182 5324 7446 965 6177 4855 5899 ...
##  $ Year_Birth         : int  1957 1954 1965 1984 1981 1967 1971 1985 1974 1950 ...
##  $ Education          : chr  "Graduation" "Graduation" "Graduation" "Graduation" ...
##  $ Marital_Status     : chr  "Single" "Single" "Together" "Together" ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 5648 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Dt_Customer        : chr  "04-09-2012" "08-03-2014" "21-08-2013" "10-02-2014" ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines           : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits          : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeatProducts    : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFishProducts    : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweetProducts   : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGoldProds       : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
```

```
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ AcceptedCmp4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp1       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp2       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Complain           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Z_CostContact      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Z_Revenue          : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
```

```
cat("\n")
```

```
table(diagnose(Clientes)$types)
```

```
##
## character   integer
##         3        26
```

```
cat("\n")
```

```
summary(Clientes)
```

```
##        ID          Year_Birth    Education         Marital_Status
##  Min.   :    0   Min.   :1893   Length:2240        Length:2240
##  1st Qu.: 2828   1st Qu.:1959   Class :character   Class :character
##  Median : 5458   Median :1970   Mode  :character   Mode  :character
##  Mean   : 5592   Mean   :1969
##  3rd Qu.: 8428   3rd Qu.:1977
##  Max.   :11191   Max.   :1996
##
##      Income          Kidhome          Teenhome      Dt_Customer
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Length:2240
##  1st Qu.: 35303   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median : 51382   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   : 52247   Mean   :0.4442   Mean   :0.5062
##  3rd Qu.: 68522   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :666666   Max.   :2.0000   Max.   :2.0000
##  NA's   :24
##     Recency         MntWines         MntFruits     MntMeatProducts
##  Min.   : 0.00   Min.   :   0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.:24.00   1st Qu.:  23.75   1st Qu.:  1.0   1st Qu.:  16.0
##  Median :49.00   Median : 173.50   Median :  8.0   Median :  67.0
##  Mean   :49.11   Mean   : 303.94   Mean   : 26.3   Mean   : 166.9
##  3rd Qu.:74.00   3rd Qu.: 504.25   3rd Qu.: 33.0   3rd Qu.: 232.0
##  Max.   :99.00   Max.   :1493.00   Max.   :199.0   Max.   :1725.0
##
##  MntFishProducts  MntSweetProducts  MntGoldProds    NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 2.000
```

```
##  Mean   : 37.53   Mean   : 27.06   Mean   : 44.02   Mean   : 2.325
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :263.00   Max.   :362.00   Max.   :15.000
##
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.00     Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.00     1st Qu.: 3.000
##  Median : 4.000   Median : 2.000      Median : 5.00     Median : 6.000
##  Mean   : 4.085   Mean   : 2.662      Mean   : 5.79     Mean   : 5.317
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.00     3rd Qu.: 7.000
##  Max.   :27.000   Max.   :28.000      Max.   :13.00     Max.   :20.000
##
##   AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.07277   Mean   :0.07455   Mean   :0.07277   Mean   :0.06429
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   AcceptedCmp2        Complain        Z_CostContact   Z_Revenue
##  Min.   :0.00000   Min.   :0.000000   Min.   :3     Min.   :11
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:3     1st Qu.:11
##  Median :0.00000   Median :0.000000   Median :3     Median :11
##  Mean   :0.01339   Mean   :0.009375   Mean   :3     Mean   :11
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:3     3rd Qu.:11
##  Max.   :1.00000   Max.   :1.000000   Max.   :3     Max.   :11
##
##     Response
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1491
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

2. Modificaciones necesarias del dataset.

2.1. Eliminaciíon de variables que tienen el mismo valor para todas las observaciones.

Todos los clientes tienen Z_CostContact = 3 y Z_Revenue = 11, por lo que se procede a la eliminación de dichas variables (de hecho, estas variables no aparecen en la descripción del dataset, por lo que no sabemos qué son).

```
Clientes[["Z_CostContact"]] = NULL
Clientes[["Z_Revenue"]] = NULL
```

2.2. Eliminación de variables con valores únicos.

Dado que los valores de la variable ID son específicos y diferentes para cada cliente, se procede a la eliminación de dicha variable.

Lo mismo ocurre con la variable Dt_Customer (fecha en la que se dio de alta el cliente), sin embargo, en este caso se considera interesante conservar el año de la fecha, en lugar de eliminar la variable.

```
Clientes[["ID"]] = NULL

Clientes = separate(Clientes, Dt_Customer,
    c("Dt_Day", "Dt_Month", "Dt_Year"), sep = "-")
Clientes[["Dt_Day"]] = NULL
Clientes[["Dt_Month"]] = NULL
head(Clientes)
```

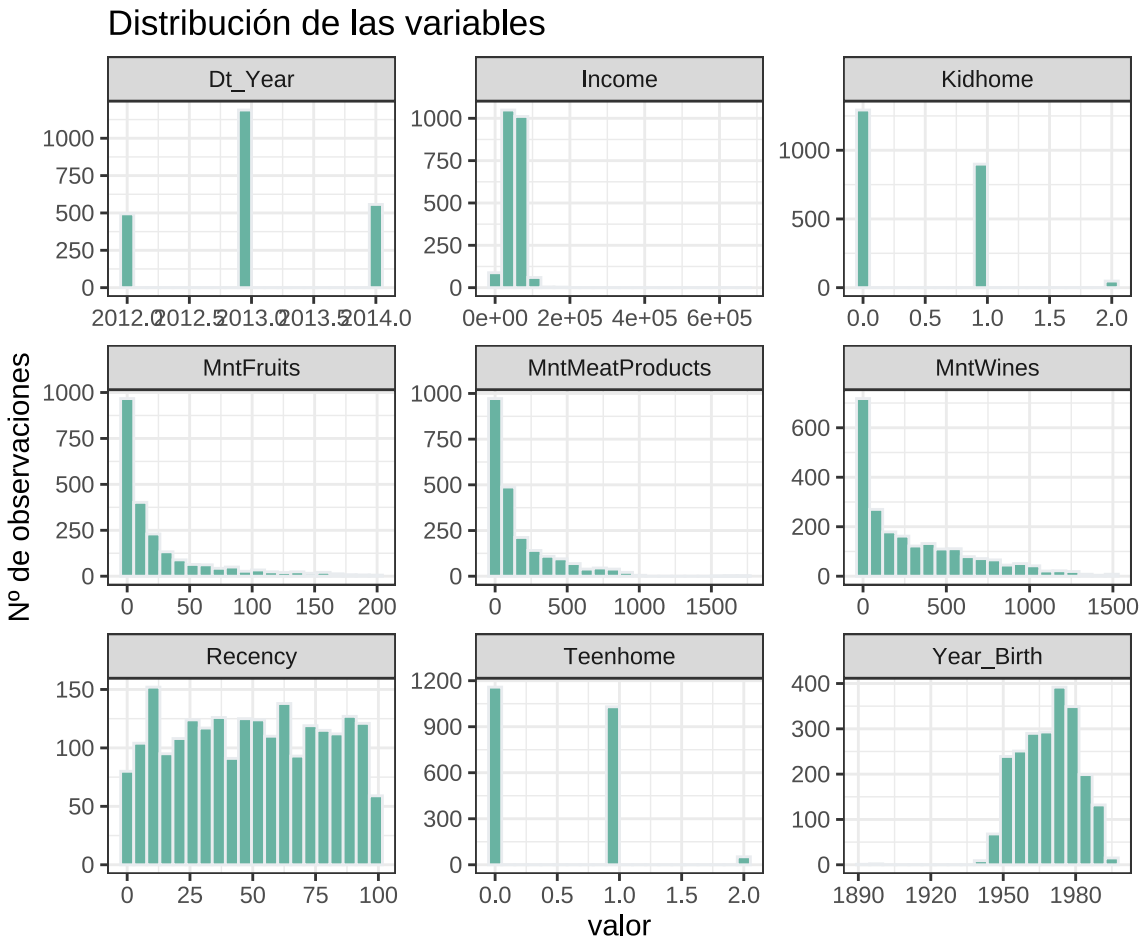
```
##   Year_Birth  Education Marital_Status Income Kidhome Teenhome Dt_Year Recency
## 1       1957 Graduation         Single  58138       0        0    2012      58
## 2       1954 Graduation         Single  46344       1        1    2014      38
## 3       1965 Graduation       Together  71613       0        0    2013      26
## 4       1984 Graduation       Together  26646       1        0    2014      26
## 5       1981        PhD        Married  58293       1        0    2014      94
## 6       1967     Master       Together  62513       0        1    2013      16
##   MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1      635        88             546             172               88
## 2       11         1               6               2                1
## 3      426        49             127             111               21
## 4       11         4              20              10                3
## 5      173        43             118              46               27
## 6      520        42              98               0               42
##   MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases
## 1           88                 3               8                  10
## 2            6                 2               1                   1
## 3           42                 1               8                   2
## 4            5                 2               2                   0
## 5           15                 5               5                   3
## 6           14                 2               6                   4
##   NumStorePurchases NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
## 1                 4                 7           0           0           0
## 2                 2                 5           0           0           0
## 3                10                 4           0           0           0
## 4                 4                 6           0           0           0
## 5                 6                 5           0           0           0
## 6                10                 6           0           0           0
##   AcceptedCmp1 AcceptedCmp2 Complain Response
## 1            0            0        0        1
## 2            0            0        0        0
## 3            0            0        0        0
## 4            0            0        0        0
## 5            0            0        0        0
## 6            0            0        0        0
```

2.3. Discretización de variables numéricas.

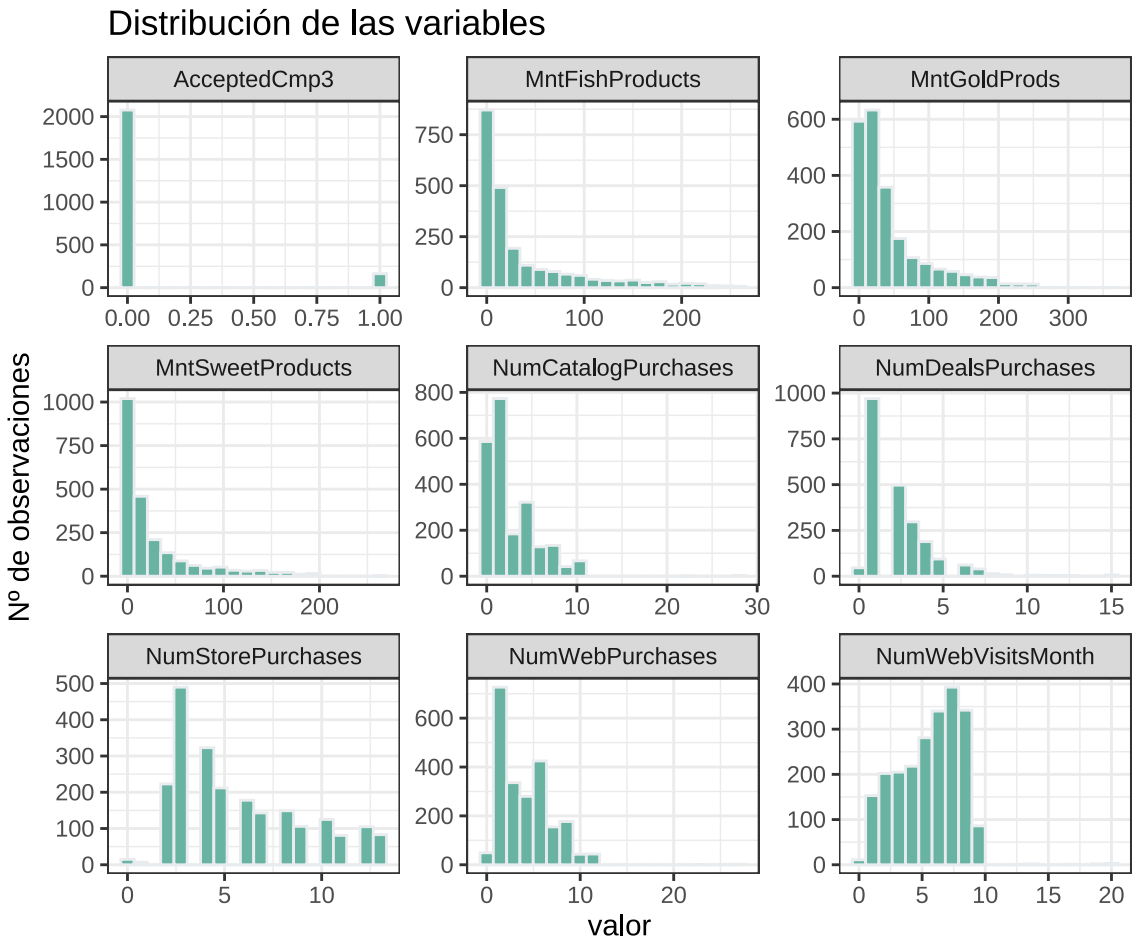
A continuación, se utilizan histogramas para conocer la distribución de las variables y poder así estimar cortes razonables para la discretización.

```
# library(rlang)

Clientes[["Dt_Year"]] <- as.numeric(Clientes[["Dt_Year"]])

plot_histogram <- function(data) {
    data %>%
        select_if(is.numeric) %>%
        pivot_longer(everything(), names_to = "cols",
            values_to = "value") %>%
        ggplot(aes(x = value)) + geom_histogram(bins = 20,
        fill = "#69b3a2", color = "#e9ecef") +
        facet_wrap(~cols, ncol = 3, scales = "free",
            ) + labs(title = "Distribución de las variables",
        x = "valor", y = "Nº de observaciones")
}

plot_histogram(Clientes[, 1:11])
```

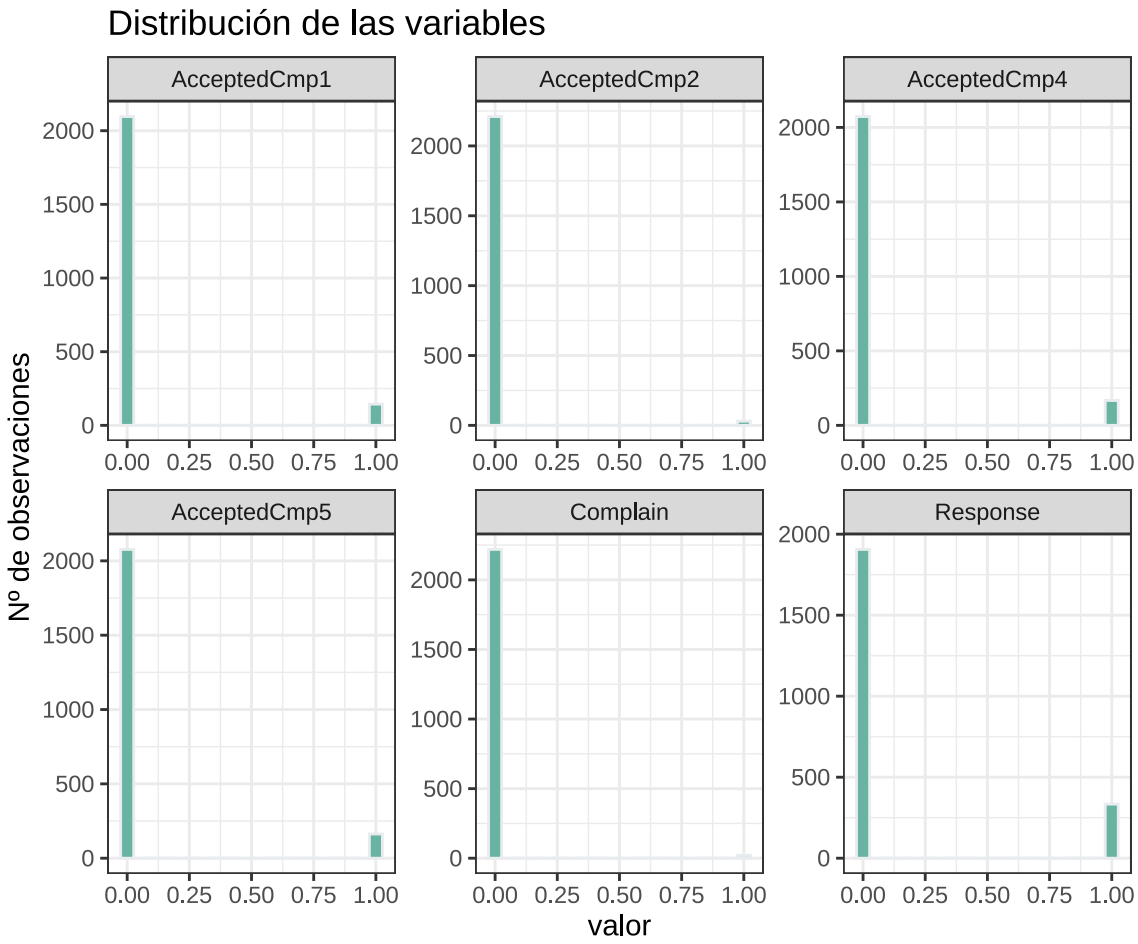
```
## Warning: Removed 24 rows containing non-finite values (stat_bin).
```




```
plot_histogram(Clientes[, 12:20])
```




```
plot_histogram(Clientes[, 21:26])
```

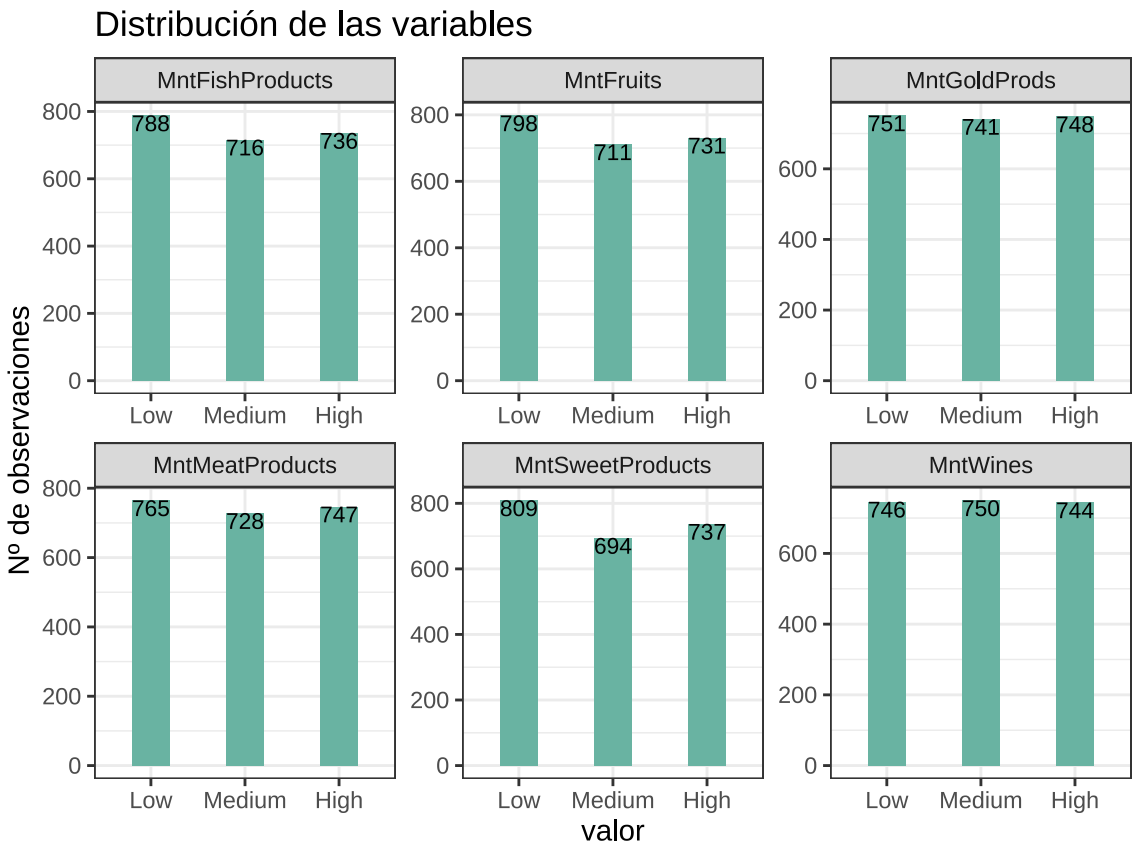


Las variables que indican la cantidad de gasto en ciertos tipos de productos (las que empiezan por "Mnt") se discretizan en 3 niveles: Low, Medium y High, en base a los cuantiles 0.333 y 0.666.

```
mnt_names <- Clientes %>%
    select(starts_with("Mnt")) %>%
    colnames()

for (mnt in mnt_names) {
    Clientes[[mnt]] = ordered(cut(Clientes[[mnt]],
        c(-Inf, quantile(Clientes[[mnt]],
            probs = 0.333), quantile(Clientes[[mnt]],
            probs = 0.666), Inf)), labels = c("Low",
        "Medium", "High"))
}
```

```
Clientes %>%
    select(starts_with("Mnt")) %>%
    pivot_longer(everything(), names_to = "cols",
        values_to = "value") %>%
    ggplot(aes(x = value)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
```

