

Format d'entrega

Aquest document s'ha realitzat mitjançant **Markdown**¹ amb l'ajuda de l'entorn de desenvolupament **RStudio**² utilitzant les característiques que aquest ofereix per a la creació de documents R reproduïbles.

La documentació generada en la realització de la pràctica es troba allotjada en **GitHub** al següent repositori:

- <https://github.com/rsanchezs/data-minig>

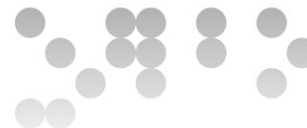
En aquest repositori es poden trobar els següents fitxers:

- Aquest document en formats **pdf** i **docx** amb el nom rsanchezs_PAC2.
- Un document **R Markdown**³ que es pot utilitzar per a reproduir tots els exemples presentats a la PAC.
- El conjunt de dades utilitzades.

¹ <https://es.wikipedia.org/wiki/Markdown>

² <https://www.rstudio.com/>

³ <http://archive.ics.uci.edu/ml/datasets/online+retail>



1. Nota: Propietat intel·lectual

Sovint és inevitable, al produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-lo en el marc d'una pràctica dels Estudis, sempre que això es documenti clarament i no suposi plagi en la pràctica.

Per tant, al presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en quin es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...).

L'estudiant haurà d'assegurar-se que la llicència no impedeix específicament el seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida per copyright.

Hauríeu de, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.

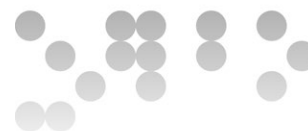
Exercici 1:

Creieu que les regles d'associació són el mètode més adequat per aconseguir els objectius que us havíeu proposat? Justifiqueu la resposta tot raonant-la. Com podria ser el model resultant? Doneu un exemple de la interpretació que es podria derivar del model generat

Exercici 2:

Pre-processament de les dades

En primer lloc, importarem el conjunt de dades amb `read_csv(path_to_file)`:



```
# Carreguem la llibreria que ens permet importar arxius CSV
if (!require("readr")) {
  # Instal·lació de la llibreria
  install.packages("readr")
# Carreguem la llibreria
library(readr)
}
# Importa el conjunt de dades a un dataframe
lastfm <- read_csv("data/lastfm.csv")
```

La funció `complete.cases(data)` retorna un vector de tipus lògic indicant-nos quines files no tenen valors desconeguts. Així, amb l'ajuda d'aquest vector filtrem les files del dataframe que no contenen valors desconeguts:

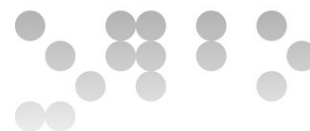
```
# Filtrem les observacions sense valors desconeguts
lastfm <- lastfm[complete.cases(lastfm), ]
dim(lastfm)

## [1] 289955      4
```

D'altra banda, amb el següent fragment de codi convertim a tipus categòric les variables `Sex` i `Country`:

```
# Carreguem ecosistema tidyverse
if (!require("tidyverse")) {
  # Instal·lació de la llibreria
  install.packages("tidyverse")
# Carreguem la llibreria
library(tidyverse)
}
lastfm <- lastfm %>%
  mutate(Sex = as.factor(lastfm$sex)) %>%
  mutate(Country = as.factor(lastfm$country))
```

Abans d'aplicar les regles d'associació ens caldrà convertir el conjunt de dades en transaccions amb la finalitat que tots els articles que es compren junts estiguin en una mateixa fila.



Per tant, ens caldrà agrupar les dades per `user`. Les següents línies de codi combinen tots els registres d'un usuari en una única fila:

```
library(plyr)
transactionData <- ddply(lastfm, c("user", "sex", "country"),
  function(df1) paste(df1$artist,
    collapse = ","))
```

Com que les columnes `user`, `sex` i `country` no les usarem en les regles d'associació les eliminem de `transactionData`:

```
# El·liminem la columna
transactionData$user <- NULL
# El·liminem la columna
transactionData$sex <- NULL
# El·liminem la columna
transactionData$country <- NULL
# Cambiem el nom de la variable a items
colnames(transactionData) <- c("items")
```

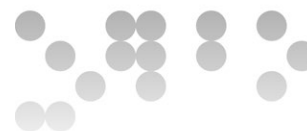
Aquest format per a dades transaccionals és conegut com a format *basket*⁴. A continuació, emmagatzemem aquestes dades en un arxiu CSV (Comma Separated Values):

```
write.csv(transactionData, file = "data/lastfm_transactions.csv",
  quote = FALSE, row.names = TRUE)
```

El següent fragment de codi llegeix l'arxiu `lastfm_transactions.csv` i l'emmagatzema en un objecte de la classe `transaction`:

```
if (!require("arules")) {
  # Instal·lació de la llibreria
  install.packages("arules")
  # Carreguem la llibreria
```

⁴ Un arxiu està en format *basket* quan cada fila representa una transacció i cada columna representa un article.



```
library(arules)
}
tr <- read.transactions("data/lastfm_transactions.csv",
                        format = "basket",
                        sep = ",")
```

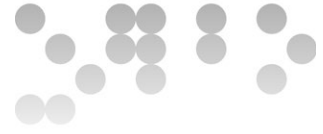
Visualitzem un resum de l'objecte `tr`:

```
# Visualitzem un resum de les transaccions
summary(tr)

## transactions as itemMatrix in sparse format with
## 15001 rows (elements/itemsets/transactions) and
## 16003 columns (items) and a density of 0.001270319
##
## most frequent items:
##           radiohead           the beatles           coldplay
##           2704             2668             2378
## red hot chili peppers           muse           (Other)
##           1786             1711           293707
##
## element (itemset/transaction) length distribution:
## sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
##  1 185 222 280 302 359 385 472 461 491 501 504 482 472 471 479 477 456
## 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
## 455 444 455 436 478 426 438 408 446 417 375 348 340 316 293 274 286 238
## 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 55
## 208 193 181 128 102 93 61 55 36 23 15 6 11 2 1 5 3 1
## 56 64 77
## 2 1 1
##
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   12.00   20.00   20.33   28.00   77.00
##
## includes extended item information - examples:
##                                labels
## 1 ...and you will know us by the trail of dead
## 2                                [unknown]
## 3                                1
```

Podem observar en la sortida la següent informació sobre les transaccions:

