# Chapter 15

## ASSOCIATION RULES (2)

## ( 關聯規則 )

# Apriori algorithm using R package (arules)

一開始先以 titanic 資料集
測試套件是否成功安裝及可正常使用

```
1  install.packages("arules")
2  library(arules)
3
4  t_data <- Titanic
5  str(t_data)
6
7  tdf <- as.data.frame(t_data)
8  head(tdf)
9  View(tdf)
```

Console Terminal

```
> str(t_data)
 table [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...
 - attr(*, "dimnames")=List of 4
  ..$ Class   : chr [1:4] "1st" "2nd" "3rd" "Crew"
  ..$ Sex     : chr [1:2] "Male" "Female"
  ..$ Age     : chr [1:2] "Child" "Adult"
  ..$ Survived: chr [1:2] "No" "Yes"
> tdf <- as.data.frame(t_data)
> head(tdf)
  Class    Sex   Age Survived Freq
1   1st   Male Child       No    0
2   2nd   Male Child       No    0
3   3rd   Male Child       No   35
4  Crew   Male Child       No    0
5   1st Female Child       No    0
6   2nd Female Child       No    0
```

# Apriori algorithm using R package (arules)

若將 table 直接轉成 data.frame 只會剩下 32 筆資料
原本是描述重複次數的 Freq 被獨立變成一個欄位

| | Class | Sex | Age | Survived | Freq |
|---|---|---|---|---|---|
| 1 | 1st | Male | Child | No | 0 |
| 2 | 2nd | Male | Child | No | 0 |
| 3 | 3rd | Male | Child | No | 35 |
| 4 | Crew | Male | Child | No | 0 |
| 5 | 1st | Female | Child | No | 0 |
| 6 | 2nd | Female | Child | No | 0 |
| 7 | 3rd | Female | Child | No | 17 |
| 8 | Crew | Female | Child | No | 0 |
| 9 | 1st | Male | Adult | No | 118 |
| 10 | 2nd | Male | Adult | No | 154 |
| 11 | 3rd | Male | Adult | No | 387 |
| 12 | Crew | Male | Adult | No | 670 |
| 13 | 1st | Female | Adult | No | 4 |
| 14 | 2nd | Female | Adult | No | 13 |
| 15 | 3rd | Female | Adult | No | 89 |
| 16 | Crew | Female | Adult | No | 3 |
| 17 | 1st | Male | Child | Yes | 5 |
| 18 | 2nd | Male | Child | Yes | 11 |
| 19 | 3rd | Male | Child | Yes | 13 |
| 20 | Crew | Male | Child | Yes | 0 |
| 21 | 1st | Female | Child | Yes | 1 |
| 22 | 2nd | Female | Child | Yes | 13 |
| 23 | 3rd | Female | Child | Yes | 14 |
| 24 | Crew | Female | Child | Yes | 0 |
| 25 | 1st | Male | Adult | Yes | 57 |
| 26 | 2nd | Male | Adult | Yes | 14 |
| 27 | 3rd | Male | Adult | Yes | 75 |
| 28 | Crew | Male | Adult | Yes | 192 |
| 29 | 1st | Female | Adult | Yes | 140 |
| 30 | 2nd | Female | Adult | Yes | 80 |
| 31 | 3rd | Female | Adult | Yes | 76 |
| 32 | Crew | Female | Adult | Yes | 20 |

# Apriori algorithm using R package (arules)

```r
library(arules)

t_data <- Titanic
str(t_data)

tdf <- as.data.frame(t_data)
head(tdf)
#View(tdf)

tdf$Freq <- NULL

ar <- apriori(tdf, parameter =list(minlen=3, supp =0.1, conf=0.5),
                   appearance =list(rhs=c("Survived=No",
                                          "Survived=Yes"),default="lhs"),
        control=list(verbose=T))

inspect(ar)
```