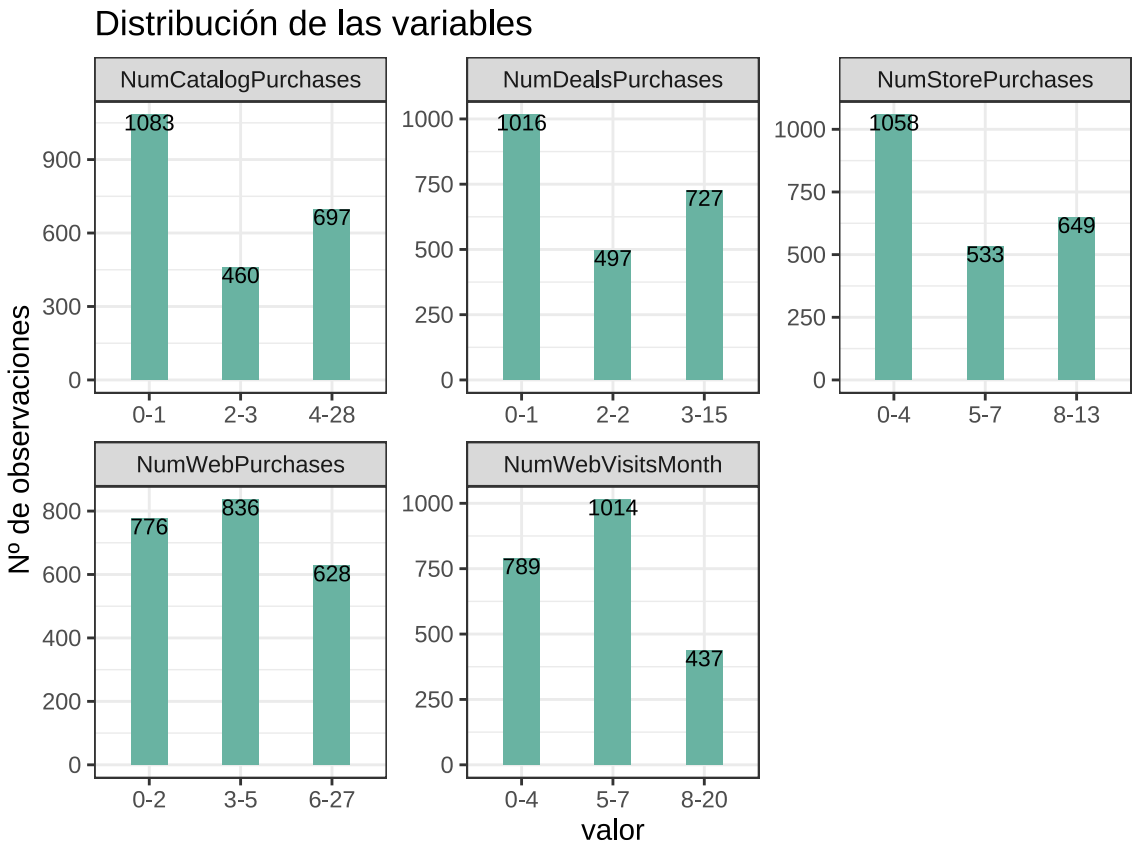
```
    facet_wrap(~cols, ncol = 3, scales = "free",
        ) + labs(title = "Distribución de las variables",
    x = "valor", y = "Nº de observaciones")
```



Se procede de la misma manera con las variables que empiezan por "Num", que hacen referencia al número de compras de distintos tipos de productos:

```
num_names <- Clientes %>%
    select(starts_with("Num")) %>%
    colnames()

for (num in num_names) {
    labels_num = c(paste(0, quantile(Clientes[[num]],
        probs = 0.333), sep = "-"), paste(quantile(Clientes[[num]],
        probs = 0.333) + 1, quantile(Clientes[[num]],
        probs = 0.666), sep = "-"), paste(quantile(Clientes[[num]],
        probs = 0.666) + 1, max(Clientes[[num]]),
        sep = "-"))
    labels_num
    Clientes[[num]] = ordered(cut(Clientes[[num]],
        c(-Inf, quantile(Clientes[[num]],
            probs = 0.333), quantile(Clientes[[num]],
            probs = 0.666), Inf), labels = labels_num,
        right = T))
}
```

```
for (num in num_names) {
    Clientes[[num]] <- as.character(Clientes[[num]])
}

Clientes %>%
    select(starts_with("Num")) %>%
    pivot_longer(everything(), names_to = "cols",
        values_to = "value") %>%
    ggplot(aes(x = value)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
    facet_wrap(~cols, ncol = 3, scales = "free",
        ) + labs(title = "Distribución de las variables",
    x = "valor", y = "Nº de observaciones")
```

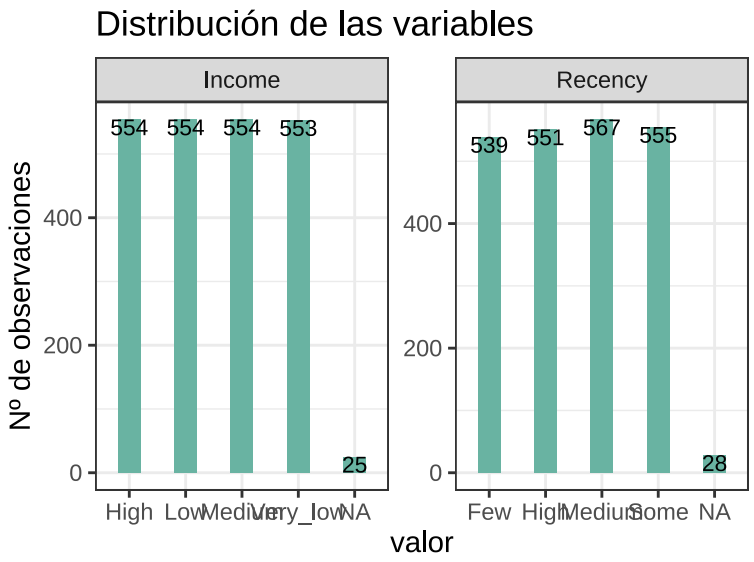


Las variables Income y Recency se discretizan en 4 intervalos (se utilizan los cuartiles en ambos casos).

```
# Income
breaks_income <- unname(quantile(Clientes[["Income"]],
    na.rm = T))
labels_income <- c("Very_low", "Low", "Medium",
    "High")
Clientes[["Income"]] = ordered(cut(Clientes[["Income"]],
    breaks_income, labels = labels_income,
    right = T))
```

```r
# Recency
breaks_rec <- unname(quantile(Clientes[["Recency"]],
    na.rm = T))
labels_rec <- c("Few", "Some", "Medium",
    "High")
Clientes[["Recency"]] = ordered(cut(Clientes[["Recency"]],
    breaks_rec, labels = labels_rec, right = T))
```

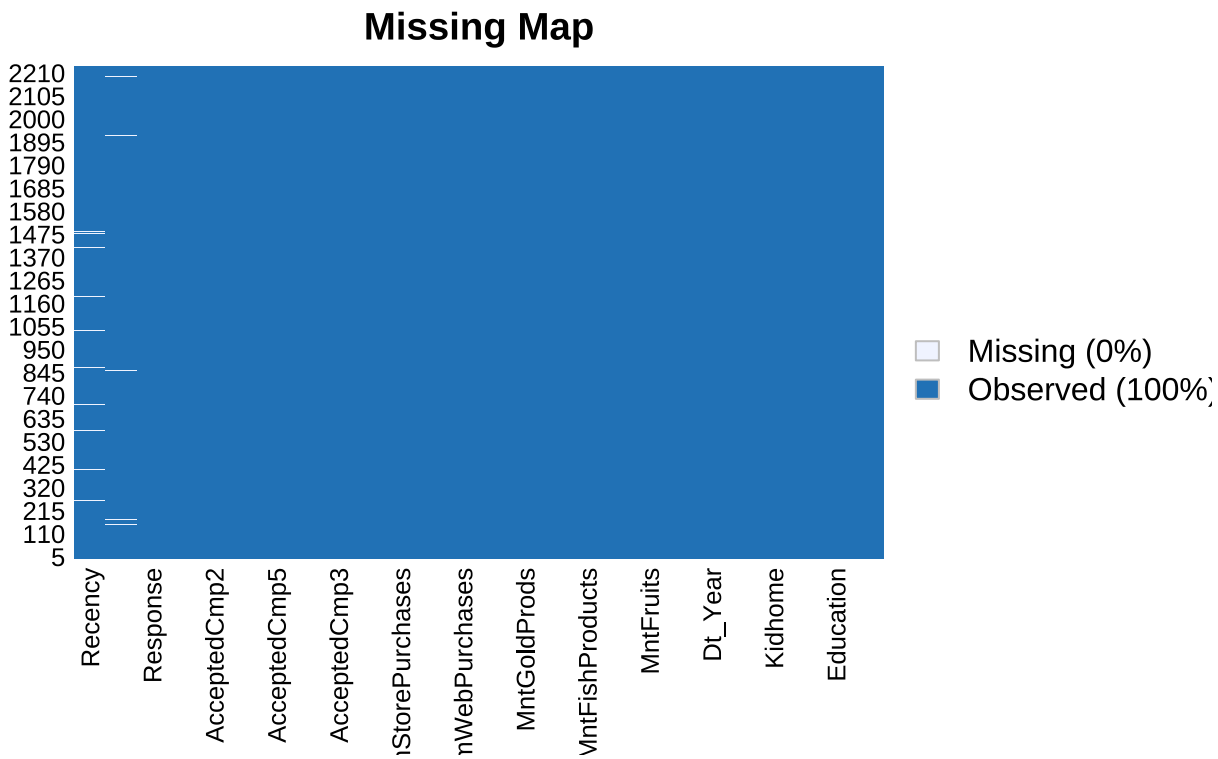
```r
last_2 <- c("Income", "Recency")
for (i in last_2) {
    Clientes[[i]] <- as.character(Clientes[[i]])
}
Clientes %>%
    select(one_of(last_2)) %>%
    pivot_longer(everything(), names_to = "cols",
        values_to = "value") %>%
    ggplot(aes(x = value)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
    facet_wrap(~cols, ncol = 3, scales = "free",
        ) + labs(title = "Distribución de las variables",
    x = "valor", y = "Nº de observaciones")
```



En ambas variables aparecen missing values (NA). En principio, no considero necesario realizar imputación (Income=NA, por ejemplo, será un item muy poco frecuente con el que no creo que se formen reglas de asociación).

A continuación se observa si alguna variable más contiene missing values:

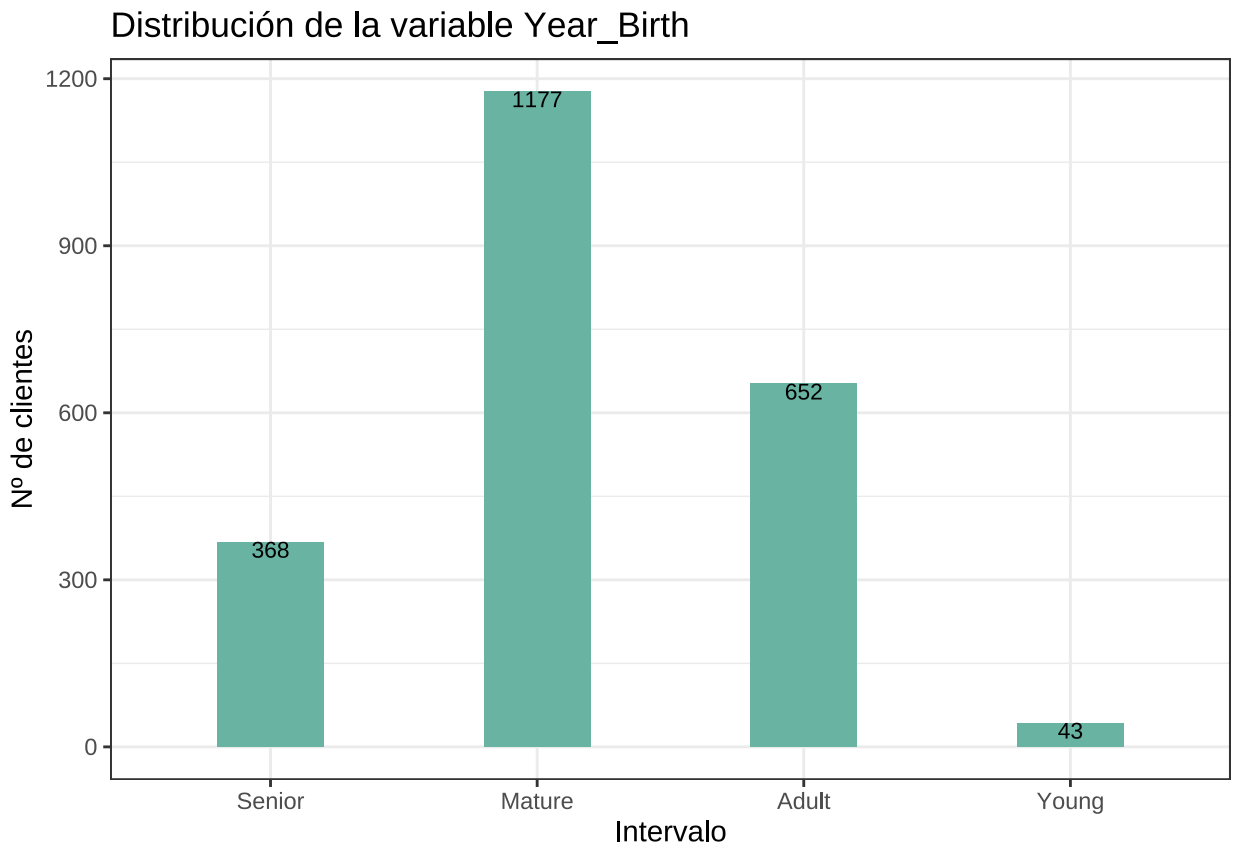
```r
missmap(Clientes, main = "Missing Map")  # Paquete Amelia
```



El año de nacimiento tiene como mínimo 1893 y como máximo 1996, por lo que la variable comprende un rango de 103 años (parece poco realista que una persona nacida en 1893 siga viva en el momento en el que una persona del 1996 tiene al menos 18 años, pero bueno). A continuación se discretiza dicho rango en los siguientes intervalos: 1893-1955, 1956-1975, 1976-1990, 1991-1996; con etiquetas: "Senior", "Mature","Adult","Young".

```
Clientes[["Year_Birth"]] = ordered(cut(Clientes[["Year_Birth"]],
    c(-Inf, 1955, 1975, 1990, Inf), labels = c("Senior",
        "Mature", "Adult", "Young"), right = T))
```

```
Clientes %>%
    ggplot(aes(x = Year_Birth)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
    labs(title = "Distribución de la variable Year_Birth",
        x = "Intervalo", y = "Nº de clientes")
```



2.4. Variables binarias a factores.

```
Clientes$AcceptedCmp1 <- fct_collapse(as.factor(Clientes$AcceptedCmp1),
    No = "0", Yes = "1")

Clientes$AcceptedCmp2 <- fct_collapse(as.factor(Clientes$AcceptedCmp2),
    No = "0", Yes = "1")

Clientes$AcceptedCmp3 <- fct_collapse(as.factor(Clientes$AcceptedCmp3),
    No = "0", Yes = "1")


Clientes$AcceptedCmp4 <- fct_collapse(as.factor(Clientes$AcceptedCmp4),
    No = "0", Yes = "1")


Clientes$AcceptedCmp5 <- fct_collapse(as.factor(Clientes$AcceptedCmp5),
    No = "0", Yes = "1")

Clientes$Complain <- fct_collapse(as.factor(Clientes$Complain),
    No = "0", Yes = "1")

Clientes$Response <- fct_collapse(as.factor(Clientes$Response),
    No = "0", Yes = "1")
```

2.5. Variables Kidhome, Teenhome y Dt_Year a factores.

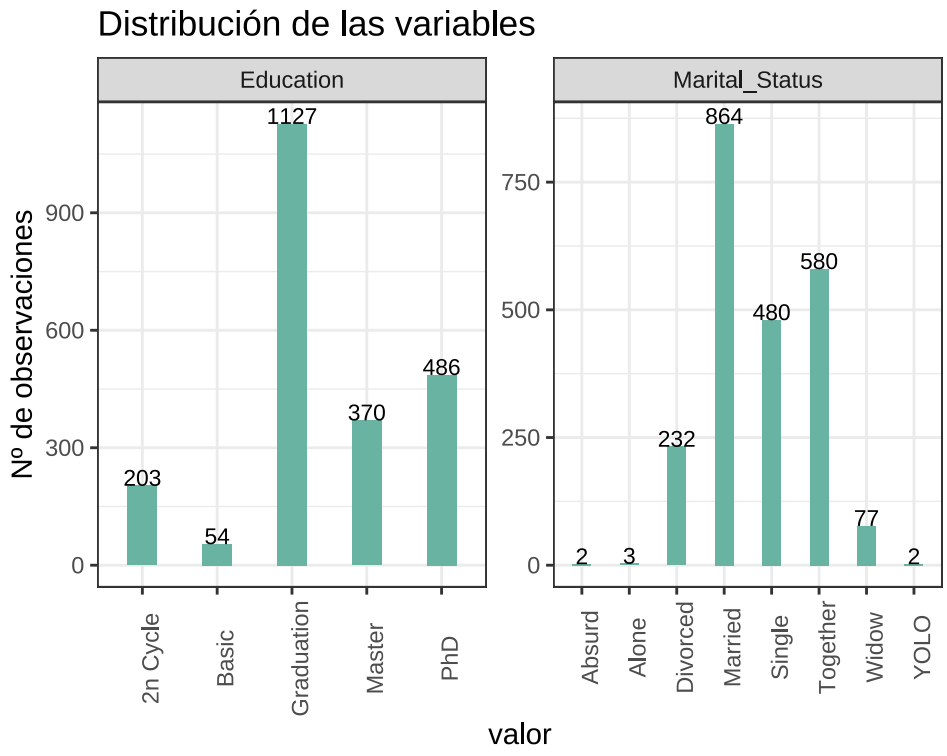
Dado que las variables KidHome y Teenhome solo tienen 3 valores cada una (0,1 o 2), se convierten a caracteres (no se discretizan). Se procede de la misma manera con la variable Dt_Year (valores: 2012, 2013, 2014).

```
Clientes$Kidhome <- as.factor(Clientes$Kidhome)
Clientes$Teenhome <- as.factor(Clientes$Teenhome)
Clientes$Dt_Year <- as.factor(Clientes$Dt_Year)
```

2.6. Visualización y modificación de variables categóricas.

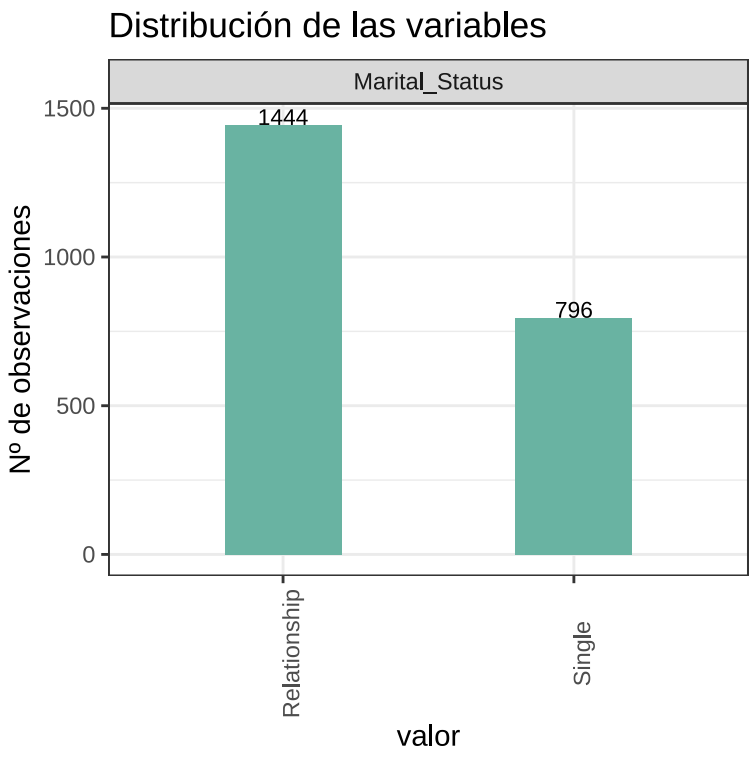
Se representa la distribución de cada una:

```
categ <- c("Education", "Marital_Status")
Clientes %>%
    select(one_of(categ)) %>%
    pivot_longer(everything(), names_to = "cols",
        values_to = "value") %>%
    ggplot(aes(x = value)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 0, cex = 3) +
    facet_wrap(~cols, ncol = 2, scales = "free",
        ) + labs(title = "Distribución de las variables",
    x = "valor", y = "Nº de observaciones") +
    theme(axis.text.x = element_text(angle = 90))
```



La variable Education, por ahora, parece razonable dejarla con las cinco categorías, pues quizás haya diferencias en cómo un graduado o un doctorando interacciona con la empresa. Sin embargo, sí parece necesario agrupar algunos valores de la variable Marital_Status. En concreto en este caso se crean dos categorías unicamente: "Single" y "Relationship".

```
Clientes$Marital_Status[Clientes$Marital_Status ==
    "Absurd" | Clientes$Marital_Status ==
    "Alone" | Clientes$Marital_Status ==
    "Divorced" | Clientes$Marital_Status ==
    "Widow" | Clientes$Marital_Status ==
    "YOLO"] <- "Single"

Clientes$Marital_Status[Clientes$Marital_Status ==
    "Married" | Clientes$Marital_Status ==
    "Together"] <- "Relationship"

Clientes %>%
    select(one_of("Marital_Status")) %>%
    pivot_longer(everything(), names_to = "cols",
        values_to = "value") %>%
    ggplot(aes(x = value)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 0, cex = 3) +
    facet_wrap(~cols, ncol = 2, scales = "free",
        ) + labs(title = "Distribución de las variables",
    x = "valor", y = "Nº de observaciones") +
    theme(axis.text.x = element_text(angle = 90))
```



3. Apriori sobre el dataset con las modificaciones estrictamente necesarias.

3.1. Obtención de información de la base de datos.

Se convierte el dataframe a un conjunto de transacciones (base de datos).

```
# Para cambiar las variables que son
# cadenas de caracteres a factores:
Clientes <- as.data.frame(unclass(Clientes),
    stringsAsFactors = TRUE)

ClientesBD <- as(Clientes, "transactions")
ClientesBD
```

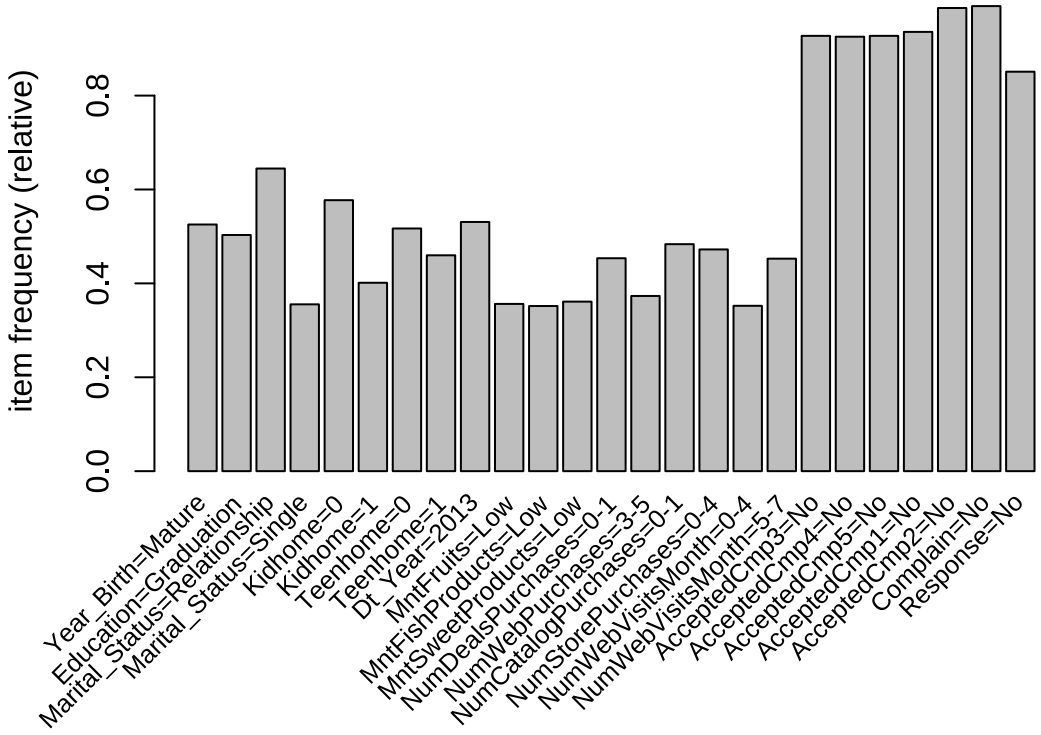
```
## transactions in sparse format with
##  2240 transactions (rows) and
##  75 items (columns)
```

```
summary(ClientesBD)
```

```
## transactions as itemMatrix in sparse format with
##  2240 rows (elements/itemsets/transactions) and
##  75 columns (items) and a density of 0.3463512
##
## most frequent items:
##     Complain=No AcceptedCmp2=No AcceptedCmp1=No AcceptedCmp3=No AcceptedCmp5=No
##            2219            2210            2096            2077            2077
##         (Other)
##           47508
##
## element (itemset/transaction) length distribution:
## sizes
##   25   26
##   53 2187
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   26.00   26.00   25.98   26.00   26.00
##
## includes extended item information - examples:
##              labels  variables levels
## 1 Year_Birth=Senior Year_Birth Senior
## 2 Year_Birth=Mature Year_Birth Mature
## 3  Year_Birth=Adult Year_Birth  Adult
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

Se observan los items más frecuentes (en este caso, con una frecuencia igual o superior a 0.35 en las transacciones de la base de datos).