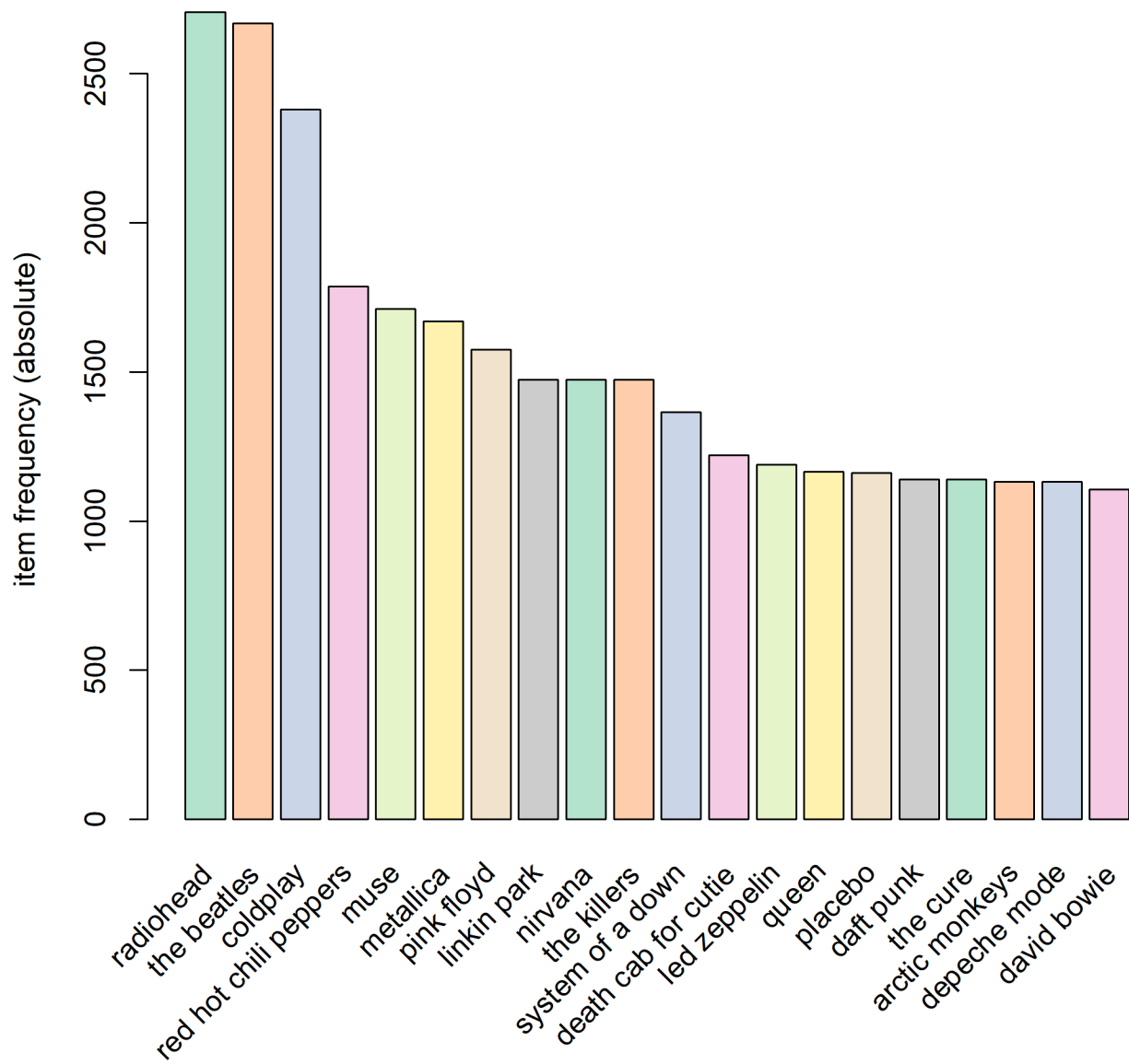
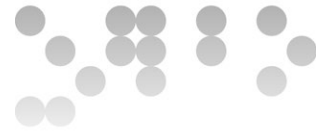
- S'han generat **15001 transaccions (files)** i **16003 articles (columnes)**.

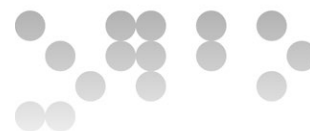


- Els articles més freqüents. Com per exemple, radiohead amb 2704 registres, beatles amb 2688, etc.

La representació gràfica, seria:

```
# Creació d'un gràfic de barres amb les freqüències absolutes
# per als top 20
if (!require("RColorBrewer")) {
  # Instal·lació de la llibreria
install.packages("RColorBrewer")
# Carreguem la llibreria
library(RColorBrewer)
}
itemFrequencyPlot(tr, topN=20, type="absolute",
  col=brewer.pal(8, 'Pastel2'))
```





Generació de les regles

En aquest apartat utilitzarem el algoritme apriori per a generar les regles d'associació. Per a trobar un conjunt de regles farem ús de la funció `apriori()` del paquet `arules`.

El prototip de la funció és el següent:

```
apriori(data, parameter = list(list(supp=0.001, conf=0.8, maxlen=10))
```

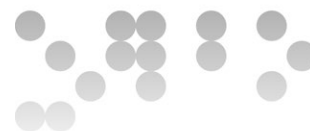
on els arguments són els següents:

- ****data:**** un objecte de tipus `transaction`.
- **parameter:** una llista especificant les mètriques i el màxim nombre d'elements:
- **supp:** el llindar de suport. Per defecte, **supp=0.001**.
- **conf:** el llindar de confiança. Per defecte, **conf=0.8**.
- **maxlen:** el màxim nombre d'elements. Per defecte, **maxlen=10**.

Com a mostra, la següent línia de codi calcula el conjunt de regles amb els valors per defecte:

```
# Executa algoritme a priori amb valors per defecte
rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8, maxlen=10))

## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##      0.8      0.1    1 none FALSE              TRUE      5   0.001      1
## maxlen target  ext
##      10 rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16003 item(s), 15001 transaction(s)] done [0.34s].
```

```
## sorting and recoding items ... [1004 item(s)] done [0.02s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 done [1.68s].
## writing ... [8952 rule(s)] done [0.10s].
## creating S4 object ... done [0.05s].
```

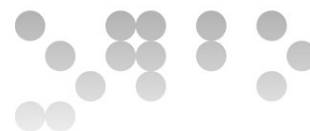
Tot seguit es mostra un resum del conjunt de regles:

```
# Visualitzem un resum
summary(rules)

## set of 8952 rules
##
## rule length distribution (lhs + rhs):sizes
##   3   4   5   6   7
## 246 4020 3807 840  39
##
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  3.000  4.000   5.000   4.599   5.000   7.000
##
## summary of quality measures:
##      support      confidence      lift      count
## Min.   :0.001067   Min.   :0.8000   Min.   : 4.438   Min.   :16.0
## 1st Qu.:0.001067   1st Qu.:0.8095   1st Qu.: 5.292   1st Qu.:16.0
## Median :0.001133   Median :0.8421   Median :10.176   Median :17.0
## Mean   :0.001260   Mean   :0.8497   Mean   :13.633   Mean   :18.9
## 3rd Qu.:0.001333   3rd Qu.:0.8824   3rd Qu.:17.015   3rd Qu.:20.0
## Max.   :0.003933   Max.   :1.0000   Max.   :123.847   Max.   :59.0
##
## mining info:
## data ntransactions support confidence
## tr      15001      0.001      0.8
```

Podem observar en la sortida la següent informació sobre els conjunt de regles:

- **Parameter Specification:** on $\text{min_sup}=0.001$ i $\text{min_confidence}=0.8$ amb 10 articles com a màxim en una regla.
- **Total number of rules:** en aquest cas 8952 regles.
- **Distribution of rule length:** Una longitud de 4 articles té la majoria de regles i la longitud 7 té el nombre més baix de regles.