步驟到這邊
已經確定可以執行
Apriori 演算法了

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.5    0.1    1 none FALSE            TRUE       5     0.1      3     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 3

set item appearances ...[2 item(s)] done [0.00s].
set transactions ...[10 item(s), 32 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [8 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

```
> inspect(ar)
    lhs                        rhs              support confidence lift count
[1] {Sex=Male,Age=Child}    => {Survived=Yes} 0.125   0.5        1    4
[2] {Sex=Male,Age=Adult}    => {Survived=Yes} 0.125   0.5        1    4
[3] {Sex=Female,Age=Child}  => {Survived=Yes} 0.125   0.5        1    4
[4] {Sex=Female,Age=Adult}  => {Survived=Yes} 0.125   0.5        1    4
[5] {Sex=Male,Age=Child}    => {Survived=No}  0.125   0.5        1    4
[6] {Sex=Male,Age=Adult}    => {Survived=No}  0.125   0.5        1    4
[7] {Sex=Female,Age=Child}  => {Survived=No}  0.125   0.5        1    4
[8] {Sex=Female,Age=Adult}  => {Survived=No}  0.125   0.5        1    4
```

# Apriori algorithm using R package (arules)

## Apriori 演算法參數說明

```
12 ar <- apriori(tdf, parameter =list(minlen=3, supp =0.1, conf=0.5),
13                    appearance =list(rhs=c("Survived=No",
14                                           "Survived=Yes"),default="lhs"),
15          control=list(verbose=T))
```

| 參數 | 子參數 | 說明 |
|---|---|---|
| paprmeter | minlen / maxlen : | 項目集長度最小 / 大值 |
| | supp / conf : | 最小支持度和最大信心度 |
| | target = “rules” / “frequent itemsets” | 產生 Rule 或 FP |
| apperance | 可以對先決條件 lhs 和關聯結果 rhs<br>實際包含那些項目進行限製 | ex. rhs = c(“Age”)<br>ex. none = c(“Sex”) |
| control | sort | 結果排序 1: 昇冪 -1: 降冪 |
| | verbose | 秀出處理程序 T/F |

# Titanic data preprocessing

確定套件可運作後
現在把 Titanic dataset 依據 Freq 的次數進行資料前處理

```
1  library(arules)
2  t_data <- Titanic
3  tdf    <- as.data.frame(t_data)
4  dp_df <- NULL
5  hname <- c("Class","Sex","Age","Survived")
6
7  #前4個維度才是我們要的,因為第5個是Freq
8  for(i in 1:4)
9  {
10    #依據Freq 數值複製幾次
11    x<- rep(as.character(tdf[,i]),tdf$Freq)
12    dp_df <- cbind(dp_df,x)
13    colnames(dp_df)[i]<-hname[i]
14 }
15 dp_df <- as.data.frame(dp_df)
16 str(dp_df)
17
18 ar <- apriori(dp_df, parameter =list(minlen=3, supp =0.15, conf=0.75),
19                control=list(verbose=F))
20 inspect(ar)
```

```
> dp_df <- as.data.frame(dp_df)
> str(dp_df)
'data.frame':   2201 obs. of  4 variables:
 $ Class   : Factor w/ 4 levels "1st","2nd","3rd",..: 3 3
 $ Sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2
 $ Age     : Factor w/ 2 levels "Adult","Child": 2 2 2 2
 $ Survived: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1
```

# Titanic data preprocessing

前處理後可依據不同的 sup, conf, minlen, maxlen, target 找出不同的結果
Ex. Target = FP

```
19 ar <- apriori(dp_df, parameter =list(minlen=3,
20                                      supp =0.3, conf=0.75,
21                                      target = "frequent itemsets"),
22               control=list(verbose=F))
```

```
> inspect(ar)
    items                                              support   count
[1] {Class=Crew,Sex=Male,Survived=No}                  0.3044071  670
[2] {Class=Crew,Age=Adult,Survived=No}                 0.3057701  673
[3] {Class=Crew,Sex=Male,Age=Adult}                    0.3916402  862
[4] {Sex=Male,Age=Adult,Survived=No}                   0.6038164 1329
[5] {Class=Crew,Sex=Male,Age=Adult,Survived=No}        0.3044071  670
```