```
# itemFrequencyPlot(ClientesBD, support
# = 0.3, cex.names=0.8)
itemFrequencyPlot(ClientesBD, support = 0.35,
    cex.names = 0.8)
```



La información de esta gráfica concuerda con la que se obtenía a través de los gráficos de barras. Para cada campaña, la gran mayoría de clientes rechaza la oferta, al igual que la gran mayoría de clientes no se ha quejado nunca a la empresa. Estos items tienen una frecuencia superior a 0.8. Los siguientes items más frecuentes son Marital_Satuts=Relationship y KidHome=0.

A continuación, se obtienen los items frecuentes para un soporte de 0.35.

```
iClientes <- apriori(ClientesBD, parameter = list(support = 0.35,
    target = "frequent"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.35      1
##  maxlen            target  ext
##      10 frequent itemsets TRUE
```

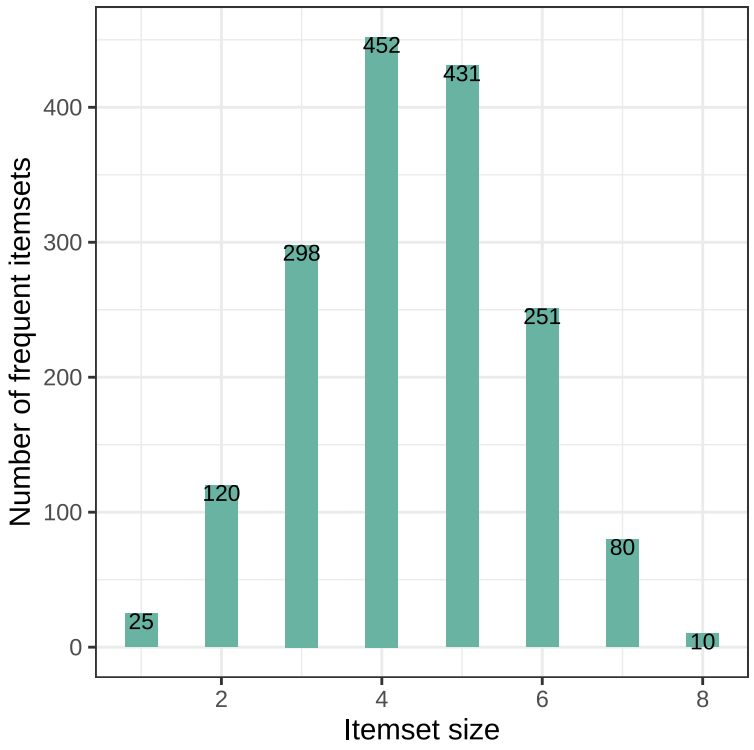
```
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 784
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[75 item(s), 2240 transaction(s)] done [0.01s].
## sorting and recoding items ... [25 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.03s].
## sorting transactions ... done [0.00s].
## writing ... [1667 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Los ordenamos por el valor del
# soporte
iClientes <- sort(iClientes, by = "support")
inspect(head(iClientes, n = 10))
```

```
##      items                                 support   count
## [1]  {Complain=No}                         0.9906250 2219
## [2]  {AcceptedCmp2=No}                     0.9866071 2210
## [3]  {AcceptedCmp2=No, Complain=No}        0.9772321 2189
## [4]  {AcceptedCmp1=No}                     0.9357143 2096
## [5]  {AcceptedCmp1=No, AcceptedCmp2=No}    0.9281250 2079
## [6]  {AcceptedCmp5=No}                     0.9272321 2077
## [7]  {AcceptedCmp3=No}                     0.9272321 2077
## [8]  {AcceptedCmp1=No, Complain=No}        0.9263393 2075
## [9]  {AcceptedCmp4=No}                     0.9254464 2073
## [10] {AcceptedCmp4=No, AcceptedCmp2=No}    0.9218750 2065
```

Se representa el número de itemsets frecuentes para cada tamaño de itemset.

```
as.data.frame(size(iClientes)) %>%
    ggplot(aes(x = size(iClientes))) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
    labs(x = "Itemset size", y = "Number of frequent itemsets")
```



Se observan algunos de los itemsets frecuentes de tamaño 8 (el máximo):

```
inspect(head(iClientes[size(iClientes) ==
    8], 3))
```

```
##     items                              support count
## [1] {Marital_Status=Relationship,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No,
##      Response=No}                    0.4772321  1069
## [2] {NumCatalogPurchases=0-1,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No,
##      Response=No}                    0.4102679   919
## [3] {Dt_Year=2013,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
```

```
##      Complain=No,
##      Response=No}                  0.3973214   890
```

En esta base de datos hay varios items muy frecuentes, por lo que al combinarlos se crean itemsets que, a pesar de su gran tamaño, siguen siendo frecuentes (con un soporte superior al soporte mínimo). Esto no es algo que interese a la hora de extraer reglas de asociación: se crearán reglas inútiles, con un alto soporte en el consecuente: si A->B es una regla donde B tiene un soporte extremadamente alto, "observar A implica una alta probabilidad de observar B" no porque haya una relación entre ambos items, sino porque B aparece en casi todas las transacciones (es decir, cualquier cosa implica una alta probabilidad de encontrar a B).

Habrá que filtrar las reglas obtenidas para eliminar este tipo de reglas.

A continuación, se extraen los itemset maximales (aquellos a los que si se les añade un solo item más, dejan de ser frecuentes -para un soporte de 0.35 en este caso-). Se muestran algunos de ellos.

Los que tienen mayor soporte:

```
imaxClientes <- iClientes[is.maximal(iClientes)]
length(imaxClientes)
```

```
## [1] 47
```

```
inspect(head(sort(imaxClientes, by = "support"),
    5))
```

```
##     items                             support count
## [1] {Marital_Status=Relationship,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No,
##      Response=No}                  0.4772321  1069
## [2] {NumCatalogPurchases=0-1,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No,
##      Response=No}                  0.4102679   919
## [3] {Dt_Year=2013,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No,
##      Response=No}                  0.3973214   890
## [4] {Teenhome=0,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
```

```
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No}                  0.3839286   860
## [5] {Year_Birth=Mature,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No,
##      Response=No}                  0.3839286   860
```

Los que tienen menor soporte:

```
inspect(head(sort(imaxClientes, by = "support",
    decreasing = F), 5))
```

```
##     items                     support count
## [1] {NumWebVisitsMonth=0-4,
##      Complain=No}                0.35   784
## [2] {Teenhome=0,
##      NumDealsPurchases=0-1}      0.35   784
## [3] {MntSweetProducts=Low,
##      AcceptedCmp1=No,
##      Complain=No}                0.35   784
## [4] {Kidhome=1,
##      AcceptedCmp1=No,
##      Response=No}                0.35   784
## [5] {NumDealsPurchases=0-1,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No}                0.35   784
```

A continuación, se extraen los itemsets cerrados (si se les añade un solo item más, su soporte baja -no necesariamente por debajo del soporte mínimo-). Se muestran algunos:

```
icloClientes <- iClientes[is.closed(iClientes)]
# length(icloClientes)
inspect(head(sort(icloClientes, by = "support")))
```

```
##     items                                support   count
## [1] {Complain=No}                        0.9906250 2219
## [2] {AcceptedCmp2=No}                    0.9866071 2210
## [3] {AcceptedCmp2=No, Complain=No}       0.9772321 2189
## [4] {AcceptedCmp1=No}                    0.9357143 2096
## [5] {AcceptedCmp1=No, AcceptedCmp2=No}   0.9281250 2079
## [6] {AcceptedCmp5=No}                    0.9272321 2077
```

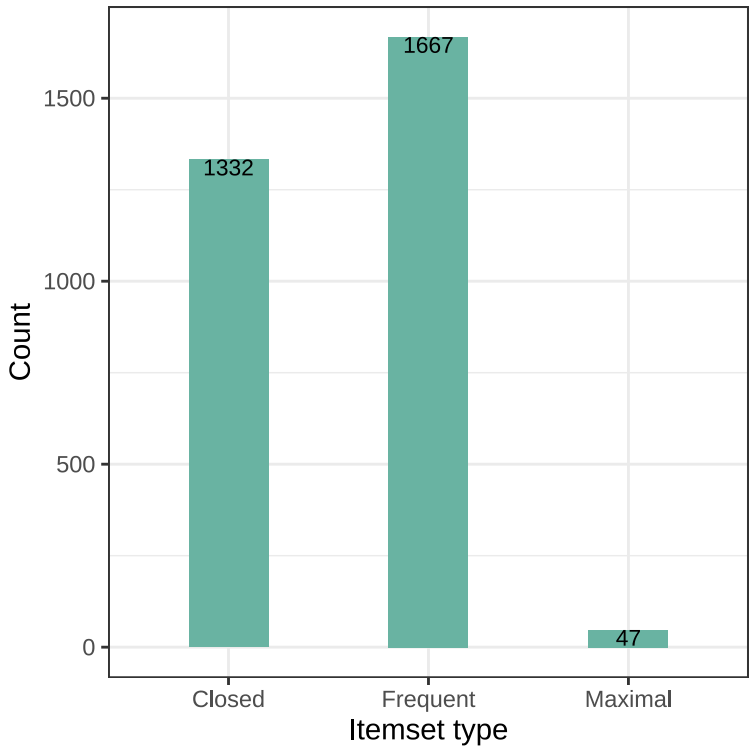
Por su definición, dentro de los cerrados se incluyen todos los maximales. Podemos ver algunos de ellos a continuación (los que se muestran tienen el soporte mínimo, por lo que si baja ese soporte, obviamente dejan de ser itemsets frecuentes):

```
inspect(head(sort(icloClientes, by = "support",
    decreasing = F)))
```

```
##     items                      support count
## [1] {NumWebVisitsMonth=0-4,
##      Complain=No}                0.35   784
## [2] {Teenhome=0,
##      NumDealsPurchases=0-1}      0.35   784
## [3] {MntSweetProducts=Low,
##      AcceptedCmp1=No,
##      Complain=No}                0.35   784
## [4] {Kidhome=1,
##      AcceptedCmp1=No,
##      Response=No}                0.35   784
## [5] {NumDealsPurchases=0-1,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Complain=No}                0.35   784
## [6] {NumWebVisitsMonth=5-7,
##      AcceptedCmp3=No,
##      AcceptedCmp4=No,
##      AcceptedCmp5=No,
##      AcceptedCmp1=No,
##      AcceptedCmp2=No,
##      Response=No}                0.35   784
```

Se representa el número de itemsets de cada tipo:

```
# barplot(
# c(frequent=length(iClientes),
# closed=length(icloClientes),
# maximal=length(imaxClientes)),
# ylab='count', xlab='itemsets')

df <- data.frame(trt = c("Frequent", "Closed",
    "Maximal"), outcome = c(length(iClientes),
    length(icloClientes), length(imaxClientes)))

ggplot(df, aes(trt, outcome)) + geom_col(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "identity",
    aes(label = outcome), vjust = 1, cex = 3) +
    labs(x = "Itemset type", y = "Count")
```



Se puede observar la gran diferencia entre el número de cerrados y de maximales. Con cualquiera de los dos grupos se pueden obtener todos los itemsets frecuentes, aunque para los maximales se desconoce el soporte (habría que volver a calcularlo).

3.2. Extracción de reglas (apriori).

Se utiliza el método apriori del paquete arules para obtener reglas con un soporte mínimo de 0.1, una confianza mínima de 0.8 y una longitud mínima de 2.

```
rules <- apriori(ClientesBD, parameter = list(support = 0.1,
    confidence = 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     0.1      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: 224
##
## set item appearances ...[0 item(s)] done [0.00s].
```

```
## set transactions ...[75 item(s), 2240 transaction(s)] done [0.00s].
## sorting and recoding items ... [64 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10

## Warning in apriori(ClientesBD, parameter = list(support = 0.1, confidence =
## 0.8, : Mining stopped (maxlen reached). Only patterns up to a length of 10
## returned!

##  done [4.70s].
## writing ... [2551741 rule(s)] done [0.64s].
## creating S4 object  ... done [2.16s].
```

Se han escrito 2551741 reglas, que, obviamente han de ser filtradas.

```
summary(rules)
```

```
## set of 2551741 rules
##
## rule length distribution (lhs + rhs):sizes
##      2      3      4      5      6      7      8      9     10
##    453   7998  51081 172572 366741 540782 590772 497038 324304
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    7.00    8.00    7.65    9.00   10.00
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.1000   Min.   :0.8000   Min.   :0.1000   Min.   :0.855
##  1st Qu.:0.1071   1st Qu.:0.9173   1st Qu.:0.1129   1st Qu.:1.014
##  Median :0.1196   Median :0.9765   Median :0.1272   Median :1.078
##  Mean   :0.1308   Mean   :0.9511   Mean   :0.1380   Mean   :1.552
##  3rd Qu.:0.1424   3rd Qu.:0.9970   3rd Qu.:0.1509   3rd Qu.:2.117
##  Max.   :0.9772   Max.   :1.0000   Max.   :0.9906   Max.   :3.637
##      count
##  Min.   : 224.0
##  1st Qu.: 240.0
##  Median : 268.0
##  Mean   : 293.1
##  3rd Qu.: 319.0
##  Max.   :2189.0
##
## mining info:
##        data ntransactions support confidence
##  ClientesBD          2240     0.1        0.8
##                                                                                call
##  apriori(data = ClientesBD, parameter = list(support = 0.1, confidence = 0.8, minlen = 2))
```

```
inspect(head(rules))
```

```
##     lhs                    rhs                support   confidence coverage
## [1] {Response=Yes}      => {AcceptedCmp4=No} 0.1214286 0.8143713  0.1491071
```

```
## [2] {Response=Yes}      => {AcceptedCmp2=No} 0.1401786 0.9401198  0.1491071
## [3] {Response=Yes}      => {Complain=No}     0.1477679 0.9910180  0.1491071
## [4] {Year_Birth=Senior} => {Response=No}     0.1383929 0.8423913  0.1642857
## [5] {Year_Birth=Senior} => {AcceptedCmp4=No} 0.1482143 0.9021739  0.1642857
## [6] {Year_Birth=Senior} => {AcceptedCmp5=No} 0.1482143 0.9021739  0.1642857
##     lift      count
## [1] 0.8799767 272
## [2] 0.9528816 314
## [3] 1.0003967 331
## [4] 0.9900087 310
## [5] 0.9748527 332
## [6] 0.9729752 332
```

Se observan aquí ejemplos de reglas inútiles (con alto soporte en el consecuente).

```
rulesSorted = sort(rules, by = "confidence")
inspect(head(rulesSorted, 10))
```

```
##      lhs                           rhs                 support confidence  coverage     lift count
## [1]  {Income=Very_low}          => {AcceptedCmp5=No} 0.2468750          1 0.2468750 1.078479   553
## [2]  {Income=Very_low}          => {AcceptedCmp1=No} 0.2468750          1 0.2468750 1.068702   553
## [3]  {Income=Very_low}          => {AcceptedCmp2=No} 0.2468750          1 0.2468750 1.013575   553
## [4]  {Income=Low}               => {AcceptedCmp5=No} 0.2473214          1 0.2473214 1.078479   554
## [5]  {MntWines=Low}             => {AcceptedCmp5=No} 0.3330357          1 0.3330357 1.078479   746
## [6]  {MntWines=Low}             => {AcceptedCmp1=No} 0.3330357          1 0.3330357 1.068702   746
## [7]  {MntMeatProducts=Low}      => {AcceptedCmp5=No} 0.3415179          1 0.3415179 1.078479   765
## [8]  {NumCatalogPurchases=0-1} => {AcceptedCmp5=No} 0.4834821          1 0.4834821 1.078479  1083
## [9]  {NumDealsPurchases=3-15,
##       NumWebVisitsMonth=8-20}  => {AcceptedCmp5=No} 0.1022321          1 0.1022321 1.078479   229
## [10] {Kidhome=1,
##       NumWebVisitsMonth=8-20}  => {AcceptedCmp2=No} 0.1294643          1 0.1294643 1.013575   290
```

Es necesario filtrar reglas mediante una medida de calidad que penalice el tener un gran soporte en el consecuente, por ejemplo, lift (valores de 1 indican independencia entre antecedente y consecuente, buscamos valores superiores a 1).

```
# rules_lift <- subset(rules, subset =
# lift > 1.2)
rules_lift <- subset(rules, subset = lift >
    1.4)
inspect(head(rules_lift, 20))
```

```
##      lhs                            rhs                       support
## [1]  {Income=Very_low}           => {MntWines=Low}            0.2183036
## [2]  {Income=Very_low}           => {MntMeatProducts=Low}     0.2008929
## [3]  {Income=Very_low}           => {NumStorePurchases=0-4}   0.2370536
## [4]  {Income=Very_low}           => {NumCatalogPurchases=0-1} 0.2325893
## [5]  {Income=High}               => {MntMeatProducts=High}    0.2165179
## [6]  {Income=High}               => {NumWebVisitsMonth=0-4}   0.1995536
## [7]  {Income=High}               => {Kidhome=0}               0.2281250
## [8]  {NumStorePurchases=8-13}    => {Kidhome=0}               0.2566964
## [9]  {NumCatalogPurchases=4-28} => {MntMeatProducts=High}    0.2522321
```

```
## [10] {NumCatalogPurchases=4-28} => {Kidhome=0}                0.2915179
## [11] {MntFruits=High}           => {Kidhome=0}                0.2897321
## [12] {MntFishProducts=High}     => {Kidhome=0}                0.2937500
## [13] {MntSweetProducts=High}    => {Kidhome=0}                0.2901786
## [14] {MntWines=High}            => {Kidhome=0}                0.3013393
## [15] {MntWines=Low}             => {MntMeatProducts=Low}      0.2901786
## [16] {MntMeatProducts=Low}      => {MntWines=Low}             0.2901786
## [17] {MntWines=Low}             => {NumWebPurchases=0-2}      0.2754464
## [18] {MntWines=Low}             => {NumStorePurchases=0-4}    0.3294643
## [19] {MntWines=Low}             => {NumCatalogPurchases=0-1} 0.3232143
## [20] {MntMeatProducts=High}     => {Kidhome=0}                0.3022321
##      confidence coverage  lift     count
## [1]  0.8842676  0.2468750 2.655174 489
## [2]  0.8137432  0.2468750 2.382725 450
## [3]  0.9602170  0.2468750 2.032974 531
## [4]  0.9421338  0.2468750 1.948642 521
## [5]  0.8754513  0.2473214 2.625182 485
## [6]  0.8068592  0.2473214 2.290703 447
## [7]  0.9223827  0.2473214 1.597941 511
## [8]  0.8859784  0.2897321 1.534874 575
## [9]  0.8106169  0.3111607 2.430766 565
## [10] 0.9368723  0.3111607 1.623043 653
## [11] 0.8878249  0.3263393 1.538073 649
## [12] 0.8940217  0.3285714 1.548808 658
## [13] 0.8819539  0.3290179 1.527902 650
## [14] 0.9072581  0.3321429 1.571739 675
## [15] 0.8713137  0.3330357 2.551298 650
## [16] 0.8496732  0.3415179 2.551298 650
## [17] 0.8270777  0.3330357 2.387441 617
## [18] 0.9892761  0.3330357 2.094498 738
## [19] 0.9705094  0.3330357 2.007332 724
## [20] 0.9062918  0.3334821 1.570065 677
```

Ahora sí se ven reglas más interesantes. Los clientes con mayor ingreso tienden a tener cero niños en casa. Por ello también vemos que los clientes que invierten mayores cantidades en la empresa y que efectúan un mayor número de compras en esta, tienden a tener 0 niños en casa.