- **Summary of Quality measures:** valors màxims i mínims per a les mètriques de suport, confiança i millora.
- ****Mining info:**** les dades, suport, confiança i nombre de transaccions.

Establiment dels llimars de suport i confiança

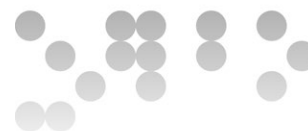
Després de provar diversos valors per a les mètriques, trobem un conjunt de regles amb un nivell de suport mínim del 3% i una confiança mínima del 80%:

```
# Executem el algoritme a priori amb
# min_supp = 3% i min_conf = 80%
rules <- apriori(tr, parameter = list(supp = 0.003, conf = 0.80))

## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##      0.8      0.1    1 none FALSE              TRUE        5   0.003      1
## maxlen target  ext
##      10  rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 45
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16003 item(s), 15001 transaction(s)] done [0.32s].
## sorting and recoding items ... [1004 item(s)] done [0.00s].
## creating transaction tree ... done [0.02s].
## checking subsets of size 1 2 3 4 5 done [0.23s].
## writing ... [18 rule(s)] done [0.02s].
## creating S4 object ... done [0.00s].
```

Tot seguit es mostra un resum d'executar l'algoritme apriori:

```
# Visualizem un resum
summary(rules)
```

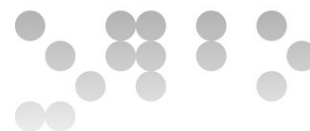


```
## set of 18 rules
##
## rule length distribution (lhs + rhs):sizes
## 3 4
## 5 13
##
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 3.000  3.250  4.000  3.722  4.000  4.000
##
## summary of quality measures:
##      support      confidence      lift      count
## Min.   :0.003066   Min.   :0.8000   Min.   : 4.477   Min.   :46.00
## 1st Qu.:0.003133   1st Qu.:0.8070   1st Qu.: 4.818   1st Qu.:47.00
## Median :0.003266   Median :0.8210   Median : 5.101   Median :49.00
## Mean   :0.003326   Mean   :0.8231   Mean   :11.381   Mean   :49.89
## 3rd Qu.:0.003466   3rd Qu.:0.8372   3rd Qu.: 7.681   3rd Qu.:52.00
## Max.   :0.003933   Max.   :0.8596   Max.   :49.763   Max.   :59.00
##
## mining info:
## data ntransactions support confidence
## tr          15001    0.003         0.8
```

Com que hi ha només 18 regles les visualitzem per pantalla:

```
# Visualitzem les 10 primeres regles d'associació
inspect(rules)
```

	lhs	rhs	support	confidence	lift	count
[1]	{above & beyond, atb}	=> {armin van buuren}	0.003133124	0.8392857	49.763340	47
[2]	{atb, ferry corsten}	=> {armin van buuren}	0.003199787	0.8135593	48.237958	48
[3]	{autechre, squarepusher}	=> {aphex twin}	0.003333111	0.8064516	21.002744	50
[4]	{björk, tricky}	=> {massive attack}	0.003266449	0.8448276	13.173866	49
[5]	{james blunt, keane}	=> {coldplay}	0.003933071	0.8309859	5.242060	59
[6]	{broken social scene, modest mouse, the beatles}	=> {radiohead}	0.003066462	0.8070175	4.477097	46
[7]	{broken social scene, death cab for cutie, the beatles}	=> {radiohead}	0.003533098	0.8548387	4.742395	53
[8]	{kaiser chiefs, keane, the killers}	=> {coldplay}	0.003066462	0.8070175	5.090862	46
[9]	{franz ferdinand, kaiser chiefs, the strokes}	=> {the killers}	0.003533098	0.8281250	8.433607	53
[10]	{keane,					



```
##      oasis,
##      snow patrol}      => {coldplay}      0.003066462  0.8070175  5.090862  46
## [11] {keane,
##      oasis,
##      the killers}      => {coldplay}      0.003799747  0.8260870  5.211157  57
## [12] {arctic monkeys,
##      keane,
##      the killers}      => {coldplay}      0.003466436  0.8000000  5.046594  52
## [13] {franz ferdinand,
##      oasis,
##      the beatles}      => {coldplay}      0.003199787  0.8000000  5.046594  48
## [14] {bloc party,
##      oasis,
##      the killers}      => {coldplay}      0.003133124  0.8103448  5.111851  47
## [15] {death cab for cutie,
##      oasis,
##      the killers}      => {coldplay}      0.003266449  0.8596491  5.422875  49
## [16] {beck,
##      the beatles,
##      the smashing pumpkins} => {radiohead} 0.003266449  0.8166667  4.530627  49
## [17] {sigur rós,
##      the cure,
##      the smashing pumpkins} => {radiohead} 0.003133124  0.8392857  4.656111  47
## [18] {nirvana,
##      placebo,
##      the smashing pumpkins} => {radiohead} 0.003466436  0.8253968  4.579060  52
```

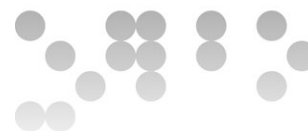
Per exemple, podem observar en la sortida les següents regles:

- 83% dels usuaris que escolten James Blunt i Keane també escolten Cold Play.
- 82% dels clients que escolten Keane, Oasis i The Killers també escolten Cold Play.