# Titanic data preprocessing

前處理後可依據不同的 sup, conf, minlen, maxlen, target 找出不同的結果

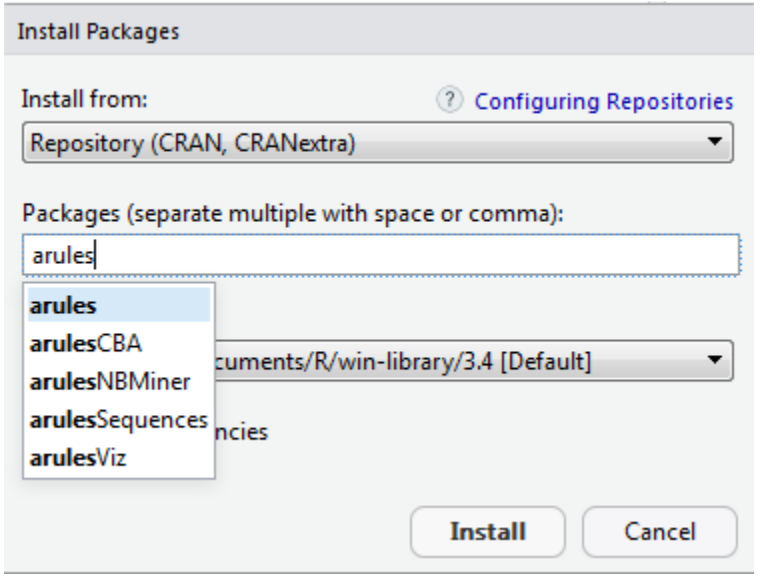
Ex. Target = Rule

```
18  ar <- apriori(dp_df, parameter =list(minlen=3, supp =0.15, conf=0.75),
19                control=list(verbose=F))
20  inspect(ar)
```

```
> inspect(ar)
     lhs                                    rhs           support   confidence lift      count
[1]  {Class=3rd,Survived=No}             => {Sex=Male}    0.1917310 0.7992424  1.0162522  422
[2]  {Class=3rd,Sex=Male}                => {Survived=No} 0.1917310 0.8274510  1.2222950  422
[3]  {Class=3rd,Survived=No}             => {Age=Adult}   0.2162653 0.9015152  0.9484870  476
[4]  {Class=3rd,Age=Adult}               => {Survived=No} 0.2162653 0.7591707  1.1214326  476
[5]  {Class=3rd,Sex=Male}                => {Age=Adult}   0.2099046 0.9058824  0.9530818  462
[6]  {Sex=Male,Survived=Yes}             => {Age=Adult}   0.1535666 0.9209809  0.9689670  338
[7]  {Class=Crew,Survived=No}            => {Sex=Male}    0.3044071 0.9955423  1.2658514  670
[8]  {Class=Crew,Sex=Male}               => {Survived=No} 0.3044071 0.7772622  1.1481571  670
[9]  {Class=Crew,Survived=No}            => {Age=Adult}   0.3057701 1.0000000  1.0521033  673
[10] {Class=Crew,Age=Adult}              => {Survived=No} 0.3057701 0.7604520  1.1233254  673
[11] {Class=Crew,Sex=Male}               => {Age=Adult}   0.3916402 1.0000000  1.0521033  862
[12] {Class=Crew,Age=Adult}              => {Sex=Male}    0.3916402 0.9740113  1.2384742  862
[13] {Sex=Male,Survived=No}              => {Age=Adult}   0.6038164 0.9743402  1.0251065 1329
[14] {Age=Adult,Survived=No}             => {Sex=Male}    0.6038164 0.9242003  1.1751385 1329
[15] {Sex=Male,Age=Adult}                => {Survived=No} 0.6038164 0.7972406  1.1776688 1329
[16] {Class=3rd,Sex=Male,Survived=No}    => {Age=Adult}   0.1758292 0.9170616  0.9648435  387
[17] {Class=3rd,Age=Adult,Survived=No}   => {Sex=Male}    0.1758292 0.8130252  1.0337773  387
[18] {Class=3rd,Sex=Male,Age=Adult}      => {Survived=No} 0.1758292 0.8376623  1.2373791  387
[19] {Class=Crew,Sex=Male,Survived=No}   => {Age=Adult}   0.3044071 1.0000000  1.0521033  670
[20] {Class=Crew,Age=Adult,Survived=No}  => {Sex=Male}    0.3044071 0.9955423  1.2658514  670
[21] {Class=Crew,Sex=Male,Age=Adult}     => {Survived=No} 0.3044071 0.7772622  1.1481571  670
```