```
# %in% -> contener alguno de los items
# que se especifiquen

`%!in%` = Negate(`%in%`)  # que no tengan ninguno de los items que se
# especifiquen

# %oin% -> que solo contenga los items
# que se especifiquen

# %ain% -> que contenga todos los items
# que se especifiquen

`%!ain%` = Negate(`%ain%`)  # que no contenga todos los items que se especifiquen.


rules_lift2 <- subset(rules, subset = lift >
    1.2 & lhs %ain% c("NumCatalogPurchases=4-28",
```

```
    "NumStorePurchases=8-13") & rhs %in%
    "Kidhome=0" & size(rules) > 4 & lhs %!in%
    c("AcceptedCmp1=No", "AcceptedCmp2=No",
        "AcceptedCmp3=No", "AcceptedCmp4=No",
        "AcceptedCmp5=No", "Complain=No"))

inspect(head(rules_lift2, 5))
```

```
##     lhs                             rhs           support confidence  coverage     lift count
## [1] {Income=High,
##      MntMeatProducts=High,
##      NumCatalogPurchases=4-28,
##      NumStorePurchases=8-13}    => {Kidhome=0} 0.1013393  0.9458333 0.1071429 1.638567   227
## [2] {MntFruits=High,
##      MntFishProducts=High,
##      NumCatalogPurchases=4-28,
##      NumStorePurchases=8-13}    => {Kidhome=0} 0.1000000  0.9491525 0.1053571 1.644317   224
## [3] {MntFruits=High,
##      MntSweetProducts=High,
##      NumCatalogPurchases=4-28,
##      NumStorePurchases=8-13}    => {Kidhome=0} 0.1017857  0.9539749 0.1066964 1.652671   228
## [4] {MntFruits=High,
##      MntMeatProducts=High,
##      NumCatalogPurchases=4-28,
##      NumStorePurchases=8-13}    => {Kidhome=0} 0.1093750  0.9459459 0.1156250 1.638762   245
## [5] {MntFishProducts=High,
##      MntSweetProducts=High,
##      NumCatalogPurchases=4-28,
##      NumStorePurchases=8-13}    => {Kidhome=0} 0.1093750  0.9607843 0.1138393 1.664468   245
```

Se guarda la primera de estas reglas como interesante:

```
R_INTERESANTES <- as(rules_lift2[1], "data.frame")
R_INTERESANTES
```

```
##                                                                                                                                  ru
## 80231 {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13} => {Kidhome=
##         support confidence  coverage     lift count
## 80231 0.1013393  0.9458333 0.1071429 1.638567   227
```

Para resolver el problema de las reglas con un alto soporte en el consecuente, también podemos, en lugar de filtrar por lift, filtrar por reglas con alta confianza y que no tengan en el consecuente los items que sabemos que tienen un gran soporte.

```
rules_mod <- subset(rules, subset = rhs %!in%
    c("AcceptedCmp1=No", "AcceptedCmp2=No",
        "AcceptedCmp3=No", "AcceptedCmp4=No",
        "AcceptedCmp5=No", "Complain=No",
        "Response=No") & confidence > 0.8)
inspect(head(rules_mod, 20))
```

```
##      lhs                            rhs                          support
## [1]  {Income=Very_low}           => {MntWines=Low}              0.2183036
## [2]  {Income=Very_low}           => {MntMeatProducts=Low}       0.2008929
## [3]  {Income=Very_low}           => {NumStorePurchases=0-4}     0.2370536
## [4]  {Income=Very_low}           => {NumCatalogPurchases=0-1}   0.2325893
## [5]  {Income=High}               => {MntMeatProducts=High}      0.2165179
## [6]  {Income=High}               => {NumWebVisitsMonth=0-4}     0.1995536
## [7]  {Income=High}               => {Kidhome=0}                 0.2281250
## [8]  {NumWebPurchases=6-27}      => {Kidhome=0}                 0.2254464
## [9]  {NumStorePurchases=8-13}    => {Kidhome=0}                 0.2566964
## [10] {NumCatalogPurchases=4-28} => {MntMeatProducts=High}      0.2522321
## [11] {NumCatalogPurchases=4-28} => {Kidhome=0}                 0.2915179
## [12] {MntFruits=High}            => {Kidhome=0}                 0.2897321
## [13] {MntFishProducts=High}      => {Kidhome=0}                 0.2937500
## [14] {MntSweetProducts=High}     => {Kidhome=0}                 0.2901786
## [15] {MntWines=High}             => {Kidhome=0}                 0.3013393
## [16] {MntWines=Low}              => {MntMeatProducts=Low}       0.2901786
## [17] {MntMeatProducts=Low}       => {MntWines=Low}              0.2901786
## [18] {MntWines=Low}              => {NumWebPurchases=0-2}       0.2754464
## [19] {MntWines=Low}              => {NumStorePurchases=0-4}     0.3294643
## [20] {MntWines=Low}              => {NumCatalogPurchases=0-1}   0.3232143
##      confidence coverage  lift     count
## [1]  0.8842676  0.2468750 2.655174 489
## [2]  0.8137432  0.2468750 2.382725 450
## [3]  0.9602170  0.2468750 2.032974 531
## [4]  0.9421338  0.2468750 1.948642 521
## [5]  0.8754513  0.2473214 2.625182 485
## [6]  0.8068592  0.2473214 2.290703 447
## [7]  0.9223827  0.2473214 1.597941 511
## [8]  0.8041401  0.2803571 1.393097 505
## [9]  0.8859784  0.2897321 1.534874 575
## [10] 0.8106169  0.3111607 2.430766 565
## [11] 0.9368723  0.3111607 1.623043 653
## [12] 0.8878249  0.3263393 1.538073 649
## [13] 0.8940217  0.3285714 1.548808 658
## [14] 0.8819539  0.3290179 1.527902 650
## [15] 0.9072581  0.3321429 1.571739 675
## [16] 0.8713137  0.3330357 2.551298 650
## [17] 0.8496732  0.3415179 2.551298 650
## [18] 0.8270777  0.3330357 2.387441 617
## [19] 0.9892761  0.3330357 2.094498 738
## [20] 0.9705094  0.3330357 2.007332 724
```

Regla número 6: El 81% de los clientes con alto ingreso han consultado poco (o nada) la web en el último mes. Guardamos esta regla como interesante.

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[6],
    "data.frame"))
R_INTERESANTES
```

```
##                                                                                                                ru
## 80231 {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13} => {Kidhome=
## 96                                                                                {Income=High} => {NumWebVisitsMonth=0-
```

```
##         support confidence  coverage     lift count
## 80231 0.1013393  0.9458333 0.1071429 1.638567   227
## 96    0.1995536  0.8068592 0.2473214 2.290703   447
```

No encontramos reglas cuyo consecuente sea únicamente uno de los items que definen tener hijos o adolescentes (no es sorprendente, pues cada uno de estos items tiene un bajo soporte, quizás habría que agrupar ambas variables).

```
rules_mod <- subset(rules, subset = rhs %in%
    c("Teenhome=1", "Teenhome=2", "Kidhome=1",
        "Kidhome=2") & length(rhs) == 1)
inspect(head(rules_mod, 20))
```

Se filtran reglas que tengan en el antecedente el atributo que indica alto número de visitas a la web en el último mes:

```
rules_mod <- subset(rules, subset = lhs %in%
    "NumWebVisitsMonth=8-20" & lhs %!in%
    "Kidhome=1" & lift > 1.2)
inspect(head(rules_mod, 5))
```

```
##     lhs                            rhs                          support confidence  coverage     lift
## [1] {NumStorePurchases=0-4,
##      NumWebVisitsMonth=8-20}    => {NumCatalogPurchases=0-1} 0.1200893  0.8966667 0.1339286 1.854601
## [2] {NumCatalogPurchases=0-1,
##      NumWebVisitsMonth=8-20}    => {NumStorePurchases=0-4}   0.1200893  0.8406250 0.1428571 1.779773
## [3] {NumStorePurchases=0-4,
##      NumWebVisitsMonth=8-20,
##      Response=No}               => {NumCatalogPurchases=0-1} 0.1017857  0.9193548 0.1107143 1.901528
## [4] {NumCatalogPurchases=0-1,
##      NumWebVisitsMonth=8-20,
##      Response=No}               => {NumStorePurchases=0-4}   0.1017857  0.8603774 0.1183036 1.821593
## [5] {NumStorePurchases=0-4,
##      NumWebVisitsMonth=8-20,
##      AcceptedCmp4=No}           => {NumCatalogPurchases=0-1} 0.1200893  0.8966667 0.1339286 1.854601
```

Estas reglas definen un grupo de clientes que visitaron mucho la web en el último mes y que compraron muy poco en tienda y realizaron muy pocas compras utilizando el catálogo en los dos últimos años.

Guardamos, por ejemplo, la primera regla como interesante.

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[1],
    "data.frame"))
R_INTERESANTES
```

```
##                                                                                                                ru
## 80231 {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13} => {Kidhome=
## 96                                                                                {Income=High} => {NumWebVisitsMonth=0
## 580                                {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} => {NumCatalogPurchases=0
##         support confidence  coverage     lift count
## 80231 0.1013393  0.9458333 0.1071429 1.638567   227
## 96    0.1995536  0.8068592 0.2473214 2.290703   447
## 580   0.1200893  0.8966667 0.1339286 1.854601   269
```

```
rules_mod <- subset(rules, subset = rhs %oin%
    "NumWebVisitsMonth=0-4" & lift > 1.2 &
    size(rules) > 3)
inspect(head(rules_mod, 13))
```

```
##      lhs                                rhs                       support confidence  coverage     li
## [1]  {Income=High,
##       Kidhome=0,
##       NumStorePurchases=8-13}       => {NumWebVisitsMonth=0-4} 0.1111607  0.8440678 0.1316964 2.39633
## [2]  {Income=High,
##       NumStorePurchases=8-13,
##       AcceptedCmp3=No}              => {NumWebVisitsMonth=0-4} 0.1066964  0.8020134 0.1330357 2.27694
## [3]  {Income=High,
##       MntFruits=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1339286  0.8547009 0.1566964 2.42651
## [4]  {Income=High,
##       MntFishProducts=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1370536  0.8504155 0.1611607 2.41430
## [5]  {Income=High,
##       MntSweetProducts=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1339286  0.8498584 0.1575893 2.41271
## [6]  {Income=High,
##       MntWines=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1254464  0.8192420 0.1531250 2.32585
## [7]  {Income=High,
##       MntMeatProducts=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1500000  0.8400000 0.1785714 2.38479
## [8]  {Income=High,
##       MntGoldProds=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1013393  0.8664122 0.1169643 2.45971
## [9]  {Income=High,
##       NumWebPurchases=3-5,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1013393  0.9227642 0.1098214 2.61970
## [10] {Income=High,
##       NumDealsPurchases=0-1,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1437500  0.9019608 0.1593750 2.56070
## [11] {Income=High,
##       Teenhome=0,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1424107  0.9036827 0.1575893 2.56558
## [12] {Income=High,
##       Kidhome=0,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1611607  0.8719807 0.1848214 2.47558
## [13] {Marital_Status=Relationship,
##       Income=High,
##       NumCatalogPurchases=4-28}     => {NumWebVisitsMonth=0-4} 0.1026786  0.8185053 0.1254464 2.32370
```

Grupo de reglas 3-8: los clientes que tienen un alto ingreso, que realizan muchas compras utilizando el catálogo y que invierten mucho dinero en cualquier tipo de productos de la empresa tienden a visitar nada o poco la web (puede que los clientes activos utilicen el catálogo y no la web o que los que hacen esto sean solo un subgrupo dentro de los clientes activos).

Guardamos, por ejemplo, la regla 3 como interesante.

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[3],
    "data.frame"))
R_INTERESANTES
```

```
##                                                                                                                                ru
## 80231 {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13} => {Kidhome=
## 96                                                                                  {Income=High} => {NumWebVisitsMonth=0-
## 580                                          {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} => {NumCatalogPurchases=0-
## 16137                        {Income=High,MntFruits=High,NumCatalogPurchases=4-28} => {NumWebVisitsMonth=0-
##         support confidence  coverage     lift count
## 80231 0.1013393  0.9458333 0.1071429 1.638567   227
## 96    0.1995536  0.8068592 0.2473214 2.290703   447
## 580   0.1200893  0.8966667 0.1339286 1.854601   269
## 16137 0.1339286  0.8547009 0.1566964 2.426527   300
```

Para seguir estudiando este grupo de gente, se buscan reglas que relacionen grandes inversiones en distintos tipos de productos y que además tengan en el antecedente el atributo de pocas visitas a la web.

```
high <- c("MntFruits=High", "MntGoldProds=High",
    "MntSweetProducts=High", "MntFishProducts=High",
    "MntMeatProducts=High", "MntWines=High")
rules_mod <- subset(rules, subset = lhs %in%
    "NumWebVisitsMonth=0-4" & lhs %in% high &
    rhs %in% high & lift > 1.2)
inspect(head(rules_mod, 13))
```

```
##      lhs                          rhs                        support confidence  coverage     lift cou
## [1]  {MntFruits=High,
##       NumWebVisitsMonth=0-4}  => {MntFishProducts=High}  0.1750000  0.8082474 0.2165179 2.459883   3
## [2]  {MntFruits=High,
##       NumWebVisitsMonth=0-4}  => {MntSweetProducts=High} 0.1767857  0.8164948 0.2165179 2.481613   3
## [3]  {MntSweetProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntFruits=High}        0.1767857  0.8032454 0.2200893 2.461381   3
## [4]  {MntFruits=High,
##       NumWebVisitsMonth=0-4}  => {MntMeatProducts=High}  0.1794643  0.8288660 0.2165179 2.485488   4
## [5]  {MntGoldProds=High,
##       NumWebVisitsMonth=0-4}  => {MntFruits=High}        0.1392857  0.8210526 0.1696429 2.515948   3
## [6]  {MntFishProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntSweetProducts=High} 0.1776786  0.8105906 0.2191964 2.463668   3
## [7]  {MntSweetProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntFishProducts=High}  0.1776786  0.8073022 0.2200893 2.457007   3
## [8]  {MntFishProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntMeatProducts=High}  0.1821429  0.8309572 0.2191964 2.491759   4
## [9]  {MntMeatProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntFishProducts=High}  0.1821429  0.8095238 0.2250000 2.463768   4
## [10] {MntGoldProds=High,
##       NumWebVisitsMonth=0-4}  => {MntFishProducts=High}  0.1397321  0.8236842 0.1696429 2.506865   3
## [11] {MntSweetProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntMeatProducts=High}  0.1808036  0.8215010 0.2200893 2.463403   4
## [12] {MntMeatProducts=High,
##       NumWebVisitsMonth=0-4}  => {MntSweetProducts=High} 0.1808036  0.8035714 0.2250000 2.442334   4
## [13] {MntWines=High,
##       NumWebVisitsMonth=0-4}  => {MntMeatProducts=High}  0.1575893  0.8588808 0.1834821 2.575493   3
```

Con todas estas reglas, queda claro que los clientes que visitaron muy poco o nada la página web en el último mes y que invierten mucho en un tipo de producto de la empresa, tienden a invertir mucho en otros tipos de productos de la tienda también.

Guardamos, por ejemplo, la regla número 8 como interesante.

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[8],
    "data.frame"))
R_INTERESANTES
```

```
##                                                                                                                                ru
## 80231 {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13} => {Kidhome=
## 96                                                                                {Income=High} => {NumWebVisitsMonth=0-
## 580                                                {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} => {NumCatalogPurchases=0-
## 16137                          {Income=High,MntFruits=High,NumCatalogPurchases=4-28} => {NumWebVisitsMonth=0-
## 4465                                    {MntFishProducts=High,NumWebVisitsMonth=0-4} => {MntMeatProducts=Hig
##         support confidence  coverage     lift count
## 80231 0.1013393  0.9458333 0.1071429 1.638567   227
## 96    0.1995536  0.8068592 0.2473214 2.290703   447
## 580   0.1200893  0.8966667 0.1339286 1.854601   269
## 16137 0.1339286  0.8547009 0.1566964 2.426527   300
## 4465  0.1821429  0.8309572 0.2191964 2.491759   408
```

Se podría pensar en la existencia de dos tipos de clientes activos: los que compran en la web y los que compran en tienda física. Estas últimas reglas podrían estar indicando que los clientes activos que no visitan la página web, no compran a través de la página web, sino en tienda física (y utilizando el catálogo, por lo visto). Esto les haría comprar productos más variados que el resto de clientes: si el cliente quiere comprar pescado, es fácil que no pase por la sección de frutas de la página web, mientras que, en la tienda física, puede que para llegar a la sección objetivo pase antes por muchas otras y acabe comprando cosas que no tenía pensado al entrar.

Sin embargo, se puede comprobar que no se han escrito reglas que relacionen los items "NumWebVisitsMonth=0-4" y "NumWebPurchases=0-2":

```
rel_web <- c("NumWebVisitsMonth=0-4", "NumWebPurchases=0-2")
rules_mod <- subset(rules, subset = lhs %in%
    rel_web & rhs %in% rel_web & lift > 1.2)
inspect(head(rules_mod, 13))
```

Además, al filtrar reglas que tengan en el antecendente un bajo número de **compras** en la web, no se obtienen las reglas que se obtienen cuando el filtro era bajo número de **visitas** a la web.

```
rules_mod <- subset(rules, subset = (rhs %in%
    high | lhs %in% high) & lhs %in% "NumWebPurchases=0-2")
inspect(head(rules_mod, 20))
```

No se obtiene ninguna regla (con los filtros aplicados al principio). No se puede asegurar que los que visitaron poco la web en el último mes, compren poco en la web.

Además, la primera regla guardada como interesante es {Income=High} => {NumWebVisitsMonth=0-4}. Por tanto, parece que los clientes activos, aquellos con alto ingreso, tienen alta probabilidad de haber visitado poco la web el último mes, de usar el catálogo y de comprar distintos tipos de productos (no se trata de un subgrupo dentro de los clientes activos).

Comprobaciones adicionales:

Al filtrar reglas que tengan en el antecendente un mayor número de visitas a la web (clientes interesados aunque no especialmente activos en la empresa), no se obtienen reglas que relacionen estos items con inversiones altas en ningún producto.