Cerca de regles segons conseqüent

A tall d'exemple, supossem que necessitem trobar les regles d'associació per a un determinat artista. Podem fer ús del paràmetre `appearance` de la funció `apriori()`, establint un o diversos antecedents i un conseqüent amb LHS (IF part) i RHS (THEN part):

Per exemple, per a respondre a la pregunta “Que artistes van escoltar els usuaris abans de escoltar Radiohead” ho fariem com es mostra a continuació:



```
rules <- apriori(tr, parameter = list(supp=0.003, conf=0.8),
               appearance = list(default="lhs",rhs="radiohead"))

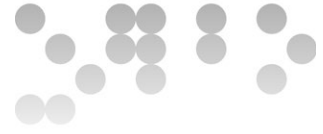
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##          0.8    0.1    1 none FALSE                TRUE         5   0.003     1
## maxlen target  ext
##      10  rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 45
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[16003 item(s), 15001 transaction(s)] done [0.27s].
## sorting and recoding items ... [1004 item(s)] done [0.01s].
## creating transaction tree ... done [0.02s].
## checking subsets of size 1 2 3 4 5 done [0.20s].
## writing ... [5 rule(s)] done [0.03s].
## creating S4 object ... done [0.00s].

inspect(head(rules))
```

	lhs	rhs	support	confidence	lift	count
[1]	{broken social scene, modest mouse, the beatles}	=> {radiohead}	0.003066462	0.8070175	4.477097	46
[2]	{broken social scene, death cab for cutie, the beatles}	=> {radiohead}	0.003533098	0.8548387	4.742395	53
[3]	{beck, the beatles, the smashing pumpkins}	=> {radiohead}	0.003266449	0.8166667	4.530627	49
[4]	{sigur rós, the cure, the smashing pumpkins}	=> {radiohead}	0.003133124	0.8392857	4.656111	47
[5]	{nirvana, placebo, the smashing pumpkins}	=> {radiohead}	0.003466436	0.8253968	4.579060	52

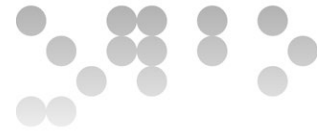
Visualització de les regles

Podem visualitzar les regles d'associació amb `plot()` del paquet `arulesViz`. Utilitza la mètrica de suport en l'eix d'ordenada i la confiança en l'eix d'abscisses. A més, la mètrica de millora (lift) es usada per colorejar els punts.

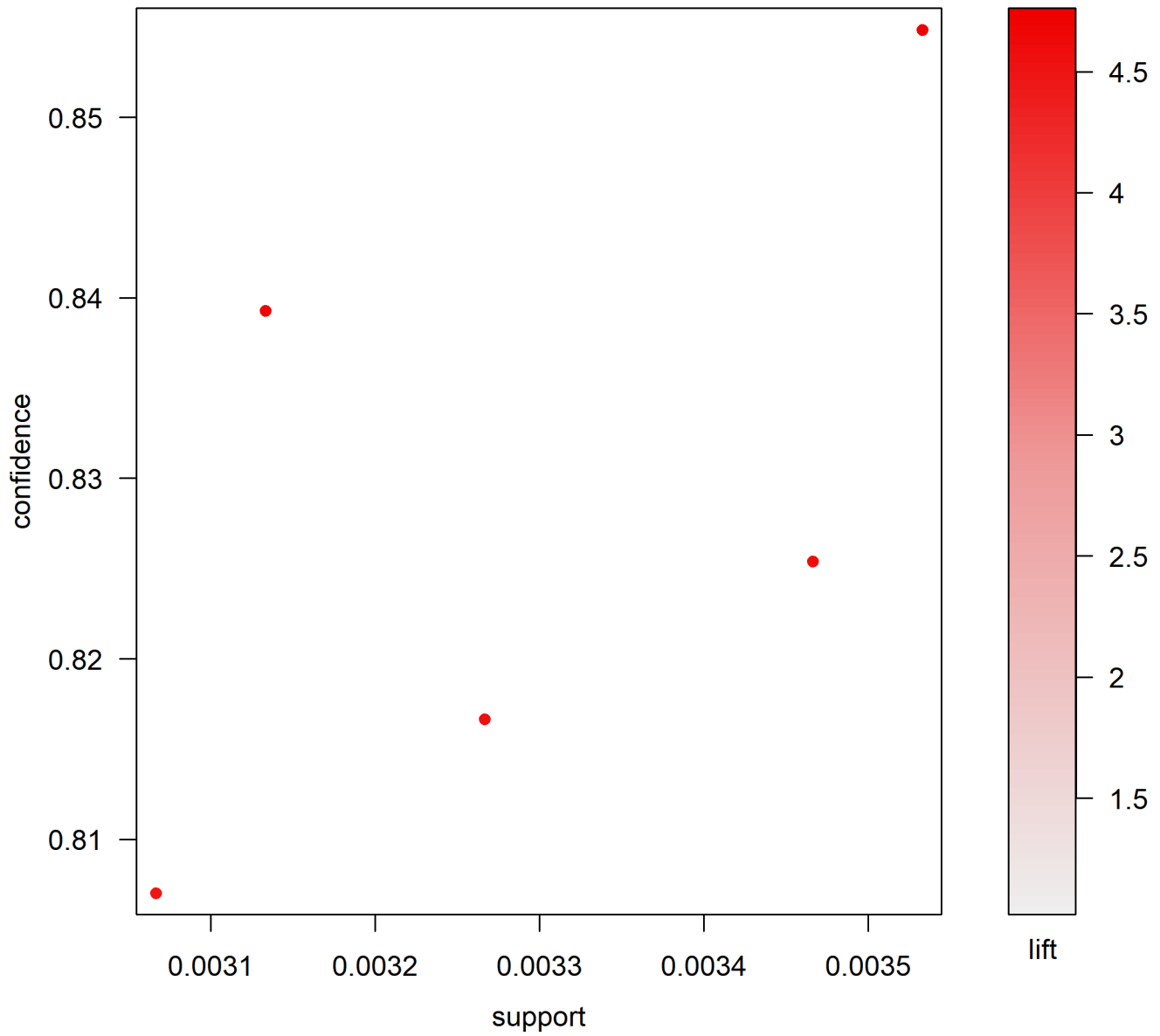


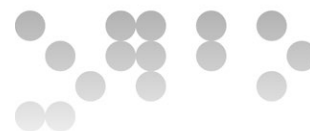
Per exemple, podem visualitzar el conjunt de regles amb un llindar de confiança del 80% com es mostra en el següent fragment de codi:

```
if (!require("arulesViz")) {  
  # Instal·lació de la llibreria  
install.packages("arulesViz")  
# Carreguem la llibreria  
library(arulesViz)  
}  
  
# Filtra les regles amb min_conf > 0.95  
subRules <- rules[quality(rules)$confidence>0.80]  
# Diagrama de dispersió amb regles associació amb min_conf>0.80  
plot(subRules, jitter=0)
```



Scatter plot for 5 rules





Exercici 3:

Pre-processament de les dades

En aquest exercici, farem ús del conjunt de dades [Online-Retail](http://archive.ics.uci.edu/ml/datasets/online+retail) ⁵ del repositori UCI Machine Learning Repository ⁶.