# 將關聯規則視覺化

## 安裝並使用 arulesViz 套件



# 將關聯規則視覺化

## 產生互動式散佈圖

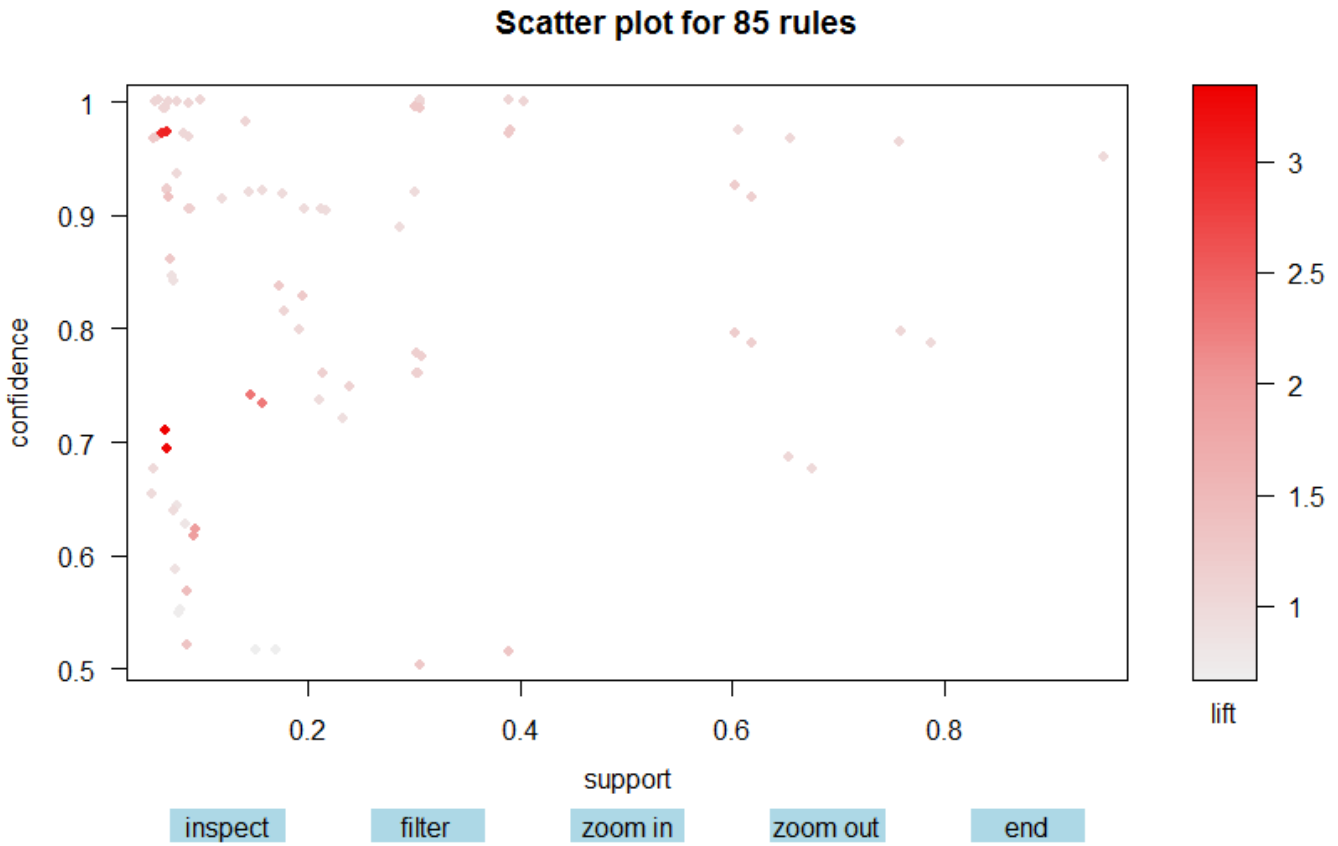
```
rm(list=ls())
library(arules)
library(arulesViz)
t_data <- Titanic
tdf    <- as.data.frame(t_data)
dp_df <- NULL
hname <- c("Class","Sex","Age","Survived")

for(i in 1:4)
{
  x<- rep(as.character(tdf[,i]),tdf$Freq)
  dp_df <- cbind(dp_df,x)
  colnames(dp_df)[i]<-hname[i]
}
dp_df <- as.data.frame(dp_df)
str(dp_df)

ar <- apriori(dp_df, parameter =list(supp =0.05, conf=0.5),
              control=list(verbose=F))
#inspect(ar)
plot(ar, interactive=T)
```