```
rules_mod <- subset(rules, subset = lhs %in%
    c("NumWebVisitsMonth=8-20", "NumWebVisitsMonth=5-7") &
    (lhs %in% high | rhs %in% high) & lift >
    1.1)
inspect(head(rules_mod, 20))
```

Tampoco se obtienen cuando el número de compras en tienda o catálogo es bajo (al igual que no se obtenían cuando el número de compras en web era bajo):

```
rules_mod <- subset(rules, subset = lhs %in%
    c("NumStorePurchases=0-4", "NumStorePurchases=5-7",
        "NumCatalogPurchases=0-1", "NumCatalogPurchases=2-3") &
    lhs %in% ("MntFruits=High") & lift >
    1.1)

inspect(head(rules_mod, 20))
```

Tampoco se encuentran reglas que relacionen un número alto de visitas a la web con un gasto alto en ninguno de los productos ni con tener un ingreso alto:

```
rules_mod <- subset(rules, subset = (lhs %in%
    "NumWebVisitsMonth=8-20" | rhs %in% "NumWebVisitsMonth=8-20") &
    (lhs %in% c(high, "Income=High") | rhs %in%
        c(high, "Income=High")) & lift >
    1.1)
inspect(head(rules_mod, 20))
```

Además, para confirmar que haber visitado poco la web en el último mes es una tendencia general de los clientes activos (y no de un subgrupo) se realiza el siguiente análisis sobre el dataset, que indica que los clientes activos (695 clientes clasificados como tal -de froma arbitraria- por tener una alta inversión en al menos 4 productos de la empresa) y con alto número de visitas a la web son solo 37, mientras que los clientes activos con bajo número de visitas a la web son 485:

```
active_and_webvisits <- subset(Clientes,
    NumWebVisitsMonth == "8-20" & (as.numeric(MntFruits ==
        "High") + as.numeric(MntGoldProds ==
        "High") + as.numeric(MntSweetProducts ==
        "High") + as.numeric(MntFishProducts ==
        "High") + as.numeric(MntMeatProducts ==
        "High") + as.numeric(MntWines ==
        "High")) >= 4)

active_and_webvisits
```

```
##      Year_Birth  Education Marital_Status   Income Kidhome Teenhome Dt_Year
## 50       Senior        PhD   Relationship     High       1        1    2012
## 91       Mature        PhD   Relationship     <NA>       2        1    2012
## 112       Young        PhD         Single   Medium       0        0    2012
```

```
## 118       Adult   2n Cycle   Relationship      Low       1        0    2013
## 146       Adult Graduation   Relationship   Medium       0        2    2012
## 210      Senior Graduation         Single   Medium       0        1    2012
## 255      Mature Graduation   Relationship     High       1        1    2012
## 504       Adult        PhD   Relationship   Medium       1        0    2012
## 529      Mature Graduation         Single   Medium       0        0    2012
## 567      Mature Graduation         Single   Medium       0        1    2012
## 631      Mature        PhD   Relationship     High       1        0    2013
## 754      Mature Graduation   Relationship     High       0        0    2013
## 798      Mature   2n Cycle         Single   Medium       0        1    2012
## 805      Mature Graduation   Relationship      Low       0        0    2013
## 809      Mature Graduation   Relationship   Medium       1        1    2013
## 818      Senior Graduation         Single Very_low       0        0    2012
## 826      Mature   2n Cycle         Single     High       0        1    2013
## 889      Mature Graduation   Relationship      Low       1        1    2013
## 926       Adult     Master   Relationship Very_low       0        0    2012
## 950      Mature   2n Cycle         Single   Medium       0        1    2012
## 970       Adult Graduation   Relationship      Low       1        0    2012
## 1036      Adult Graduation         Single Very_low       0        0    2012
## 1127     Mature Graduation         Single     High       1        1    2012
## 1284     Senior     Master   Relationship      Low       0        0    2013
## 1508     Mature   2n Cycle   Relationship      Low       0        1    2012
## 1531     Mature   2n Cycle   Relationship   Medium       0        1    2012
## 1608      Adult   2n Cycle         Single      Low       1        0    2013
## 1627     Mature        PhD         Single      Low       0        1    2012
## 1643      Adult Graduation   Relationship   Medium       1        1    2012
## 1669     Mature        PhD   Relationship     High       1        0    2013
## 1789      Adult   2n Cycle   Relationship   Medium       1        1    2012
## 1812     Mature        PhD         Single      Low       0        1    2012
## 1877     Mature     Master         Single Very_low       0        0    2012
## 1880     Mature Graduation         Single     High       1        0    2013
## 1914     Mature Graduation   Relationship     High       0        1    2012
## 2029     Senior     Master   Relationship      Low       0        1    2012
## 2095     Senior     Master   Relationship      Low       0        1    2012
##      Recency MntWines MntFruits MntMeatProducts MntFishProducts
## 50      Some     High      High            High            High
## 91       Few   Medium      High            High            High
## 112     Some     High    Medium            High          Medium
## 118   Medium   Medium      High          Medium            High
## 146     Some     High    Medium          Medium            High
## 210      Few     High      High            High            High
## 255     High     High      High            High            High
## 504   Medium     High       Low          Medium            High
## 529      Few     High      High            High            High
## 567     High     High    Medium            High            High
## 631     Some     High      High            High            High
## 754      Few     High      High            High            High
## 798      Few     High      High            High            High
## 805     High     High      High            High            High
## 809     High     High      High            High          Medium
## 818   Medium   Medium       Low            High            High
## 826   Medium     High      High            High            High
## 889     Some   Medium    Medium            High            High
## 926     High   Medium      High            High          Medium
```

```
## 950     Some     High      High          Medium            High
## 970     Some   Medium      High            High            High
## 1036  Medium      Low      High          Medium            High
## 1127    High     High      High            High            High
## 1284     Few   Medium      High            High            High
## 1508    High   Medium      High            High            High
## 1531    Some     High      High            High          Medium
## 1608    Some   Medium      High            High            High
## 1627    Some     High      High            High            High
## 1643  Medium   Medium      High            High          Medium
## 1669    Some     High      High            High            High
## 1789  Medium     High    Medium          Medium            High
## 1812  Medium     High      High            High             Low
## 1877     Few   Medium      High            High            High
## 1880    High     High      High            High            High
## 1914    High     High      High            High            High
## 2029     Few     High    Medium            High            High
## 2095     Few     High    Medium            High            High
##      MntSweetProducts MntGoldProds NumDealsPurchases NumWebPurchases
## 50               High       Medium              3-15             3-5
## 91               High         High              3-15            6-27
## 112              High         High               0-1            6-27
## 118              High         High              3-15            6-27
## 146              High         High              3-15             3-5
## 210              High         High              3-15            6-27
## 255              High       Medium              3-15             3-5
## 504              High         High               2-2             3-5
## 529            Medium         High               0-1             3-5
## 567              High         High              3-15             3-5
## 631              High       Medium               2-2            6-27
## 754              High         High               0-1            6-27
## 798            Medium         High              3-15            6-27
## 805              High       Medium              3-15             0-2
## 809            Medium         High              3-15             3-5
## 818              High         High              3-15            6-27
## 826              High       Medium               0-1            6-27
## 889              High         High              3-15            6-27
## 926              High         High              3-15            6-27
## 950            Medium         High               2-2             3-5
## 970              High         High              3-15            6-27
## 1036             High         High               2-2             3-5
## 1127             High         High              3-15             3-5
## 1284             High         High               2-2            6-27
## 1508             High         High              3-15            6-27
## 1531           Medium         High              3-15            6-27
## 1608             High       Medium              3-15            6-27
## 1627             High       Medium              3-15             3-5
## 1643             High         High              3-15            6-27
## 1669             High       Medium               2-2            6-27
## 1789             High         High              3-15             0-2
## 1812             High         High              3-15             3-5
## 1877             High         High              3-15            6-27
## 1880             High         High              3-15             3-5
## 1914             High         High              3-15             3-5
```

```
## 2029             High         High              3-15            6-27
## 2095             High         High              3-15            6-27
##      NumCatalogPurchases NumStorePurchases NumWebVisitsMonth AcceptedCmp3
## 50                   2-3              8-13              8-20           No
## 91                   2-3              8-13              8-20           No
## 112                 4-28               5-7              8-20           No
## 118                  0-1               5-7              8-20           No
## 146                  2-3               5-7              8-20           No
## 210                 4-28              8-13              8-20          Yes
## 255                 4-28              8-13              8-20           No
## 504                  2-3              8-13              8-20           No
## 529                 4-28               0-4              8-20           No
## 567                  2-3              8-13              8-20           No
## 631                  2-3              8-13              8-20           No
## 754                 4-28              8-13              8-20           No
## 798                 4-28              8-13              8-20           No
## 805                 4-28              8-13              8-20           No
## 809                  2-3              8-13              8-20           No
## 818                  2-3               0-4              8-20           No
## 826                  2-3              8-13              8-20           No
## 889                  2-3               5-7              8-20           No
## 926                  0-1               5-7              8-20           No
## 950                  2-3              8-13              8-20           No
## 970                  2-3               5-7              8-20           No
## 1036                 0-1               5-7              8-20           No
## 1127                 2-3               5-7              8-20           No
## 1284                 0-1               5-7              8-20           No
## 1508                4-28               5-7              8-20          Yes
## 1531                 2-3              8-13              8-20          Yes
## 1608                4-28               0-4              8-20          Yes
## 1627                 2-3              8-13              8-20           No
## 1643                 0-1               5-7              8-20           No
## 1669                 2-3              8-13              8-20           No
## 1789                 2-3              8-13              8-20           No
## 1812                4-28               5-7              8-20           No
## 1877                 0-1              8-13              8-20           No
## 1880                4-28              8-13              8-20           No
## 1914                4-28               5-7              8-20           No
## 2029                4-28               5-7              8-20          Yes
## 2095                4-28               5-7              8-20          Yes
##      AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Complain Response
## 50             No           No           No           No       No       No
## 91             No           No           No           No       No       No
## 112           Yes           No           No           No       No      Yes
## 118            No           No           No           No       No       No
## 146            No           No           No           No       No       No
## 210            No           No           No           No       No      Yes
## 255            No           No           No           No       No       No
## 504            No           No           No           No       No       No
## 529            No           No           No           No       No       No
## 567            No           No           No           No       No       No
## 631            No           No           No           No       No       No
## 754           Yes          Yes          Yes           No       No      Yes
## 798            No           No           No           No       No       No
```

```
## 805           Yes           No          Yes          Yes       No      Yes
## 809           Yes           No           No           No       No       No
## 818            No           No           No           No       No       No
## 826            No           No           No           No       No       No
## 889            No           No           No           No       No       No
## 926            No           No           No           No       No       No
## 950            No           No           No           No       No       No
## 970            No           No           No           No       No       No
## 1036           No           No           No           No       No       No
## 1127           No           No           No           No       No       No
## 1284           No           No           No           No       No       No
## 1508           No           No           No           No       No       No
## 1531           No           No           No           No       No       No
## 1608           No           No           No           No       No       No
## 1627           No           No           No           No       No       No
## 1643           No           No           No           No       No       No
## 1669           No           No           No           No       No       No
## 1789           No           No           No           No       No       No
## 1812           No           No           No           No       No       No
## 1877           No           No           No           No       No      Yes
## 1880           No           No           No           No       No       No
## 1914          Yes           No          Yes           No       No      Yes
## 2029           No           No           No           No       No      Yes
## 2095           No           No           No           No       No      Yes
```

```
N_act_web_visits <- nrow(active_and_webvisits)
```

```
tot_active <- subset(Clientes, (as.numeric(MntFruits ==
    "High") + as.numeric(MntGoldProds ==
    "High") + as.numeric(MntSweetProducts ==
    "High") + as.numeric(MntFishProducts ==
    "High") + as.numeric(MntMeatProducts ==
    "High") + as.numeric(MntWines == "High")) >=
    4) %>%
    nrow()


active_few_web_visits <- subset(Clientes,
    NumWebVisitsMonth == "0-4" & (as.numeric(MntFruits ==
        "High") + as.numeric(MntGoldProds ==
        "High") + as.numeric(MntSweetProducts ==
        "High") + as.numeric(MntFishProducts ==
        "High") + as.numeric(MntMeatProducts ==
        "High") + as.numeric(MntWines ==
        "High")) >= 4) %>%
    nrow()

cat("clientes activos (inversión alta en al menos 4 productos:",
    tot_active, "\n")
```

```
## clientes activos (inversión alta en al menos 4 productos: 695
```

```
cat("% de clientes activos con muchas visitas a la web en
    el último mes (8-20): ",
    N_act_web_visits * 100/tot_active, "%\n")
```

```
## % de clientes activos con muchas visitas a la web en
##     el último mes (8-20):  5.323741 %
```

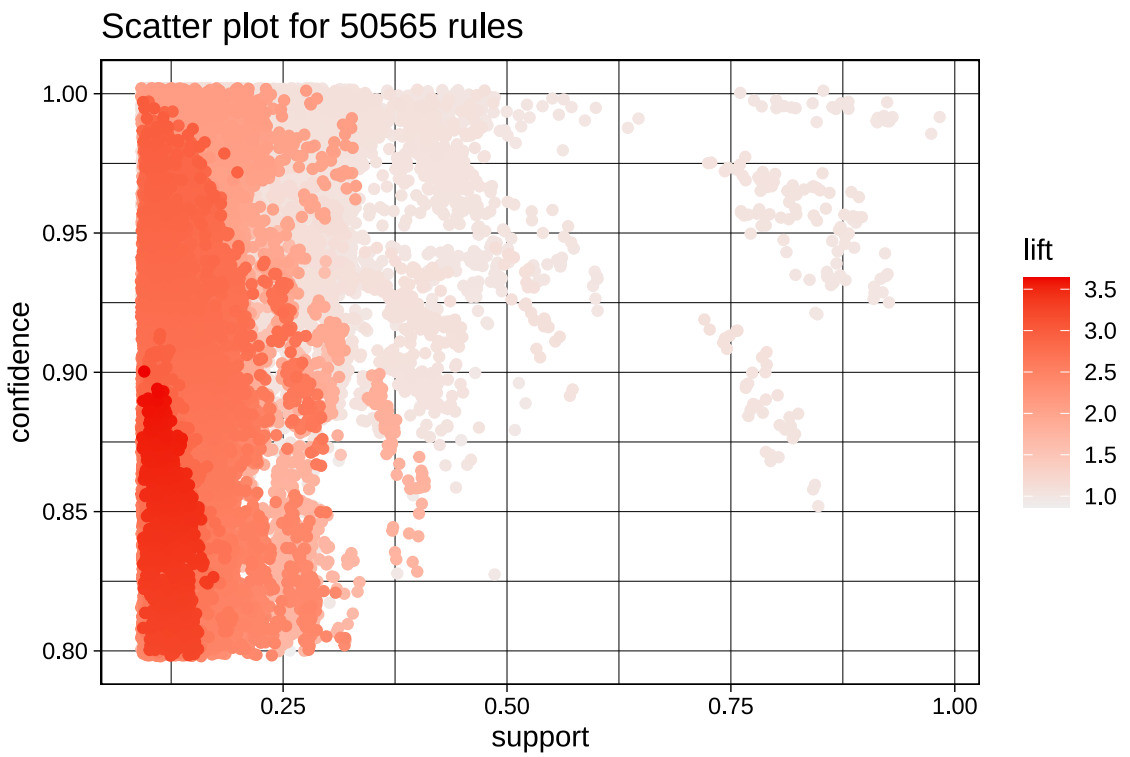
```
cat("% de clientes activos con pocas visitas
    a la web en el último mes (0-4): ",
    active_few_web_visits * 100/tot_active,
    "%")
```

```
## % de clientes activos con pocas visitas
##     a la web en el último mes (0-4):  69.78417 %
```

A continuación, se eliminan las reglas redundantes y se representan en función de lift, confianza y soporte:

```
rulesSorted = sort(rules, by = "confidence")
redundant <- is.redundant(x = rulesSorted,
    measure = "confidence")
rulesPruned <- rulesSorted[!redundant]
plot(rulesPruned)
```

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



Igual que antes, obtenemos reglas obvias si no filtramos por itemsets.

```
rules_mod <- subset(rulesPruned, subset = lift >
    2 & confidence < 0.85 & support > 0.2)
inspect(head(rules_mod, 10))
```

```
##      lhs                            rhs                       support confidence  coverage     lift co
## [1]  {MntMeatProducts=Low}       => {MntWines=Low}          0.2901786  0.8496732 0.3415179 2.551298
## [2]  {MntGoldProds=Low,
##       NumCatalogPurchases=0-1,
##       NumStorePurchases=0-4,
##       AcceptedCmp4=No,
##       Complain=No,
##       Response=No}               => {NumWebPurchases=0-2}  0.2093750  0.8496377 0.2464286 2.452562
## [3]  {NumCatalogPurchases=4-28,
##       AcceptedCmp3=No,
##       AcceptedCmp4=No}           => {MntMeatProducts=High} 0.2008929  0.8490566 0.2366071 2.546033
## [4]  {MntWines=Low,
##       NumStorePurchases=0-4,
##       Complain=No,
##       Response=No}               => {NumWebPurchases=0-2}  0.2531250  0.8488024 0.2982143 2.450151
## [5]  {MntGoldProds=Low,
##       NumCatalogPurchases=0-1,
##       NumStorePurchases=0-4,
##       AcceptedCmp4=No,
##       Response=No}               => {NumWebPurchases=0-2}  0.2125000  0.8484848 0.2504464 2.449235
## [6]  {MntGoldProds=Low,
##       NumCatalogPurchases=0-1,
##       NumStorePurchases=0-4}     => {MntWines=Low}         0.2223214  0.8483816 0.2620536 2.547419
## [7]  {MntMeatProducts=Low,
##       NumCatalogPurchases=0-1,
##       AcceptedCmp4=No,
##       Response=No}               => {NumWebPurchases=0-2}  0.2566964  0.8480826 0.3026786 2.448073
## [8]  {MntGoldProds=Low,
##       NumCatalogPurchases=0-1,
##       NumStorePurchases=0-4,
##       Complain=No,
##       Response=No}               => {NumWebPurchases=0-2}  0.2111607  0.8476703 0.2491071 2.446883
## [9]  {Income=Very_low,
##       NumStorePurchases=0-4}     => {MntMeatProducts=Low}  0.2008929  0.8474576 0.2370536 2.481445
## [10] {MntMeatProducts=Low,
##       NumCatalogPurchases=0-1,
##       NumStorePurchases=0-4,
##       AcceptedCmp4=No}           => {NumWebPurchases=0-2}  0.2700893  0.8473389 0.3187500 2.445927
```

4. Apriori sobre el dataset con las modificaciones adicionales.

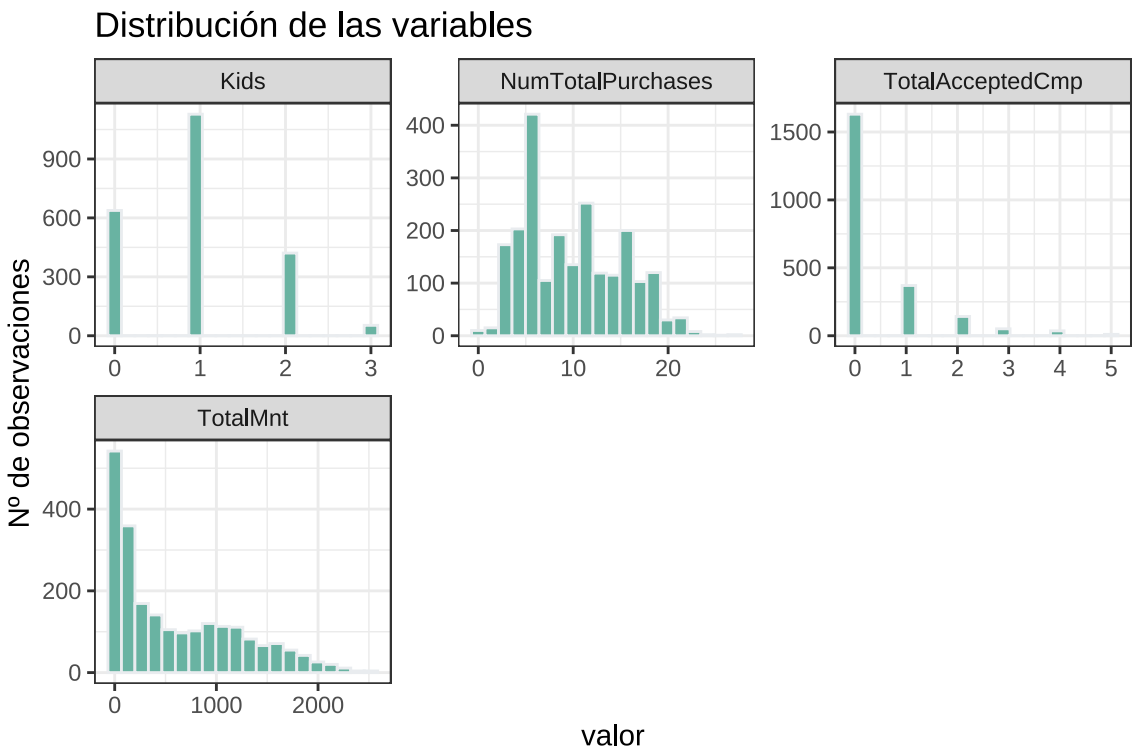
4.1. Modificaciones del dataset.

Al realizar la extracción de reglas sobre el dataset anterior se observa fácilmente que algunas variables pueden ser optimizadas para obtener reglas más interesantes.

A continuación se realizan algunos cambios (que aun así, no tienen por qué funcionar).

- 1. Se juntan las variables Kidhome y Teenhome en una sola (que representa la suma de las dos), Kids. (Se eliminan las variables originales).
- 2. Se crea una nueva variable, TotalMnt, que representa la suma de todas las cantidades invertidas en cada uno de los productos de la empresa (no se eliminan las variables originales).
- 3. Se crea una variable, TotalAcceptedCmp, con el número total de campañas aceptadas. (Se eliminan las variables originales).
- 4. Se crea una variable, NumTotalPurchases, con el número total de compras realizadas *(a través de la tienda o a través de la web, no se incluyen las compras utilizando el catálogo o utilizando descuentos, porque entiendo que no son incompatibles con comprar en tienda o en web).* (Se eliminan las variables originales).
- 5. La variable Education pasa a tener solo dos categorías: no licenciado y licenciado.

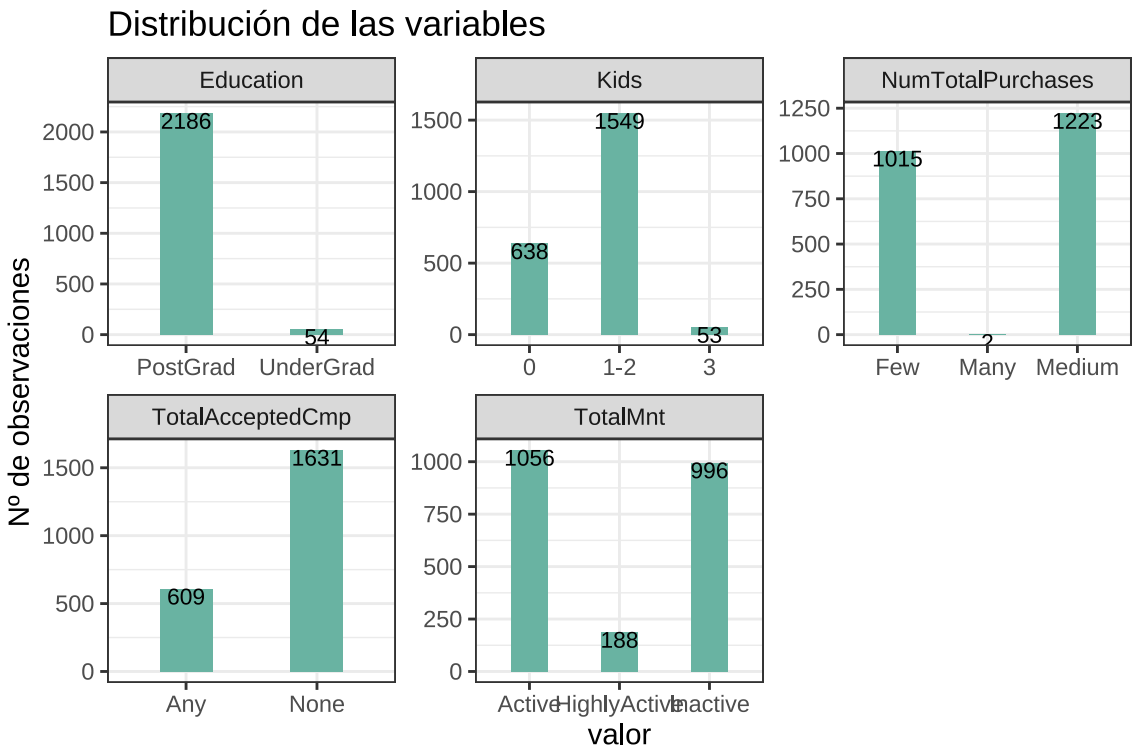
```
Clientes$Kids = Clientes_o$Kidhome + Clientes_o$Teenhome