En primer lloc, importarem el conjunt de dades amb `read_excel(path_to_file)`:

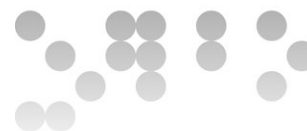
```
# Carreguem la llibreria que ens permet importar arxius excel
if (!require("readxl")) {
  # Instal·lació de la llibreria
  install.packages("readxl")
# Carreguem la llibreria
library(readxl)
}
# Importa el conjunt de dades a un dataframe
retail <- read_excel(path = "data/Online Retail.xlsx")
str(retail)

## Classes 'tbl_df', 'tbl' and 'data.frame':   541909 obs. of  8 variables:
## $ InvoiceNo   : chr  "536365" "536365" "536365" "536365" ...
## $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
## $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID
HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
## $ Quantity   : num  6 6 8 6 6 2 6 6 6 32 ...
## $ InvoiceDate: POSIXct, format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" ...
## $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
## $ CustomerID: num  17850 17850 17850 17850 17850 ...
## $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

La funció `complete.cases(data)` retorna un vector de tipus lògic indicant-nos quines files no tenen valors desconeguts. Així, amb l'ajuda d'aquest vector filtrem les files del dataframe:

⁵ <http://archive.ics.uci.edu/ml/datasets/online+retail>

⁶ <http://archive.ics.uci.edu/ml/index.php>



```
# Filtrem les observacions sense valors desconeguts
retail <- retail[complete.cases(retail), ]
dim(retail)

## [1] 406829      8
```

D'altra banda, amb la següent línia de codi convertim a tipus categòric les variables `Description` i `Country`:

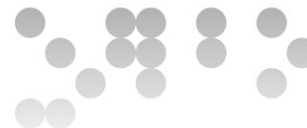
```
# Carreguem ecosistema tidyverse
if (!require("tidyverse")) {
  # Instal·lació de la llibreria
  install.packages("tidyverse")
# Carreguem la llibreria
library(tidyverse)
}
retail <- retail %>%
  mutate(Description = as.factor(retail$Description)) %>%
  mutate(Country = as.factor(retail$Country))
```

A continuació, separem la data i l'hora de la variable `InvoiceDate` i les emmagatzemem a les variables `dateInvoice` i `timeInvoice` respectivament:

```
# Emmagatzema la data en la variable `dateInvoice`
dateInvoice <- as.Date(retail$InvoiceDate)
# Emmagatzema la hora en la variable `timeInvoice`
timeInvoice <- format(retail$InvoiceDate, "%H:%M:%S")
# Afegim les noves variables al dataframe
retail <- cbind(retail, dateInvoice)
retail <- cbind(retail, timeInvoice)
```

Per últim, convertim la variable `InvoiceNo` de tipus `character` a `numeric`:

```
# Convertim de character a numeric variable InvoiceNo
InvoiceNo <- as.numeric(as.character(retail$InvoiceNo))
# Afegim la variable al dataframe
retail <- cbind(retail, InvoiceNo)
```



Abans d'aplicar les regles d'associació ens caldrà convertir el conjunt de dades en transaccions amb la finalitat que tots els articles que es compren junts estiguin en una mateixa fila.

Per tant, ens caldrà agrupar les dades o bé per `CustomerID` o bé per `CustomerID` i `Date`; o també podem agrupar els articles per `InvoiceNo` i `Date`.

Les següents línies de codi combinen tots els articles de una `InvoiceNo` i `date` en una fila i separen els elements amb una coma:

```
# Agrupem articles per `InvoiceNo` i `dateInvoice`
library(plyr)
transactionData <- ddply(retail, c("InvoiceNo", "dateInvoice"),
  function(df1) paste(df1$Description,
    collapse = ","))
```

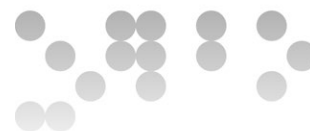
Com que les columnes `InvoiceNo` i `dateInvoice` no les usarem en les regles d'associació les eliminem de `transactionData`:

```
# El·liminem la columna
transactionData$InvoiceNo <- NULL
# El·liminem la columna
transactionData$dateInvoice <- NULL
# Cambiem el nom de la variable a items
colnames(transactionData) <- c("items")
```

Aquest format per a dades transaccionals és conegut com a format **basket**^[7]. A continuació, emmagatzemem aquestes dades en un arxiu CSV (Comma Separated Values):

```
# Emmagatzemem transaccions en un arxiu CSV
write.csv(transactionData, file = "data/market_basket_transactions.csv",
  quote = FALSE, row.names = TRUE)
```

El següent fragment de codi llegeix l'arxiu `market_basket_transaction.csv` i l'emmagatzema en un objecte de la classe `transaction`:

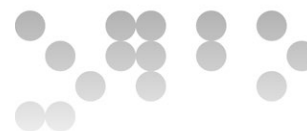


```
if (!require("arules")) {
  # Instal·lació de la llibreria
install.packages("arules")
# Carreguem la llibreria
library(arules)
}
# Llegeix arxiu CSV i emmagatzema contingut en un objecte de la
# classe `transaction`
tr <- read.transactions("data/market_basket_transactions.csv",
                        format = "basket",
                        sep = ",")
```

Visualitzem un resum de l'objecte `tr`:

```
# Visualitzem un resum
summary(tr)