Clientes$TotalMnt = Clientes_o$MntWines +
    Clientes_o$MntFruits + Clientes_o$MntMeatProducts +
    Clientes_o$MntFishProducts + Clientes_o$MntSweetProducts +
    Clientes_o$MntGoldProds

Clientes$TotalAcceptedCmp = Clientes_o$AcceptedCmp1 +
    Clientes_o$AcceptedCmp2 + Clientes_o$AcceptedCmp3 +
    Clientes_o$AcceptedCmp4 + Clientes_o$AcceptedCmp5 +
    Clientes_o$Response

Clientes$NumTotalPurchases = Clientes_o$NumWebPurchases +
    Clientes_o$NumStorePurchases

# Cambiamos la variable de estudios:
# postgraduados y pregraduados.

Clientes$Education <- as.character(Clientes$Education)
Clientes$Education[Clientes_o$Education ==
    "PhD" | Clientes_o$Education == "2n Cycle" |
    Clientes_o$Education == "Graduation" |
    Clientes_o$Education == "Master"] <- "PostGrad"

Clientes$Education[Clientes_o$Education ==
    "Basic"] <- "UnderGrad"

del_these <- c("AcceptedCmp1", "AcceptedCmp2",
    "AcceptedCmp3", "AcceptedCmp4", "AcceptedCmp5",
    "Response", "NumWebPurchases", "NumCatalogPurchases",
    "NumStorePurchases", "Kidhome", "Teenhome")
Clientes[del_these] = NULL


plot_histogram(Clientes[c("Kids", "TotalMnt",
    "TotalAcceptedCmp", "NumTotalPurchases")])
```



Discretización de las variables de nueva creación:

```
Clientes[["Kids"]] = ordered(cut(Clientes[["Kids"]],
    c(-Inf, 0, 2, Inf), labels = c("0", "1-2",
        "3"), right = T))

Clientes[["TotalMnt"]] = ordered(cut(Clientes[["TotalMnt"]],
    c(-Inf, 270, 1600, Inf), labels = c("Inactive",
        "Active", "HighlyActive"), right = T))

Clientes[["NumTotalPurchases"]] = ordered(cut(Clientes[["NumTotalPurchases"]],
    c(-Inf, 8, 25, Inf), labels = c("Few",
        "Medium", "Many"), right = T))

Clientes[["TotalAcceptedCmp"]] = ordered(cut(Clientes[["TotalAcceptedCmp"]],
    c(-Inf, 0, Inf), labels = c("None", "Any"),
    right = T))

# Para la representación, por alguna
# razón, necesito pasar las variables a
# caracteres.
Clientes$Education <- as.character(Clientes$Education)
Clientes$Kids <- as.character(Clientes$Kids)
Clientes$TotalMnt <- as.character(Clientes$TotalMnt)
Clientes$NumTotalPurchases <- as.character(Clientes$NumTotalPurchases)
Clientes$TotalAcceptedCmp <- as.character(Clientes$TotalAcceptedCmp)
```

```
Clientes %>%
    select(one_of(c("Kids", "TotalMnt", "Education",
        "NumTotalPurchases", "TotalAcceptedCmp"))) %>%
    pivot_longer(everything(), names_to = "cols",
        values_to = "value") %>%
    ggplot(aes(x = value)) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
    facet_wrap(~cols, ncol = 3, scales = "free",
        ) + labs(title = "Distribución de las variables",
    x = "valor", y = "Nº de observaciones")
```



A continuación, se introducen **items negativos**. En este caso, se utiliza la variable income para crear 4 variables binarias (una por cada posible valor de income).

```
Clientes$Income_VeryLow <- ifelse(Clientes$Income ==
    "Very_low", "TRUE", "FALSE")
Clientes$Income_Low <- ifelse(Clientes$Income ==
    "Low", "TRUE", "FALSE")
Clientes$Income_Medium <- ifelse(Clientes$Income ==
    "Medium", "TRUE", "FALSE")
Clientes$Income_High <- ifelse(Clientes$Income ==
    "High", "TRUE", "FALSE")

Clientes[["Income"]] = NULL
```

```
Clientes <- as.data.frame(unclass(Clientes),
    stringsAsFactors = TRUE)

head(Clientes)
```

```
##   Year_Birth Education Marital_Status Dt_Year Recency MntWines MntFruits
## 1     Mature  PostGrad         Single    2012  Medium     High      High
## 2     Senior  PostGrad         Single    2014    Some      Low       Low
## 3     Mature  PostGrad   Relationship    2013    Some     High      High
## 4      Adult  PostGrad   Relationship    2014    Some      Low    Medium
## 5      Adult  PostGrad   Relationship    2014    High   Medium      High
## 6     Mature  PostGrad   Relationship    2013     Few     High      High
##   MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds
## 1            High            High             High         High
## 2             Low             Low              Low          Low
## 3          Medium            High           Medium         High
## 4             Low          Medium              Low          Low
## 5          Medium            High             High       Medium
## 6          Medium             Low             High       Medium
##   NumDealsPurchases NumWebVisitsMonth Complain Kids     TotalMnt
## 1              3-15               5-7       No    0 HighlyActive
## 2               2-2               5-7       No  1-2     Inactive
## 3               0-1               0-4       No    0       Active
## 4               2-2               5-7       No  1-2     Inactive
## 5              3-15               5-7       No  1-2       Active
## 6               2-2               5-7       No  1-2       Active
##   TotalAcceptedCmp NumTotalPurchases Income_VeryLow Income_Low Income_Medium
## 1              Any            Medium          FALSE      FALSE          TRUE
## 2             None               Few          FALSE       TRUE         FALSE
## 3             None            Medium          FALSE      FALSE         FALSE
## 4             None               Few           TRUE      FALSE         FALSE
## 5             None            Medium          FALSE      FALSE          TRUE
## 6             None            Medium          FALSE      FALSE          TRUE
##   Income_High
## 1       FALSE
## 2       FALSE
## 3        TRUE
## 4       FALSE
## 5       FALSE
## 6       FALSE
```

4.2. Obtención de información de la base de datos.

```
ClientesBD2 <- as(Clientes, "transactions")
ClientesBD2
```

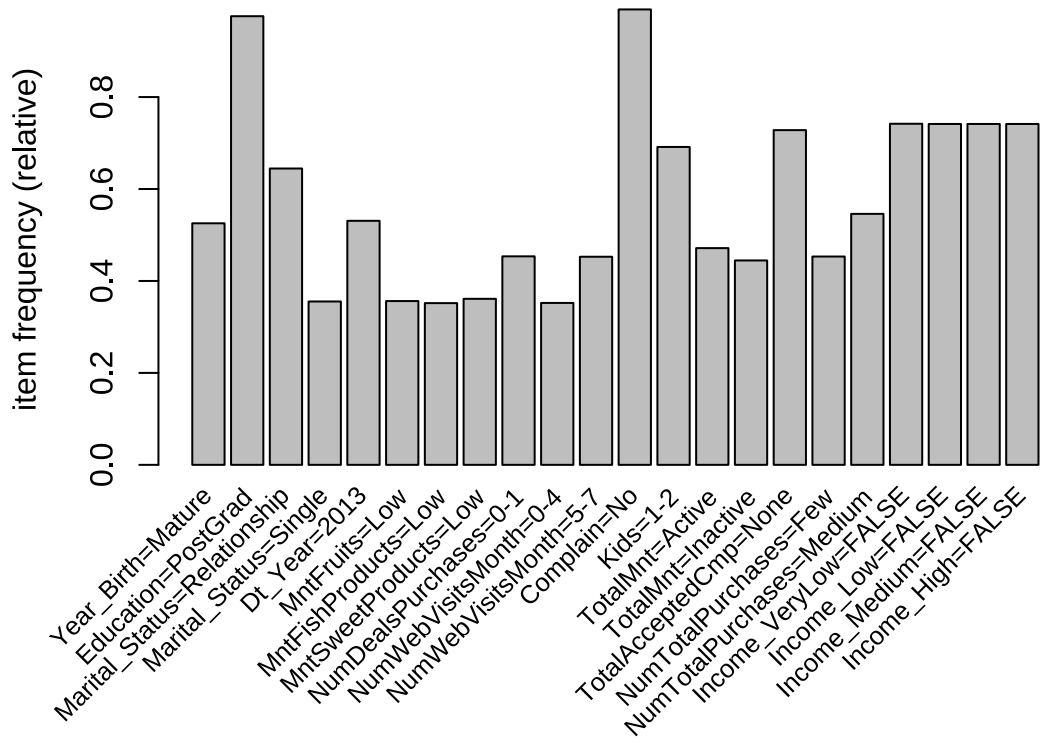
```
## transactions in sparse format with
##  2240 transactions (rows) and
##  60 items (columns)
```

```
summary(ClientesBD2)
```

```
## transactions as itemMatrix in sparse format with
##  2240 rows (elements/itemsets/transactions) and
##  60 columns (items) and a density of 0.3657143
##
## most frequent items:
##          Complain=No   Education=PostGrad Income_VeryLow=FALSE
##                 2219                 2186                 1662
##     Income_Low=FALSE  Income_Medium=FALSE              (Other)
##                 1661                 1661                39763
##
## element (itemset/transaction) length distribution:
## sizes
##   18   21   22
##   25   28 2187
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   22.00   22.00   21.94   22.00   22.00
##
## includes extended item information - examples:
##              labels  variables levels
## 1 Year_Birth=Senior Year_Birth Senior
## 2 Year_Birth=Mature Year_Birth Mature
## 3  Year_Birth=Adult Year_Birth  Adult
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

Items frecuentes (mínimo soporte = 0.35):

```
itemFrequencyPlot(ClientesBD2, support = 0.35,
    cex.names = 0.8)
```



Ya no existen los items tan frecuentes relativos al rechazo de la ofertas de cada campaña. Sin embargo, aparece un nuevo item muy frecuente: Education=PostGrad.

Complain=No sigue apareciendo también con la misma frecuencia que antes (muy alta),

Itemsets frecuentes:

```
iClientes <- sort(apriori(ClientesBD2, parameter = list(support = 0.35,
    target = "frequent")), by = "support")
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.35      1
##  maxlen            target  ext
##      10 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 784
```

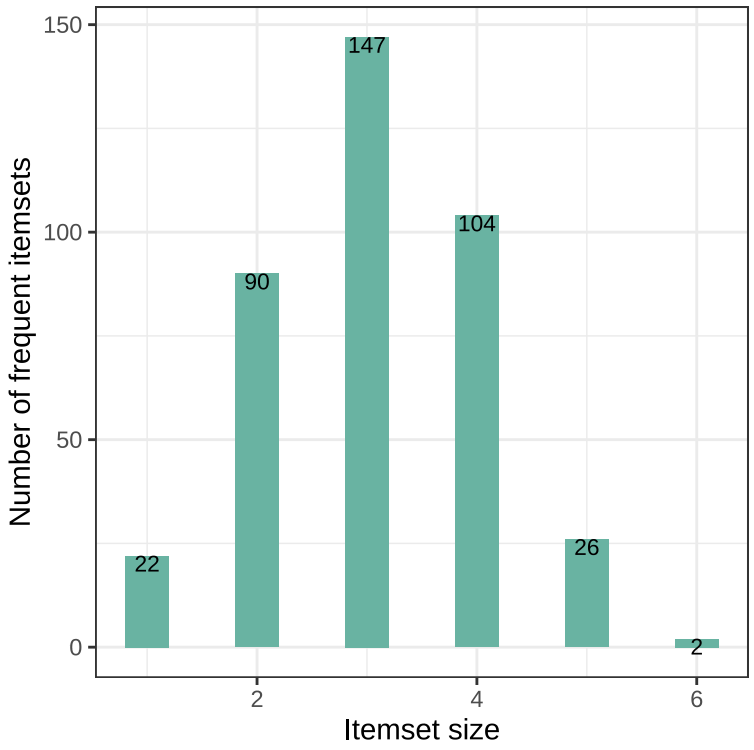
```
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[60 item(s), 2240 transaction(s)] done [0.01s].
## sorting and recoding items ... [22 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## sorting transactions ... done [0.00s].
## writing ... [391 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(iClientes, n = 10))
```

```
##      items                                       support   count
## [1]  {Complain=No}                               0.9906250 2219
## [2]  {Education=PostGrad}                        0.9758929 2186
## [3]  {Education=PostGrad, Complain=No}           0.9665179 2165
## [4]  {Income_VeryLow=FALSE}                      0.7419643 1662
## [5]  {Education=PostGrad, Income_VeryLow=FALSE} 0.7419643 1662
## [6]  {Income_High=FALSE}                         0.7415179 1661
## [7]  {Income_Low=FALSE}                          0.7415179 1661
## [8]  {Income_Medium=FALSE}                       0.7415179 1661
## [9]  {Complain=No, Income_Low=FALSE}             0.7361607 1649
## [10] {Complain=No, Income_VeryLow=FALSE}         0.7348214 1646
```

Número de itemsets frecuentes para cada tamaño de itemset.

```
as.data.frame(size(iClientes)) %>%
    ggplot(aes(x = size(iClientes))) + geom_bar(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "count",
    aes(label = ..count..), vjust = 1, cex = 3) +
    labs(x = "Itemset size", y = "Number of frequent itemsets")
```



Itemsets frecuentes de mayor tamaño (6; antes era 8):

```
inspect(iClientes[size(iClientes) == 6])
```

```
##     items                          support count
## [1] {Education=PostGrad,
##      Complain=No,
##      TotalMnt=Inactive,
##      NumTotalPurchases=Few,
##      Income_Medium=FALSE,
##      Income_High=FALSE}         0.3571429   800
## [2] {Education=PostGrad,
##      Complain=No,
##      TotalMnt=Active,
##      NumTotalPurchases=Medium,
##      Income_VeryLow=FALSE,
##      Income_Low=FALSE}          0.3553571   796
```

Itemsets maximales:

```
imaxClientes <- iClientes[is.maximal(iClientes)]
# length(imaxClientes)
inspect(head(sort(imaxClientes, by = "support")))
```

```
##     items                              support count
## [1] {Education=PostGrad,
##      Complain=No,
```

```
##      Income_VeryLow=FALSE,
##      Income_Medium=FALSE}          0.4897321  1097
## [2] {Education=PostGrad,
##      Marital_Status=Relationship,
##      Complain=No,
##      Income_VeryLow=FALSE}         0.4723214  1058
## [3] {Education=PostGrad,
##      Complain=No,
##      Kids=1-2,
##      TotalAcceptedCmp=None,
##      Income_High=FALSE}            0.4705357  1054
## [4] {Education=PostGrad,
##      Complain=No,
##      Income_Low=FALSE,
##      Income_Medium=FALSE}          0.4669643  1046
## [5] {Education=PostGrad,
##      Marital_Status=Relationship,
##      Complain=No,
##      Income_Low=FALSE}             0.4642857  1040
## [6] {Education=PostGrad,
##      Marital_Status=Relationship,
##      Complain=No,
##      Income_Medium=FALSE}          0.4558036  1021
```

```
# inspect(head(sort(imaxClientes,
# by='support',decreasing=F)))
```

Itemsets cerrados:

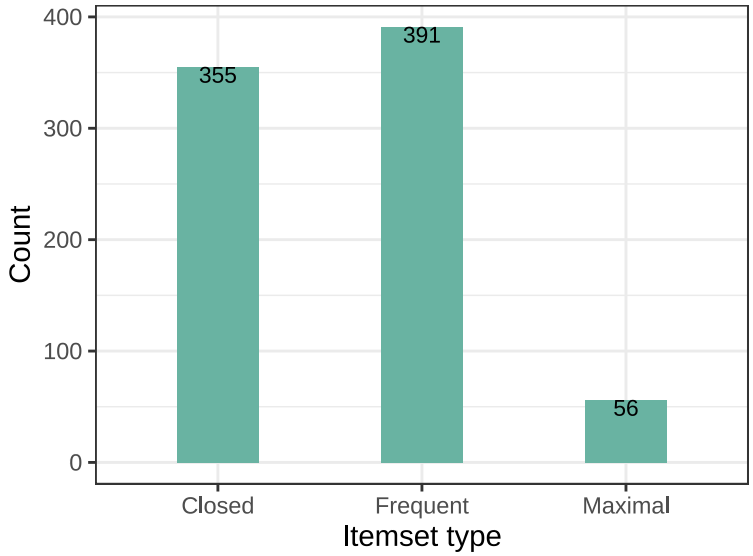
```
icloClientes <- iClientes[is.closed(iClientes)]
# length(icloClientes)
inspect(head(sort(icloClientes, by = "support")))
```

```
##     items                                       support   count
## [1] {Complain=No}                               0.9906250 2219
## [2] {Education=PostGrad}                        0.9758929 2186
## [3] {Education=PostGrad, Complain=No}           0.9665179 2165
## [4] {Education=PostGrad, Income_VeryLow=FALSE} 0.7419643 1662
## [5] {Income_High=FALSE}                         0.7415179 1661
## [6] {Income_Low=FALSE}                          0.7415179 1661
```

Número de itemsets de cada tipo:

```
# barplot(
# c(frequent=length(iClientes),
# closed=length(icloClientes),
# maximal=length(imaxClientes)),
# ylab='count', xlab='itemsets')

df <- data.frame(trt = c("Frequent", "Closed",
    "Maximal"), outcome = c(length(iClientes),
    length(icloClientes), length(imaxClientes)))
```

```
ggplot(df, aes(trt, outcome)) + geom_col(fill = "#69b3a2",
    width = 0.4) + geom_text(stat = "identity",
    aes(label = outcome), vjust = 1, cex = 3) +
    labs(x = "Itemset type", y = "Count")
```



El número de itemsets frecuentes ha disminuido de 1667 a 391; el número de cerrados de 1332 a 355 y el de maximales ha aumentado ligeramente.

4.3. Extracción de reglas (apriori).

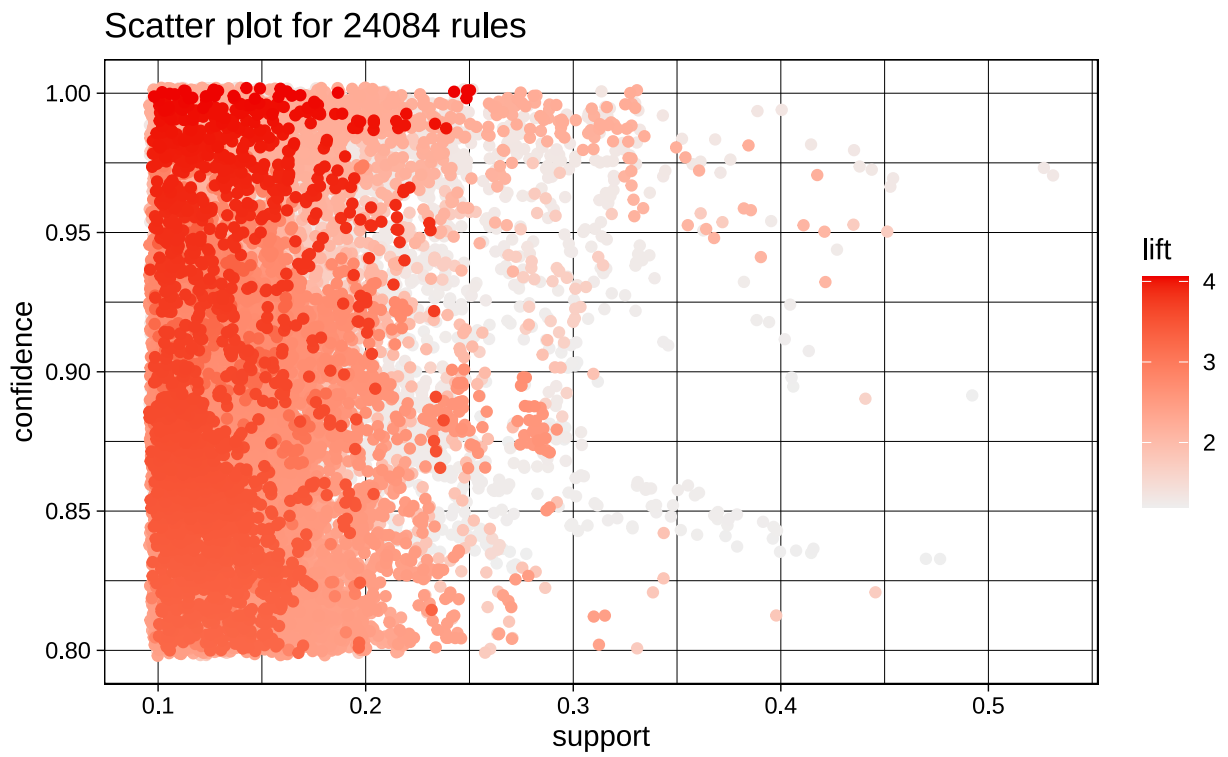
```
rules <- apriori(ClientesBD2, parameter = list(support = 0.1,
    confidence = 0.8, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      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: 224
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[60 item(s), 2240 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 9 10
```

```
## Warning in apriori(ClientesBD2, parameter = list(support = 0.1, confidence
## = 0.8, : Mining stopped (maxlen reached). Only patterns up to a length of 10
## returned!

##  done [0.72s].
## writing ... [547574 rule(s)] done [0.14s].
## creating S4 object  ... done [0.24s].
```

```
rules_lift <- subset(rules, subset = lift >
    1.2)

rulesSorted = sort(rules_lift, by = "confidence")
redundant <- is.redundant(x = rulesSorted,
    measure = "confidence")
rulesPruned <- rulesSorted[!redundant]  # remove redundant rules
plot(rulesPruned)
```

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



Se han obtenido 24084 reglas (tras eliminar las redundantes).

Se aplica algún filtro más (soporte menor de 0.5).

```
rules_filtered <- subset(rulesPruned, subset = lift >
    1.2 & confidence > 0.8 & support > 0.1 &
    support < 0.5)
length(rulesPruned)
```

```
## [1] 24084
```

```
length(rules_filtered)
```

```
## [1] 23721
```

```
# plot(rules_filtered,jitter=0)
```

A continuación, se observan algunas, las que más confianza tienen:

```
rules_filtered = sort(rules_filtered, by = "confidence")
inspect(head(rules_filtered, 20))
```

```
##      lhs                             rhs                       support confidence  coverage     lift co
## [1]  {Income_VeryLow=TRUE}      => {Income_High=FALSE}     0.2468750          1 0.2468750 1.348585
## [2]  {Income_VeryLow=TRUE}      => {Income_Low=FALSE}      0.2468750          1 0.2468750 1.348585
## [3]  {Income_VeryLow=TRUE}      => {Income_Medium=FALSE}   0.2468750          1 0.2468750 1.348585
## [4]  {Income_Medium=TRUE}       => {Income_High=FALSE}     0.2473214          1 0.2473214 1.348585
## [5]  {Income_Medium=TRUE}       => {Income_Low=FALSE}      0.2473214          1 0.2473214 1.348585
## [6]  {Income_Medium=TRUE}       => {Income_VeryLow=FALSE}  0.2473214          1 0.2473214 1.347774
## [7]  {Income_Low=TRUE}          => {Income_High=FALSE}     0.2473214          1 0.2473214 1.348585
## [8]  {Income_Low=TRUE}          => {Income_Medium=FALSE}   0.2473214          1 0.2473214 1.348585
## [9]  {Income_Low=TRUE}          => {Income_VeryLow=FALSE}  0.2473214          1 0.2473214 1.347774
## [10] {Income_High=TRUE}         => {Income_Low=FALSE}      0.2473214          1 0.2473214 1.348585
## [11] {Income_High=TRUE}         => {Income_Medium=FALSE}   0.2473214          1 0.2473214 1.348585
## [12] {Income_High=TRUE}         => {Income_VeryLow=FALSE}  0.2473214          1 0.2473214 1.347774
## [13] {Dt_Year=2014,
##       NumTotalPurchases=Medium} => {Income_VeryLow=FALSE}  0.1053571          1 0.1053571 1.347774
## [14] {MntWines=Low,
##       MntFishProducts=Medium}   => {NumTotalPurchases=Few} 0.1218750          1 0.1218750 2.206897
## [15] {MntWines=High,
##       Kids=1-2}                 => {Income_VeryLow=FALSE}  0.1660714          1 0.1660714 1.347774
## [16] {MntWines=High,
##       Income_High=FALSE}        => {Income_VeryLow=FALSE}  0.1437500          1 0.1437500 1.347774
## [17] {MntWines=High,
##       Income_Low=FALSE}         => {Income_VeryLow=FALSE}  0.3147321          1 0.3147321 1.347774
## [18] {MntWines=High,
##       Income_Medium=FALSE}      => {Income_VeryLow=FALSE}  0.2022321          1 0.2022321 1.347774
## [19] {MntWines=Low,
##       MntSweetProducts=Low}     => {NumTotalPurchases=Few} 0.2049107          1 0.2049107 2.206897
## [20] {MntWines=Low,
##       TotalMnt=Inactive}        => {NumTotalPurchases=Few} 0.3267857          1 0.3267857 2.206897
```

Se eliminan las reglas que contienen, en el antecedente un item TRUE relativo a cualquiera de las cuatro variables de Income acompañado de uno o varios items FALSE (relativos a Income también), por ser redundantes. Se eliminan, además, las reglas que contienen cualquier item relativo a income en antecedente y consecuente, por ser poco interesantes.

```
binars <- c("Income_VeryLow=TRUE", "Income_VeryLow=FALSE",
    "Income_Medium=TRUE", "Income_Medium=FALSE",
    "Income_Low=TRUE", "Income_Low=FALSE",
```

```
    "Income_High=TRUE", "Income_High=FALSE")

binarsT <- c("Income_VeryLow=TRUE", "Income_Medium=TRUE",
    "Income_Low=TRUE", "Income_High=TRUE")

binarsF <- c("Income_VeryLow=FALSE", "Income_Medium=FALSE",
    "Income_Low=FALSE", "Income_High=FALSE")

delete <- subset(rules_filtered, subset = (lhs %in%
    binarsT & lhs %in% binarsF) | (lhs %in%
    binars & rhs %in% binars))

inspect(head(delete, 5))
```

```
##     lhs                       rhs                     support   confidence
## [1] {Income_VeryLow=TRUE} => {Income_High=FALSE}   0.2468750 1
## [2] {Income_VeryLow=TRUE} => {Income_Low=FALSE}    0.2468750 1
## [3] {Income_VeryLow=TRUE} => {Income_Medium=FALSE} 0.2468750 1
## [4] {Income_Medium=TRUE}  => {Income_High=FALSE}   0.2473214 1
## [5] {Income_Medium=TRUE}  => {Income_Low=FALSE}    0.2473214 1
##     coverage  lift     count
## [1] 0.2468750 1.348585 553
## [2] 0.2468750 1.348585 553
## [3] 0.2468750 1.348585 553
## [4] 0.2473214 1.348585 554
## [5] 0.2473214 1.348585 554
```

```
# length(delete)

filt <- rules_filtered %!in% delete
filt2 <- ifelse(is.na(filt), TRUE, FALSE)
rules_opt <- rules_filtered[filt2]

length(rules_filtered) == (length(rules_opt) +
    length(delete))
```

```
## [1] TRUE
```

Se observan algunas de las reglas que quedan tras aplicar este último filtro:

```
inspect(head(rules_opt, 5))
```

```
##     lhs                           rhs                      support confidence  coverage     lift co
## [1] {Dt_Year=2014,
##      NumTotalPurchases=Medium} => {Income_VeryLow=FALSE}  0.1053571          1 0.1053571 1.347774   1
## [2] {MntWines=Low,
##      MntFishProducts=Medium}   => {NumTotalPurchases=Few} 0.1218750          1 0.1218750 2.206897   1
## [3] {MntWines=High,
##      Kids=1-2}                 => {Income_VeryLow=FALSE}  0.1660714          1 0.1660714 1.347774   2
## [4] {MntWines=Low,
##      MntSweetProducts=Low}     => {NumTotalPurchases=Few} 0.2049107          1 0.2049107 2.206897   4
## [5] {MntWines=Low,
##      TotalMnt=Inactive}        => {NumTotalPurchases=Few} 0.3267857          1 0.3267857 2.206897   7
```

```
# plot(rules_opt,jitter=0)
```

Se buscan reglas con un poco menos de confianza:

```
rules_mod <- subset(rules_opt, subset = confidence <
    0.95)
inspect(head(rules_mod, 5))
```

```
##     lhs                                rhs                           support confidence  coverage     l
## [1] {TotalMnt=Inactive,
##      TotalAcceptedCmp=None}        => {NumTotalPurchases=Few}    0.3642857  0.9499418 0.3834821 2.09
## [2] {MntFruits=High,
##      MntSweetProducts=High,
##      Income_Low=FALSE}             => {NumTotalPurchases=Medium} 0.2200893  0.9499037 0.2316964 1.73
## [3] {Marital_Status=Relationship,
##      MntFruits=High,
##      MntSweetProducts=High,
##      NumTotalPurchases=Medium}     => {Income_Low=FALSE}         0.1437500  0.9498525 0.1513393 1.28
## [4] {Marital_Status=Relationship,
##      NumDealsPurchases=3-15,
##      TotalMnt=Active,
##      Income_VeryLow=FALSE}         => {Kids=1-2}                 0.1183036  0.9498208 0.1245536 1.37
## [5] {Education=PostGrad,
##      Marital_Status=Relationship,
##      Kids=0,
##      Income_Low=FALSE,
##      Income_Medium=FALSE}          => {NumDealsPurchases=0-1}    0.1183036  0.9498208 0.1245536 2.09
```

La regla número 10 describe muy bien el grupo de clientes moderadamente activos en la empresa: aquellos que más utilizan los descuentos, que no tienen un ingreso demasiado alto ni demasiado bajo y cuyo número total de compras en los dos últimos años ha sido medio tienden a tener entre 1 y 2 hijos/adolescentes en casa. Se guarda esta regla como interesante.

Se guarda también la regla número 5: describe el prototipo de clientes que utiliza muy pocos o 0 descuentos.

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[10],
    "data.frame"))
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[5],
    "data.frame"))
R_INTERESANTES
```

```
##
## 80231                        {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurcl
## 96                                                                                  {Income=High} =
## 580                                                  {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} =>
## 16137                                          {Income=High,MntFruits=High,NumCatalogPurchases=4-28} =
## 4465                                                    {MntFishProducts=High,NumWebVisitsMonth=0-4}
## 54035                    {NumDealsPurchases=3-15,NumTotalPurchases=Medium,Income_VeryLow=FALSE,Income
## 138373 {Education=PostGrad,Marital_Status=Relationship,Kids=0,Income_Low=FALSE,Income_Medium=FALSE} =
##          support confidence  coverage     lift count
## 80231  0.1013393  0.9458333 0.1071429 1.638567   227
## 96     0.1995536  0.8068592 0.2473214 2.290703   447
```

```
## 580    0.1200893  0.8966667 0.1339286 1.854601   269
## 16137  0.1339286  0.8547009 0.1566964 2.426527   300
## 4465   0.1821429  0.8309572 0.2191964 2.491759   408
## 54035  0.1687500  0.9497487 0.1776786 1.373426   378
## 138373 0.1183036  0.9498208 0.1245536 2.094093   265
```

A continuación, se buscan reglas que incluyan en el antecedente algún item poco frecuente:

```
not_freq <- c("TotalAcceptedCmp=Any", "Marital_Status=Single",
    "Education=UnderGrad", "Complain=Yes",
    "Year_Birth=Young", "NumDealsPurchases=3-15")
rules_mod <- subset(rules_opt, subset = lhs %in%
    not_freq & confidence < 0.9)
inspect(head(rules_mod, 20))
```

```
##      lhs                            rhs                          support confidence  coverage     li
## [1]  {Marital_Status=Single,
##       Kids=1-2,
##       TotalAcceptedCmp=None}     => {Income_High=FALSE}        0.1629464  0.8990148 0.1812500 1.2123
## [2]  {NumDealsPurchases=3-15,
##       Income_VeryLow=FALSE,
##       Income_Low=FALSE}          => {TotalMnt=Active}          0.1321429  0.8969697 0.1473214 1.9026
## [3]  {Marital_Status=Single,
##       Complain=No,
##       Kids=1-2}                  => {Income_High=FALSE}        0.2058036  0.8968872 0.2294643 1.2095
## [4]  {Marital_Status=Single,
##       Kids=1-2}                  => {Income_High=FALSE}        0.2084821  0.8963532 0.2325893 1.2088
## [5]  {Marital_Status=Single,
##       MntMeatProducts=Low,
##       TotalMnt=Inactive}         => {TotalAcceptedCmp=None}    0.1026786  0.8949416 0.1147321 1.2291
## [6]  {Marital_Status=Single,
##       MntMeatProducts=Low,
##       NumTotalPurchases=Few}     => {TotalAcceptedCmp=None}    0.1026786  0.8949416 0.1147321 1.2291
## [7]  {Education=PostGrad,
##       MntGoldProds=High,
##       NumDealsPurchases=3-15}    => {NumTotalPurchases=Medium} 0.1058036  0.8943396 0.1183036 1.6380
## [8]  {NumDealsPurchases=0-1,
##       Kids=0,
##       TotalAcceptedCmp=Any}      => {Income_Medium=FALSE}      0.1044643  0.8931298 0.1169643 1.2044
## [9]  {NumDealsPurchases=0-1,
##       TotalAcceptedCmp=Any}      => {Income_Medium=FALSE}      0.1294643  0.8923077 0.1450893 1.2033
## [10] {NumDealsPurchases=0-1,
##       TotalAcceptedCmp=Any}      => {Income_VeryLow=FALSE}     0.1294643  0.8923077 0.1450893 1.2026
## [11] {NumDealsPurchases=0-1,
##       TotalAcceptedCmp=Any,
##       Income_VeryLow=FALSE,
##       Income_Medium=FALSE}       => {NumTotalPurchases=Medium} 0.1026786  0.8914729 0.1151786 1.6327
## [12] {TotalAcceptedCmp=Any,
##       NumTotalPurchases=Medium,
##       Income_High=TRUE}          => {MntWines=High}            0.1022321  0.8910506 0.1147321 2.6827
## [13] {TotalAcceptedCmp=Any,
##       NumTotalPurchases=Medium,
##       Income_VeryLow=FALSE,
```

```
##       Income_Low=FALSE,
##       Income_Medium=FALSE}       => {MntWines=High}            0.1022321  0.8910506 0.1147321 2.68273
## [14] {MntSweetProducts=High,
##       TotalAcceptedCmp=Any,
##       Income_VeryLow=FALSE,
##       Income_Low=FALSE}          => {MntMeatProducts=High}     0.1017857  0.8906250 0.1142857 2.67068
## [15] {NumWebVisitsMonth=0-4,
##       TotalAcceptedCmp=Any}      => {NumDealsPurchases=0-1}    0.1008929  0.8897638 0.1133929 1.96168
## [16] {NumDealsPurchases=0-1,
##       TotalAcceptedCmp=Any,
##       Income_VeryLow=FALSE}      => {NumTotalPurchases=Medium} 0.1151786  0.8896552 0.1294643 1.62945
## [17] {Marital_Status=Single,
##       MntMeatProducts=Low}       => {TotalAcceptedCmp=None}    0.1044643  0.8863636 0.1178571 1.21731
## [18] {MntGoldProds=High,
##       NumDealsPurchases=3-15}    => {NumTotalPurchases=Medium} 0.1058036  0.8843284 0.1196429 1.61970
## [19] {MntFruits=High,
##       TotalAcceptedCmp=Any,
##       Income_VeryLow=FALSE,
##       Income_Low=FALSE}          => {MntMeatProducts=High}     0.1013393  0.8832685 0.1147321 2.64861
## [20] {TotalAcceptedCmp=Any,
##       Income_High=TRUE}          => {MntWines=High}            0.1075893  0.8827839 0.1218750 2.65784
```

Regla número 15: el 89% de los que han visitado muy poco la web en el último mes y han aceptado alguna de las campañas ha realizado 0 o 1 compras con descuento en los últimos dos años. Se guarda esta regla como interesante:

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod[15],
    "data.frame"))
R_INTERESANTES
```

```
##
## 80231                        {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurcl
## 96                                                                                  {Income=High} =
## 580                                                  {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} =>
## 16137                                          {Income=High,MntFruits=High,NumCatalogPurchases=4-28} =
## 4465                                                 {MntFishProducts=High,NumWebVisitsMonth=0-4}
## 54035                    {NumDealsPurchases=3-15,NumTotalPurchases=Medium,Income_VeryLow=FALSE,Income
## 138373 {Education=PostGrad,Marital_Status=Relationship,Kids=0,Income_Low=FALSE,Income_Medium=FALSE} =
## 1380                                                {NumWebVisitsMonth=0-4,TotalAcceptedCmp=Any} =
##          support confidence  coverage     lift count
## 80231  0.1013393  0.9458333 0.1071429 1.638567   227
## 96     0.1995536  0.8068592 0.2473214 2.290703   447
## 580    0.1200893  0.8966667 0.1339286 1.854601   269
## 16137  0.1339286  0.8547009 0.1566964 2.426527   300
## 4465   0.1821429  0.8309572 0.2191964 2.491759   408
## 54035  0.1687500  0.9497487 0.1776786 1.373426   378
## 138373 0.1183036  0.9498208 0.1245536 2.094093   265
## 1380   0.1008929  0.8897638 0.1133929 1.961684   226
```

No se han escrito reglas que describan el tipo de clientes que han utilizado (con una probabilidad de por lo menos 0.8) algún descuento en los últimos dos años. Quizás si se hubieran juntado los items "NumDealsPurchases=2-2" y "NumDealsPurchases=3-15" en uno, sí se hubieran obtenido este tipo de reglas (por lo que sabemos del dataset, en el antecedente aparecerían una combinación de items como "Kids=1-2", "TotalMnt=Active", "Income_High=FALSE", etc.).

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    c("NumDealsPurchases=2-2", "NumDealsPurchases=3-15"))
inspect(head(rules_mod, 20))
```

A continuación se buscan reglas que describan el prototipo de cliente que no utiliza descuentos (aunque ya se ha guardado alguna regla de este tipo).

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    "NumDealsPurchases=0-1")
inspect(head(rules_mod, 5))
```

```
##     lhs                             rhs                       support confidence  coverage     lift co
## [1] {MntFruits=High,
##      MntMeatProducts=High,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_Medium=FALSE}       => {NumDealsPurchases=0-1} 0.1183036  0.9888060 0.1196429 2.180045
## [2] {MntFruits=High,
##      MntMeatProducts=High,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}          => {NumDealsPurchases=0-1} 0.1178571  0.9887640 0.1191964 2.179952
## [3] {MntFruits=High,
##      MntMeatProducts=High,
##      Kids=0,
##      NumTotalPurchases=Medium,
##      Income_High=TRUE}          => {NumDealsPurchases=0-1} 0.1156250  0.9885496 0.1169643 2.179479
## [4] {MntFruits=High,
##      MntMeatProducts=High,
##      Kids=0,
##      NumTotalPurchases=Medium,
##      Income_VeryLow=FALSE,
##      Income_Low=FALSE,
##      Income_Medium=FALSE}       => {NumDealsPurchases=0-1} 0.1156250  0.9885496 0.1169643 2.179479
## [5] {MntFruits=High,
##      MntSweetProducts=High,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_Medium=FALSE}       => {NumDealsPurchases=0-1} 0.1093750  0.9879032 0.1107143 2.178054
```

Se sustituye la penúltima regla del dataframe de interesantes por la segunda regla de las obtenidas en esta última celda, por ser más completa:

```
R_INTERESANTES[7, ] <- as(rules_mod[2], "data.frame")
R_INTERESANTES
```

```
##
## 80231              {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13
## 96                                                                          {Income=High} => {NumWeb
## 580                                        {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} => {NumCatal
## 16137                                {Income=High,MntFruits=High,NumCatalogPurchases=4-28} => {NumWeb
## 4465                                       {MntFishProducts=High,NumWebVisitsMonth=0-4} => {MntMe
```

```
## 54035          {NumDealsPurchases=3-15,NumTotalPurchases=Medium,Income_VeryLow=FALSE,Income_High=FALS
## 138373 {MntFruits=High,MntMeatProducts=High,NumWebVisitsMonth=0-4,Kids=0,Income_High=TRUE} => {NumDea
## 1380                                          {NumWebVisitsMonth=0-4,TotalAcceptedCmp=Any} => {NumDea
##          support confidence  coverage     lift count
## 80231  0.1013393  0.9458333 0.1071429 1.638567   227
## 96     0.1995536  0.8068592 0.2473214 2.290703   447
## 580    0.1200893  0.8966667 0.1339286 1.854601   269
## 16137  0.1339286  0.8547009 0.1566964 2.426527   300
## 4465   0.1821429  0.8309572 0.2191964 2.491759   408
## 54035  0.1687500  0.9497487 0.1776786 1.373426   378
## 138373 0.1178571  0.9887640 0.1191964 2.179952   264
## 1380   0.1008929  0.8897638 0.1133929 1.961684   226
```

Se ha comprobado que clientes activos, con grandes inversiones en varios productos, sin hijos y con un ingreso alto tienden a no usar descuentos. A continuación, se muestran los clientes del dataset que a pesar de cumplir todas estas características, sí usan los descuentos:

```
subset(Clientes, TotalMnt == "HighlyActive" &
    Income_High == "TRUE" & NumDealsPurchases ==
    "3-15")
```

```
##      Year_Birth Education Marital_Status Dt_Year Recency MntWines MntFruits
## 165      Mature  PostGrad   Relationship    2014    High      Low       Low
## 628      Mature  PostGrad   Relationship    2012    Some     High    Medium
## 688       Adult  PostGrad   Relationship    2012     Few   Medium    Medium
## 827      Senior  PostGrad   Relationship    2013    High     High    Medium
## 1127     Mature  PostGrad         Single    2012    High     High      High
## 1326     Mature  PostGrad   Relationship    2013     Few     High       Low
## 1444     Senior  PostGrad         Single    2012    Some     High    Medium
## 1513     Mature  PostGrad   Relationship    2012    High     High      High
## 1914     Mature  PostGrad   Relationship    2012    High     High      High
## 2057     Mature  PostGrad         Single    2012    High     High      High
##      MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds
## 165             High             Low              Low          Low
## 628             High            High             High       Medium
## 688             High          Medium              Low          Low
## 827             High          Medium              Low       Medium
## 1127            High            High             High         High
## 1326            High          Medium           Medium         High
## 1444            High            High             High         High
## 1513            High            High             High         High
## 1914            High            High             High         High
## 2057            High          Medium              Low       Medium
##      NumDealsPurchases NumWebVisitsMonth Complain Kids     TotalMnt
## 165               3-15               0-4       No  1-2 HighlyActive
## 628               3-15               5-7       No  1-2 HighlyActive
## 688               3-15               0-4       No    0 HighlyActive
## 827               3-15               5-7       No    3 HighlyActive
## 1127              3-15             8-20        No  1-2 HighlyActive
## 1326              3-15               5-7       No  1-2 HighlyActive
## 1444              3-15               0-4       No    0 HighlyActive
## 1513              3-15               5-7       No  1-2 HighlyActive
## 1914              3-15             8-20        No  1-2 HighlyActive
```

```
## 2057              3-15               5-7       No  1-2 HighlyActive
##      TotalAcceptedCmp NumTotalPurchases Income_VeryLow Income_Low Income_Medium
## 165              None               Few          FALSE      FALSE         FALSE
## 628               Any            Medium          FALSE      FALSE         FALSE
## 688              None               Few          FALSE      FALSE         FALSE
## 827               Any            Medium          FALSE      FALSE         FALSE
## 1127             None            Medium          FALSE      FALSE         FALSE
## 1326             None            Medium          FALSE      FALSE         FALSE
## 1444              Any            Medium          FALSE      FALSE         FALSE
## 1513             None            Medium          FALSE      FALSE         FALSE
## 1914              Any            Medium          FALSE      FALSE         FALSE
## 2057             None            Medium          FALSE      FALSE         FALSE
##      Income_High
## 165         TRUE
## 628         TRUE
## 688         TRUE
## 827         TRUE
## 1127        TRUE
## 1326        TRUE
## 1444        TRUE
## 1513        TRUE
## 1914        TRUE
## 2057        TRUE
```

Son solo 10, pero llama la atención que todos ellos gastan grandes cantidades en los productos de carne, y casi todos en vino también, mientras que para el resto de productos hay varios clientes cuya inversión no es alta. Quizás los descuentos que usan este tipo de clientes (adinerados) son descuentos aplicados sobre esos productos.