## transactions as itemMatrix in sparse format with
## 22191 rows (elements/itemsets/transactions) and
## 30066 columns (items) and a density of 0.0005390256
##
## most frequent items:
## WHITE HANGING HEART T-LIGHT HOLDER          REGENCY CAKESTAND 3 TIER
##                               1803                      1709
##           JUMBO BAG RED RETROSPOT              PARTY BUNTING
##                               1460                      1285
## ASSORTED COLOUR BIRD ORNAMENT                  (Other)
##                               1250                      352128
##
## element (itemset/transaction) length distribution:
## sizes
##   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##   1 3597 1594 1141  908  861  758  696  676  663  593  624  537  516  531
##  16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
## 551  522  464  441  483  419  395  315  306  272  238  253  229  213  222
##  31   32   33   34   35   36   37   38   39   40   41   42   43   44   45
## 215  170  159  138  142  134  109  111   90  113   94   93   87   88   65
##  46   47   48   49   50   51   52   53   54   55   56   57   58   59   60
##  63   67   63   60   59   49   64   40   41   49   43   36   29   39   30
##  61   62   63   64   65   66   67   68   69   70   71   72   73   74   75
##  27   28   17   25   25   20   27   24   22   15   20   19   13   16   16
##  76   77   78   79   80   81   82   83   84   85   86   87   88   89   90
##  11   15   12   7    9   14   15   12   8    9   11   11   14   8    6
##  91   92   93   94   95   96   97   98   99  100  101  102  103  104  105
##    5    6   11    6    4    4    3    6    5    2    4    2    4    4    3
## 106  107  108  109  110  111  112  113  114  115  117  118  119  121  122
##    2    2    6    3    4    3    2    1    3    1    3    3    3    1    2
## 123  124  126  127  128  132  133  134  135  141  142  143  144  146  147
##    2    1    3    2    2    1    1    2    1    1    2    2    1    1    2
```



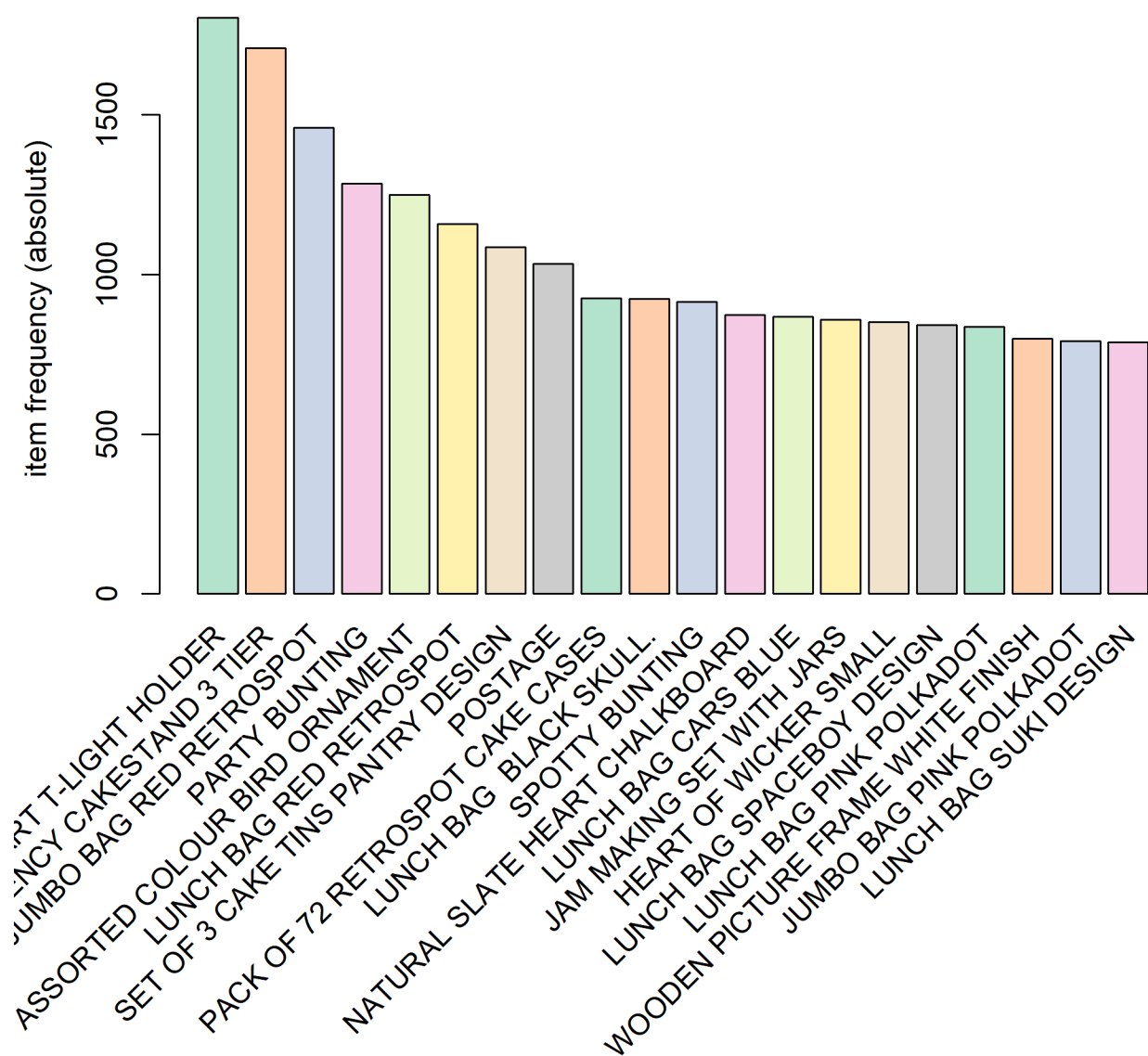
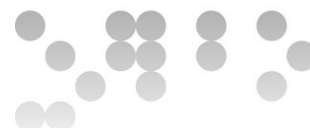
```
## 148 151 155 158 169 172 178 179 181 203 205 229 237 250 251
## 1 1 3 2 2 2 1 1 1 1 1 1 1 1 1
## 286 321 401 420
## 1 1 1 1
##
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 1.00 4.00 11.00 16.21 22.00 420.00
##
## includes extended item information - examples:
## labels
## 1 1
## 2 1 HANGER
## 3 10
```

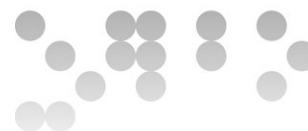
Podem observar en la sortida la següent informació sobre les transaccions:

- S'han generat **22191 transaccions (files)** i **30066 articles (columnes)**.
- Els articles més freqüents. Com per exemple 'WHITE HANGING HEART T-LIGHT HOLDER' amb 1803 articles, REGENCY CAKESTAND 3 TIER amb 1709, etc.

La representació gràfica, seria:

```
# Creació d'un gràfic de freqüència dels articles top 20
if (!require("RColorBrewer")) {
  # Instal·lació de la llibreria
  install.packages("RColorBrewer")
# Carreguem la llibreria
library(RColorBrewer)
}
itemFrequencyPlot(tr, topN=20, type="absolute",
  col=brewer.pal(8, 'Pastel1'))
```





Generació de les regles

En aquest apartat utilitzarem el algoritme apriori per a generar les regles d'associació. Per a trobar un conjunt de regles farem ús de la funció `apriori()` del paquet `arules`.

El prototip de la funció és el següent:

```
apriori(data, parameter = list(list(supp=0.001, conf=0.8, maxlen=10))
```

on els arguments són els següents:

- ****data:**** un objecte de tipus `transaction`.
- **parameter:** una llista especificant les mètriques i el màxim nombre d'elements:
- **supp:** el llindar de suport. Per defecte, **supp=0.001**.
- **conf:** el llindar de confiança. Per defecte, **conf=0.8**.
- **maxlen:** el màxim nombre d'elements. Per defecte, **maxlen=10**.

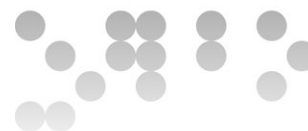
Com a mostra, la següent línia de codi calcula el conjunt de regles amb els valors per defecte:

```
# Executa algoritme a priori amb valors per defecte
rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8, maxlen=10))
```

Tot seguit es mostra un resum del conjunt de regles:

```
# Visualitzem un resum
summary(rules)

## set of 5 rules
##
## rule length distribution (lhs + rhs):sizes
## 4
## 5
##
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```



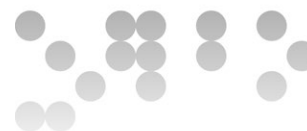
```
##      4      4      4      4      4      4
##
## summary of quality measures:
##      support      confidence      lift      count
## Min.   :0.003066   Min.   :0.8070   Min.   :4.477   Min.   :46.0
## 1st Qu.:0.003133   1st Qu.:0.8167   1st Qu.:4.531   1st Qu.:47.0
## Median :0.003266   Median :0.8254   Median :4.579   Median :49.0
## Mean   :0.003293   Mean   :0.8286   Mean   :4.597   Mean   :49.4
## 3rd Qu.:0.003466   3rd Qu.:0.8393   3rd Qu.:4.656   3rd Qu.:52.0
## Max.   :0.003533   Max.   :0.8548   Max.   :4.742   Max.   :53.0
##
## mining info:
## data ntransactions support confidence
## tr      15001      0.003      0.8
```

Podem observar en la sortida la següent informació sobre els conjunt de regles:

- **Parameter Specification:** on $\text{min_sup}=0.001$ i $\text{min_confidence}=0.8$ amb 10 articles com a màxim en una regla.
- **Total number of rules:** en aquest cas 49122 regles.
- **Distribution of rule lenght:** Una longitud de 5 articles té la majoria de regles i la longitud 2 té el nombre més baix de regles.
- **Summary of Quality measures:** valors màxims i mínims per a les mètriques de suport, confiança i millora.
- ****Mining info:**** les dades, suport, confiança i nombre de transaccions.

Establiment dels llindars de suport i confiança

Seleccionem un nivell de suport del 5% i una confiança del 95%. Amb lo primer aconseguim que cadascuna de les regles estigui present al menys el 5% de les mostres, lo que els hi otorga representativitat, mentres que amb lo segon obtenim la probabilitat de que les regles siguin certes a les mostres en les que els seus antecedents són certs també.



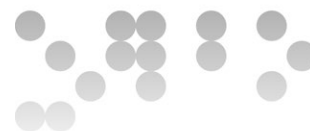
```
# Executem el algoritme a priori amb
# min_supp = 5% i min_conf = 95%
rules <- apriori(tr, parameter = list(supp = 0.005, conf = 0.95))

## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##      0.95      0.1      1 none FALSE              TRUE        5   0.005      1
## maxlen target  ext
##      10  rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE     2    TRUE
##
## Absolute minimum support count: 110
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[30066 item(s), 22191 transaction(s)] done [0.45s].
## sorting and recoding items ... [923 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 done [0.09s].
## writing ... [80 rule(s)] done [0.00s].
## creating S4 object ... done [0.03s].
```

Tot seguit es mostra un resum d'executar l'algoritme apriori:

```
# Visualizem un resum
summary(rules)

## set of 80 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4  5  6
##  8 20 30 18  4
##
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      2.000  3.000  4.000  3.875  5.000  6.000
##
## summary of quality measures:
##      support      confidence      lift      count
## Min.   :0.005137 Min.   :0.9500 Min.   : 69.35 Min.   :114.0
## 1st Qu.:0.005363 1st Qu.:0.9611 1st Qu.:124.23 1st Qu.:119.0
## Median :0.005588 Median :0.9749 Median :128.22 Median :124.0
## Mean   :0.006117 Mean   :0.9734 Mean   :122.00 Mean   :135.8
## 3rd Qu.:0.005926 3rd Qu.:0.9831 3rd Qu.:131.42 3rd Qu.:131.5
## Max.   :0.010410 Max.   :1.0000 Max.   :134.96 Max.   :231.0
##
## mining info:
```

```
## data ntransactions support confidence
## tr          22191    0.005    0.95
```

Com que hi ha 80, mostrem per pantalla les 10 primeres regles:

```
# Visualitzem les 10 primeres regles d'associació
```

```
inspect(rules[1:10])
```

##	lhs	rhs	support	confidence	lift	count
## [1]	{FRONT DOOR}	=> {KEY FOB}	0.005677978	1.00000	71.58387	126
## [2]	{HOT PINK}	=> {FEATHER PEN}	0.005452661	1.00000	113.21939	121
## [3]	{SET 3 RETROSPOT TEA}	=> {SUGAR}	0.010409626	1.00000	96.06494	231
## [4]	{SUGAR}	=> {SET 3 RETROSPOT TEA}	0.010409626	1.00000	96.06494	231
## [5]	{SET 3 RETROSPOT TEA}	=> {COFFEE}	0.010409626	1.00000	69.34687	231
## [6]	{SUGAR}	=> {COFFEE}	0.010409626	1.00000	69.34687	231
## [7]	{BACK DOOR}	=> {KEY FOB}	0.008832410	1.00000	71.58387	196
## [8]	{SHED}	=> {KEY FOB}	0.009598486	1.00000	71.58387	213
## [9]	{HERB MARKER BASIL, HERB MARKER CHIVES}	=> {HERB MARKER THYME}	0.005587851	0.96875	131.08251	124
## [10]	{HERB MARKER BASIL, HERB MARKER CHIVES}	=> {HERB MARKER PARSLEY}	0.005587851	0.96875	131.88669	124

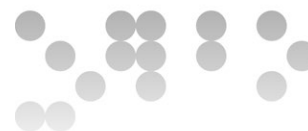
Per exemple, podem observar en la sortida les següents regles:

- 100% dels clients que compren 'FRONT DOOR' també compren 'KEY FOB'
- 96% dels clients que compren 'HERB MARKER BASIL' i 'HERB MARKER CHIVES' també compren 'HERB MARKER THYM'.