Se observa también que la mayoría tienen hijos. Aunque ya se había observado previamente que los que más descuentos usan, tienden a tener hijos, también se había observado la relación entre ser un cliente muy adinerado y activo y no tener hijos (este grupo de clientes son una excepcion a esta relación).

Se observa, además, que el número total de compras en estos clientes oscila entre pocas y un valor medio: clientes activos pueden invertir grandes cantidades de dinero en un bajo número de compras, aunque seguramente, se podría optimizar la discretización de la variable NumTotalPurchases, ya que parece muy complicado obtener un valor High en ella.

A continuación, se buscan aquellos clientes que, a pesar de tener bajo ingreso, son clientes activos de la empresa (en base a la cantidad de dinero total invertida en los dos últimos años).

```
subset(Clientes, (TotalMnt == "Active" |
    TotalMnt == "HighlyActive") & (Income_VeryLow ==
    "TRUE"))
```

```
##      Year_Birth Education Marital_Status Dt_Year Recency MntWines MntFruits
## 22         Adult  PostGrad   Relationship    2013    Some      Low       Low
## 383       Senior  PostGrad         Single    2012     Few      Low    Medium
## 404       Mature  PostGrad   Relationship    2013    Some   Medium      High
## 483       Mature  PostGrad         Single    2012    Some   Medium    Medium
## 620       Mature  PostGrad   Relationship    2012     Few   Medium    Medium
## 818       Senior  PostGrad         Single    2012  Medium   Medium       Low
## 926        Adult  PostGrad   Relationship    2012    High   Medium      High
## 1013       Adult  PostGrad         Single    2013  Medium   Medium    Medium
## 1036       Adult  PostGrad         Single    2012  Medium      Low      High
```

```
## 1049      Adult  PostGrad   Relationship    2012  Medium   Medium    Medium
## 1245      Adult  PostGrad   Relationship    2012    High   Medium    Medium
## 1285     Senior UnderGrad   Relationship    2012    High      Low      High
## 1329      Adult  PostGrad         Single    2013     Few   Medium    Medium
## 1430     Mature  PostGrad         Single    2013    High      Low      High
## 1455     Mature  PostGrad   Relationship    2012    Some   Medium    Medium
## 1716      Adult  PostGrad   Relationship    2012  Medium   Medium    Medium
## 1785     Mature  PostGrad   Relationship    2013     Few   Medium      High
## 1807     Mature  PostGrad         Single    2013    High   Medium    Medium
## 1869     Mature  PostGrad   Relationship    2012    Some   Medium       Low
## 1877     Mature  PostGrad         Single    2012     Few   Medium      High
## 1973     Mature  PostGrad   Relationship    2013     Few   Medium      High
## 1976     Mature  PostGrad   Relationship    2013    <NA>      Low    Medium
## 1977     Mature  PostGrad   Relationship    2012    Some   Medium      High
## 1999     Mature  PostGrad   Relationship    2013    Some   Medium       Low
## 2014      Adult UnderGrad   Relationship    2013    High   Medium      High
##      MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds
## 22              High             Low              Low          Low
## 383           Medium            High             High         High
## 404           Medium            High           Medium         High
## 483           Medium          Medium           Medium         High
## 620           Medium          Medium           Medium       Medium
## 818             High            High             High         High
## 926             High          Medium             High         High
## 1013          Medium             Low           Medium       Medium
## 1036          Medium            High             High         High
## 1049          Medium            High             High         High
## 1245          Medium          Medium           Medium         High
## 1285          Medium            High             High         High
## 1329          Medium             Low              Low         High
## 1430          Medium            High           Medium         High
## 1455            High            High           Medium         High
## 1716          Medium            High             High         High
## 1785          Medium          Medium              Low       Medium
## 1807          Medium          Medium              Low         High
## 1869          Medium          Medium              Low       Medium
## 1877            High            High             High         High
## 1973          Medium          Medium              Low       Medium
## 1976             Low             Low           Medium         High
## 1977          Medium            High           Medium         High
## 1999          Medium             Low           Medium       Medium
## 2014          Medium            High             High       Medium
##      NumDealsPurchases NumWebVisitsMonth Complain Kids     TotalMnt
## 22                3-15               0-4       No  1-2 HighlyActive
## 383               3-15               5-7       No    0       Active
## 404                0-1               5-7       No    0       Active
## 483                2-2              8-20       No  1-2       Active
## 620               3-15               0-4       No  1-2       Active
## 818               3-15              8-20       No    0       Active
## 926               3-15              8-20       No    0       Active
## 1013              3-15              8-20       No  1-2       Active
## 1036               2-2              8-20       No    0       Active
## 1049              3-15              8-20       No  1-2       Active
## 1245              3-15              8-20       No  1-2       Active
```

```
## 1285              2-2              5-7       No    0       Active
## 1329              0-1             8-20       No    0       Active
## 1430              0-1              5-7       No    0       Active
## 1455              2-2             8-20       No    0       Active
## 1716             3-15             8-20       No  1-2       Active
## 1785             3-15             8-20       No  1-2       Active
## 1807              0-1              0-4       No  1-2       Active
## 1869             3-15             8-20       No  1-2       Active
## 1877             3-15             8-20       No    0       Active
## 1973             3-15             8-20       No  1-2       Active
## 1976              0-1              0-4       No  1-2       Active
## 1977              2-2              5-7       No    0       Active
## 1999             3-15             8-20       No  1-2       Active
## 2014             3-15             8-20       No    0       Active
##      TotalAcceptedCmp NumTotalPurchases Income_VeryLow Income_Low Income_Medium
## 22               None               Few           TRUE      FALSE         FALSE
## 383              None            Medium           TRUE      FALSE         FALSE
## 404              None            Medium           TRUE      FALSE         FALSE
## 483               Any               Few           TRUE      FALSE         FALSE
## 620              None            Medium           TRUE      FALSE         FALSE
## 818              None            Medium           TRUE      FALSE         FALSE
## 926              None            Medium           TRUE      FALSE         FALSE
## 1013             None            Medium           TRUE      FALSE         FALSE
## 1036             None            Medium           TRUE      FALSE         FALSE
## 1049              Any            Medium           TRUE      FALSE         FALSE
## 1245             None            Medium           TRUE      FALSE         FALSE
## 1285             None            Medium           TRUE      FALSE         FALSE
## 1329             None               Few           TRUE      FALSE         FALSE
## 1430             None            Medium           TRUE      FALSE         FALSE
## 1455             None            Medium           TRUE      FALSE         FALSE
## 1716             None            Medium           TRUE      FALSE         FALSE
## 1785              Any            Medium           TRUE      FALSE         FALSE
## 1807             None            Medium           TRUE      FALSE         FALSE
## 1869             None            Medium           TRUE      FALSE         FALSE
## 1877              Any            Medium           TRUE      FALSE         FALSE
## 1973              Any            Medium           TRUE      FALSE         FALSE
## 1976             None            Medium           TRUE      FALSE         FALSE
## 1977             None            Medium           TRUE      FALSE         FALSE
## 1999              Any            Medium           TRUE      FALSE         FALSE
## 2014             None            Medium           TRUE      FALSE         FALSE
##      Income_High
## 22         FALSE
## 383        FALSE
## 404        FALSE
## 483        FALSE
## 620        FALSE
## 818        FALSE
## 926        FALSE
## 1013       FALSE
## 1036       FALSE
## 1049       FALSE
## 1245       FALSE
## 1285       FALSE
## 1329       FALSE
```

```
## 1430       FALSE
## 1455       FALSE
## 1716       FALSE
## 1785       FALSE
## 1807       FALSE
## 1869       FALSE
## 1877       FALSE
## 1973       FALSE
## 1976       FALSE
## 1977       FALSE
## 1999       FALSE
## 2014       FALSE
```

Casi todos ellos son clientes "Active" (solo uno es "HighlyActive"). De la interacción de estos clientes con la empresa se puede decir que visitan mucho la web y tienden a hacer gran uso de los descuentos.

A continuación, se buscan aquellos clientes que, a pesar de tener un alto ingreso, son clientes inactivos de la empresa:

```
subset(Clientes, TotalMnt == "Inactive" &
    Income_High == "TRUE")
```

```
##      Year_Birth Education Marital_Status Dt_Year Recency MntWines MntFruits
## 618       Adult  PostGrad   Relationship    2013    Some   Medium       Low
## 656      Mature  PostGrad         Single    2014    High      Low       Low
## 1301     Mature  PostGrad   Relationship    2013    Some      Low       Low
## 1354     Mature  PostGrad   Relationship    2014     Few   Medium    Medium
## 1619     Mature  PostGrad   Relationship    2014     Few   Medium    Medium
## 2133     Senior  PostGrad   Relationship    2013    High      Low       Low
## 2234      Adult  PostGrad   Relationship    2013     Few      Low    Medium
##      MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds
## 618              Low             Low              Low          Low
## 656              Low             Low              Low          Low
## 1301             Low             Low              Low          Low
## 1354          Medium            High           Medium          Low
## 1619          Medium            High           Medium          Low
## 2133             Low             Low              Low          Low
## 2234             Low          Medium              Low          Low
##      NumDealsPurchases NumWebVisitsMonth Complain Kids TotalMnt
## 618                0-1               0-4       No  1-2 Inactive
## 656                0-1               0-4       No    0 Inactive
## 1301               0-1               0-4       No  1-2 Inactive
## 1354               0-1               0-4       No  1-2 Inactive
## 1619               0-1               0-4       No  1-2 Inactive
## 2133               0-1               0-4       No    0 Inactive
## 2234             3-15                5-7       No  1-2 Inactive
##      TotalAcceptedCmp NumTotalPurchases Income_VeryLow Income_Low Income_Medium
## 618              None               Few          FALSE      FALSE         FALSE
## 656              None               Few          FALSE      FALSE         FALSE
## 1301             None               Few          FALSE      FALSE         FALSE
## 1354             None            Medium          FALSE      FALSE         FALSE
## 1619             None            Medium          FALSE      FALSE         FALSE
## 2133             None               Few          FALSE      FALSE         FALSE
## 2234             None               Few          FALSE      FALSE         FALSE
```

```
##      Income_High
## 618         TRUE
## 656         TRUE
## 1301        TRUE
## 1354        TRUE
## 1619        TRUE
## 2133        TRUE
## 2234        TRUE
```

Lo único a destacar: todos tienen una inversión baja en oro. El único que ha visitado la web en el último mes (además varias veces) ha utilizado varios descuentos en los dos últimos años.

Previamente se comprobó la relación entre invertir mucho dinero en un tipo de producto y hacerlo en el resto de productos también. Sin embargo, a continuación se buscan reglas que tengan en el consecuente un item que indique una gran inversión en algún tipo de producto de la empresa y que no tengan en el antecedente items que indiquen una gran inversión en otro/s tipo/s de productos. Se realiza esta búsqueda por si salieran reglas que definan características específicas de clientes que compran un producto específico (del tipo "los que se quejan compran mucho vino").

Vino

```
rules_mod_vino <- subset(rules_opt, subset = rhs %oin%
    "MntWines=High" & lhs %!ain% c("Income_Low=FALSE",
    "Income_VeryLow=FALSE") & lhs %!in% c(high,
    "NumTotalPurchases=Medium"))
inspect(head(rules_mod_vino, 5))
```

```
##     lhs                          rhs               support confidence  coverage     lift count
## [1] {TotalAcceptedCmp=Any,
##      Income_High=TRUE}       => {MntWines=High} 0.1075893  0.8827839 0.1218750 2.657844   241
## [2] {Kids=0,
##      TotalAcceptedCmp=Any,
##      Income_VeryLow=FALSE}   => {MntWines=High} 0.1022321  0.8450185 0.1209821 2.544142   229
## [3] {Education=PostGrad,
##      Kids=0,
##      TotalAcceptedCmp=Any}   => {MntWines=High} 0.1031250  0.8133803 0.1267857 2.448887   231
## [4] {Kids=0,
##      TotalAcceptedCmp=Any}   => {MntWines=High} 0.1031250  0.8105263 0.1272321 2.440294   231
## [5] {NumDealsPurchases=0-1,
##      TotalAcceptedCmp=Any,
##      Income_VeryLow=FALSE}   => {MntWines=High} 0.1040179  0.8034483 0.1294643 2.418984   233
```

Fruta

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    "MntFruits=High" & lhs %!ain% c("Income_Low=FALSE",
    "Income_VeryLow=FALSE") & lhs %!in% c(high,
    "NumTotalPurchases=Medium"))
inspect(head(rules_mod, 5))
```

```
##     lhs                          rhs                support confidence  coverage     lift count
## [1] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
```

```
##      Kids=0,
##      Income_High=TRUE}       => {MntFruits=High} 0.1281250  0.8176638 0.1566964 2.505564   287
## [2] {NumDealsPurchases=0-1,
##      Kids=0,
##      Income_High=TRUE}       => {MntFruits=High} 0.1370536  0.8164894 0.1678571 2.501965   307
## [3] {NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}       => {MntFruits=High} 0.1299107  0.8128492 0.1598214 2.490810   291
## [4] {Kids=0,
##      Income_High=TRUE}       => {MntFruits=High} 0.1392857  0.8103896 0.1718750 2.483273   312
## [5] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Income_High=TRUE}       => {MntFruits=High} 0.1379464  0.8046875 0.1714286 2.465800   309
```

Carne

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    "MntMeatProducts=High" & lhs %!ain% c("Income_Low=FALSE",
    "Income_VeryLow=FALSE") & lhs %!in% c(high,
    "NumTotalPurchases=Medium"))
inspect(head(rules_mod, 5))
```

```
##     lhs                                rhs                       support confidence  coverage     lift
## [1] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}             => {MntMeatProducts=High} 0.1424107  0.9088319 0.1566964 2.725279
## [2] {NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}             => {MntMeatProducts=High} 0.1450893  0.9078212 0.1598214 2.722248
## [3] {NumDealsPurchases=0-1,
##      Kids=0,
##      Income_High=TRUE}             => {MntMeatProducts=High} 0.1517857  0.9042553 0.1678571 2.711555
## [4] {Kids=0,
##      Income_High=TRUE}             => {MntMeatProducts=High} 0.1553571  0.9038961 0.1718750 2.710478
## [5] {Marital_Status=Relationship,
##      NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_Low=FALSE}             => {MntMeatProducts=High} 0.1004464  0.8858268 0.1133929 2.656294
```

Pescado

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    "MntFishProducts=High" & lhs %!ain% c("Income_Low=FALSE",
    "Income_VeryLow=FALSE") & lhs %!in% c(high,
    "NumTotalPurchases=Medium"))
inspect(head(rules_mod, 5))
```

```
##     lhs                          rhs                      support confidence  coverage     lift count
## [1] {Kids=0,
##      TotalMnt=Active,
```

```
##      Income_VeryLow=FALSE,
##      Income_Medium=FALSE}    => {MntFishProducts=High} 0.1017857  0.8382353 0.1214286 2.551151   228
## [2] {Kids=0,
##      TotalMnt=Active,
##      Income_Medium=FALSE}    => {MntFishProducts=High} 0.1062500  0.8380282 0.1267857 2.550521   238
## [3] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}       => {MntFishProducts=High} 0.1312500  0.8376068 0.1566964 2.549238   294
## [4] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Complain=No,
##      Kids=0,
##      TotalMnt=Active}        => {MntFishProducts=High} 0.1049107  0.8362989 0.1254464 2.545258   235
## [5] {NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}       => {MntFishProducts=High} 0.1334821  0.8351955 0.1598214 2.541899   299
```

Dulces

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    "MntSweetProducts=High" & lhs %!ain%
    c("Income_Low=FALSE", "Income_VeryLow=FALSE") &
    lhs %!in% c(high, "NumTotalPurchases=Medium"))
inspect(head(rules_mod, 5))
```

```
##     lhs                          rhs                       support confidence  coverage     lift coun
## [1] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}       => {MntSweetProducts=High} 0.1276786  0.8148148 0.1566964 2.476506   280
## [2] {NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_High=TRUE}       => {MntSweetProducts=High} 0.1299107  0.8128492 0.1598214 2.470532   291
## [3] {NumDealsPurchases=0-1,
##      Kids=0,
##      Income_High=TRUE}       => {MntSweetProducts=High} 0.1361607  0.8111702 0.1678571 2.465429   305
## [4] {Kids=0,
##      Income_High=TRUE}       => {MntSweetProducts=High} 0.1388393  0.8077922 0.1718750 2.455162   311
## [5] {NumDealsPurchases=0-1,
##      NumWebVisitsMonth=0-4,
##      Kids=0,
##      Income_VeryLow=FALSE,
##      Income_Medium=FALSE}    => {MntSweetProducts=High} 0.1312500  0.8032787 0.1633929 2.441444   294
```

Oro

```
rules_mod <- subset(rules_opt, subset = rhs %in%
    "MntGoldProds=High" & lhs %!ain% c("Income_Low=FALSE",
    "Income_VeryLow=FALSE") & lhs %!in% c(high,
    "NumTotalPurchases=Medium"))
inspect(head(rules_mod, 5))
```

En general no se observan reglas que ofrezcan información nueva: en el antecedente suelen aparecer los mismos items que se relacionan con los clientes activos (Kids=0, Income=High, NumDealsPurchases=0-1, NumWebVisitsMonth=0-4, ...). Sin embargo, llaman la atención dos cosas:

- 1. Solo en el caso de la gran inversión en oro no aparecen reglas (aplicando los mismos filtros que con los otros productos).
- 2. Solo en el caso de la gran inversión en vino aparece en el antecedente de las reglas el item "TotalAcceptedCmp=Any". Se comprueba que no se han escrito reglas con "TotalAcceptedCmp=Any" en el antecedente -puede ir acompañado de otros items, menos los que se han descartado también en el resto de casos- y en el consecuente un item de gran inversión en uno de los productos que no sea vino.

```
high_nowine <- c("MntFruits=High", "MntGoldProds=High",
    "MntSweetProducts=High", "MntFishProducts=High",
    "MntMeatProducts=High")

rules_mod <- subset(rules_opt, subset = rhs %in%
    high_nowine & length(rhs) == 1 & lhs %!ain%
    c("Income_Low=FALSE", "Income_VeryLow=FALSE") &
    lhs %!in% c(high, "NumTotalPurchases=Medium") &
    lhs %in% "TotalAcceptedCmp=Any")

inspect(head(rules_mod, 5))
```

Esto podría estar indicando que las campañas están funcionando para incrementar la compra de vino.

Se guarda una de estas reglas relativas al vino, por ejemplo, la primera, que indica que más del 88% de los clientes que tienen ingreso alto y que aceptaron la oferta de alguna campaña, invierten mucho en vino.

```
R_INTERESANTES <- rbind(R_INTERESANTES, as(rules_mod_vino[1],
    "data.frame"))
# R_INTERESANTES
```

```
R_INT_1 <- R_INTERESANTES[1:5, ]
R_INT_2 <- R_INTERESANTES[6:9, ]
```

```
cat("Reglas interesantes del dataset con las modificaciones estrictamente necesarias:")
```

```
## Reglas interesantes del dataset con las modificaciones estrictamente necesarias:
```

```
R_INT_1
```

```
##                                                                                                                  ru
## 80231 {Income=High,MntMeatProducts=High,NumCatalogPurchases=4-28,NumStorePurchases=8-13} => {Kidhome=
## 96                                                                          {Income=High} => {NumWebVisitsMonth=0
## 580                                  {NumStorePurchases=0-4,NumWebVisitsMonth=8-20} => {NumCatalogPurchases=0
## 16137                          {Income=High,MntFruits=High,NumCatalogPurchases=4-28} => {NumWebVisitsMonth=0
## 4465                                    {MntFishProducts=High,NumWebVisitsMonth=0-4} => {MntMeatProducts=Hig
##         support confidence  coverage     lift count
## 80231 0.1013393  0.9458333 0.1071429 1.638567   227
## 96    0.1995536  0.8068592 0.2473214 2.290703   447
```

```
## 580   0.1200893  0.8966667 0.1339286 1.854601   269
## 16137 0.1339286  0.8547009 0.1566964 2.426527   300
## 4465  0.1821429  0.8309572 0.2191964 2.491759   408
```

```
cat("Reglas interesantes del dataset con las modificaciones adicionales:")
```

```
## Reglas interesantes del dataset con las modificaciones adicionales:
```

```
R_INT_2
```

```
##
## 54035          {NumDealsPurchases=3-15,NumTotalPurchases=Medium,Income_VeryLow=FALSE,Income_High=FALS
## 138373 {MntFruits=High,MntMeatProducts=High,NumWebVisitsMonth=0-4,Kids=0,Income_High=TRUE} => {NumDea
## 1380                                          {NumWebVisitsMonth=0-4,TotalAcceptedCmp=Any} => {NumDea
## 972                                                  {TotalAcceptedCmp=Any,Income_High=TRUE} =>
##          support confidence  coverage     lift count
## 54035  0.1687500  0.9497487 0.1776786 1.373426   378
## 138373 0.1178571  0.9887640 0.1191964 2.179952   264
## 1380   0.1008929  0.8897638 0.1133929 1.961684   226
## 972    0.1075893  0.8827839 0.1218750 2.657844   241
```