Cerca de regles segons antecedent

Per posar un exemple, supossem que necessitem trobar les regles d'associació per a un determinat article. Podem fer ús del paràmetre `appearance` de la funció `apriori()`. En aquest sentit, podem establir un o diversos antecedents i un conseqüent amb LHS (IF part) i RHS (THEN part):

Per exemple, per a respondre a la pregunta *"Els clients que compren METAL també compren ..."* ho farem com es mostra a continuació:



```
# Cerca regles segons antecedent `METAL`
metal_rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8),
                      appearance = list(lhs="METAL", default="rhs"))

## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##      0.8      0.1    1 none FALSE              TRUE      5   0.001      1
## maxlen target  ext
##     10 rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 22
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[30066 item(s), 22191 transaction(s)] done [0.35s].
## sorting and recoding items ... [2324 item(s)] done [0.02s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [1 rule(s)] done [0.00s].
## creating S4 object ... done [0.02s].

inspect(head(metal_rules))

##      lhs      rhs      support  confidence lift  count
## [1] {METAL} => {DECORATION} 0.002253166 1      443.82 50
```

Visualització de les regles

Podem visualitzar les regles d'associació amb `plot()` del paquet `arulesViz`. Utilitza la mètrica de suport en l'eix d'ordenada i la confiança en l'eix d'abscisses. A més, la mètrica de millora (lift) es usada per colorejar els punts.

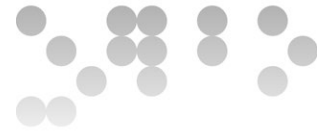
Per exemple, podem visualitzar el conjunt de regles amb un llindar de confiança del 95% com es mostra en el següent fragment de codi:

```
if (!require("arulesViz")) {
  # Instal·lació de la llibreria
  install.packages("arulesViz")
  # Carreguem la llibreria
```

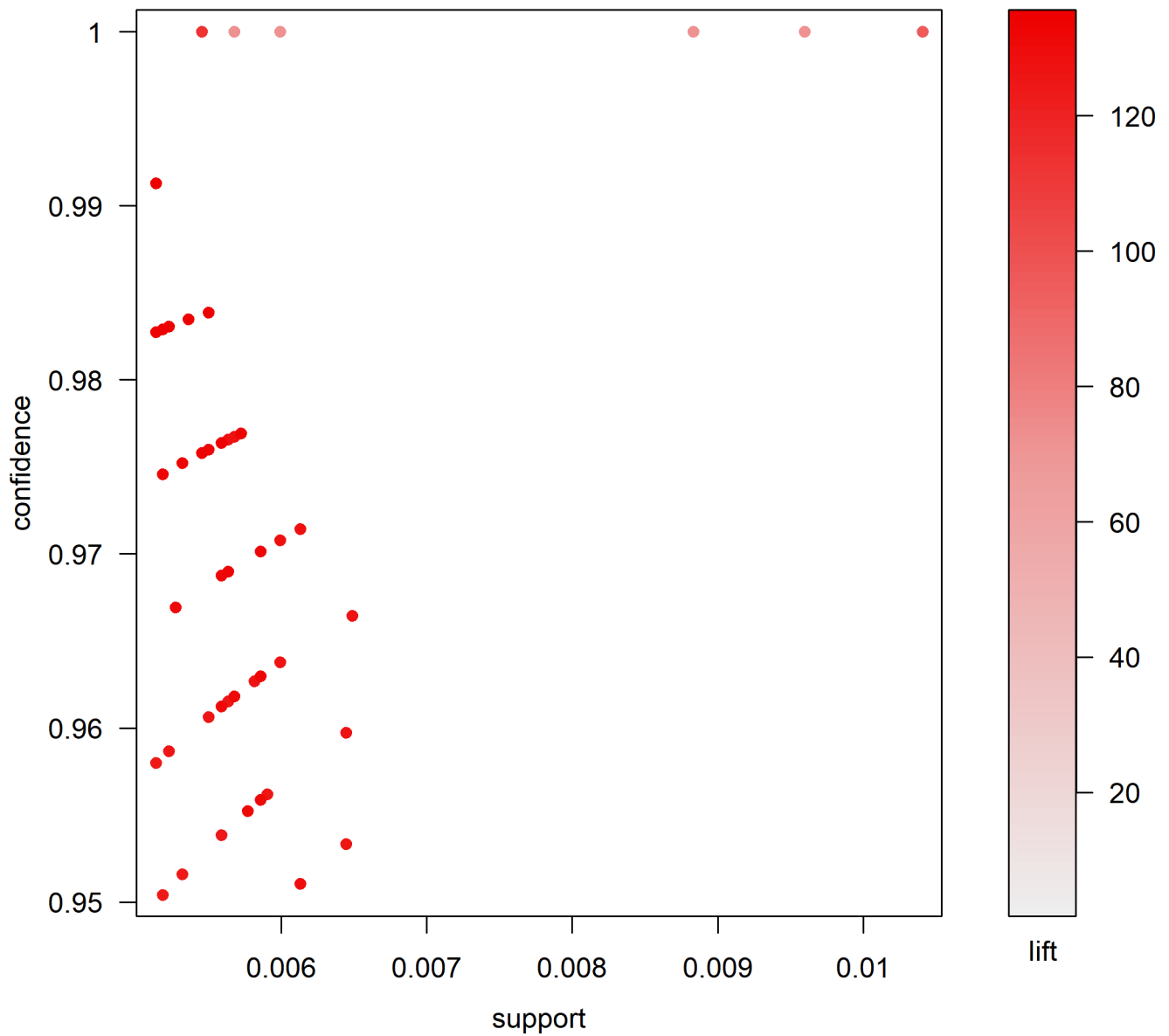


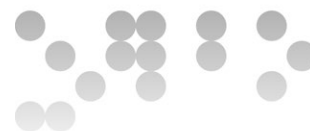
```
library(arulesViz)
}

# Filtra les regles amb min_conf > 0.95
subRules <- rules[quality(rules)$confidence>0.95]
# Diagrama de dispersió amb regles associació amb min_conf>0.95
plot(subRules, jitter=0)
```



Scatter plot for 79 rules





Bibliografia

[1] Daniel T. Larouse, Chantal D. Larouse: Data Mininig and Predictive Analytics.USA, John Wiley & Sons,2015,ISBN 978-1-118-11619-7

[2] Jordi Gironés Roig, Jordi Casas Roma, Julià Minguillón Alfonso, Ramon Caihuelas Quiles : Minería de Datos: Modelos y Algoritmos. Barcelona, Editorial UOC, 2017, ISBN: 978-84-9116-904-8.

[3] Jiawe Han, Michellie Chamber & Jian Pei: Data mining : concepts and techniques. 3º Edition. USA, Editorial Elsevier, 2012, ISBN 978-0-12-381479-1

[4] A Gentle Introduction on Market Basket Analysis - Association Rules. [Fecha de consulta: 29 noviembre 2018]. Disponible en : <https://datascienceplus.com/a-gentle-introduction-on-market-basket-analysis%E2%80%8A-%E2%80%8Aassociation-rules/>

[4] Market Basket Analysis using R. [Fecha de consulta: 30 noviembre 2018]. Disponible en : <https://www.datacamp.com/community/tutorials/market-basket-analysis-r>