# 將關聯規則視覺化

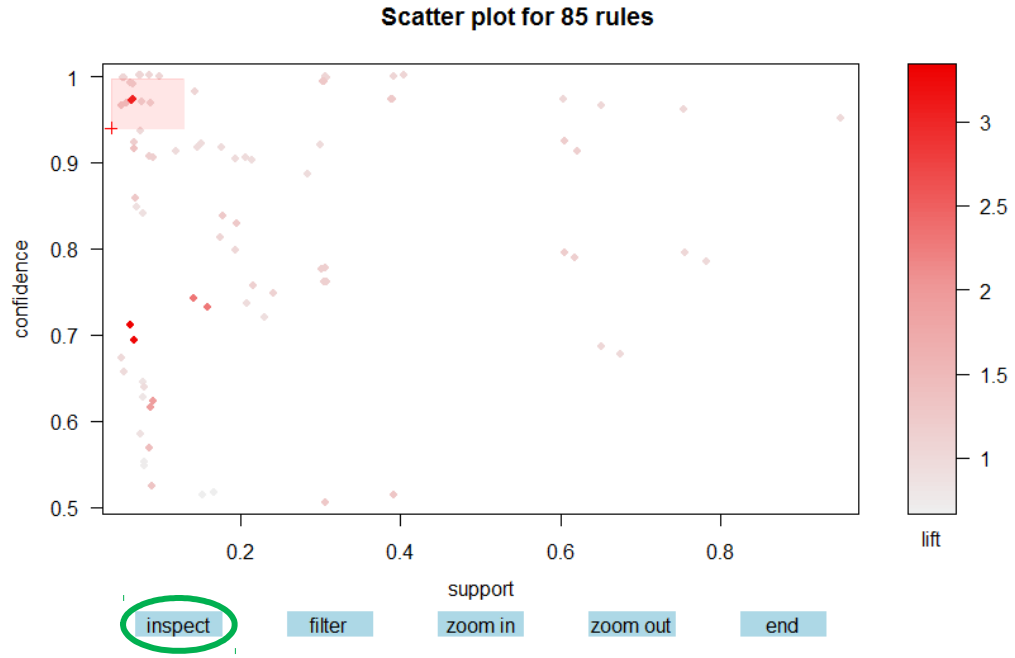
## 產生互動式散佈圖



# 將關聯規則視覺化

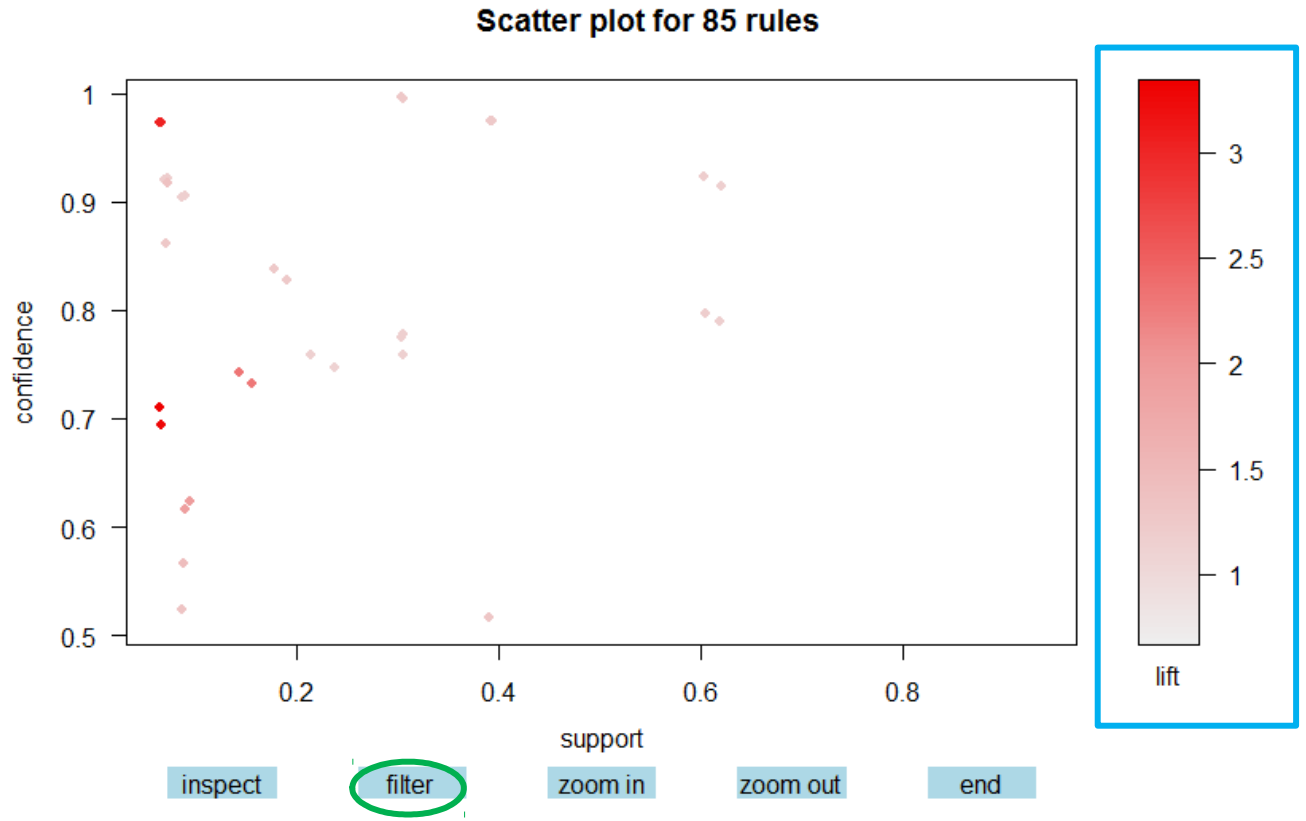
## 產生互動式散佈圖

```
Number of rules selected: 8
    lhs                                  rhs             support    confidence lift     count order
[1] {Class=1st,Sex=Female}               => {Survived=Yes} 0.06406179 0.9724138  3.010243 141   3
[2] {Class=1st,Sex=Female,Age=Adult}     => {Survived=Yes} 0.06360745 0.9722222  3.009650 140   4
[3] {Class=1st,Survived=No}              => {Sex=Male}     0.05361199 0.9672131  1.229830 118   3
[4] {Class=1st,Age=Adult,Survived=No}    => {Sex=Male}     0.05361199 0.9672131  1.229830 118   4
[5] {Class=1st,Sex=Female}               => {Age=Adult}    0.06542481 0.9931034  1.044847 144   3
[6] {Class=1st,Sex=Female,Survived=Yes}  => {Age=Adult}    0.06360745 0.9929078  1.044642 140   4
[7] {Class=1st,Sex=Male}                 => {Age=Adult}    0.07950931 0.9722222  1.022878 175   3
[8] {Class=1st,Survived=Yes}             => {Age=Adult}    0.08950477 0.9704433  1.021007 197   3
```



# 將關聯規則視覺化

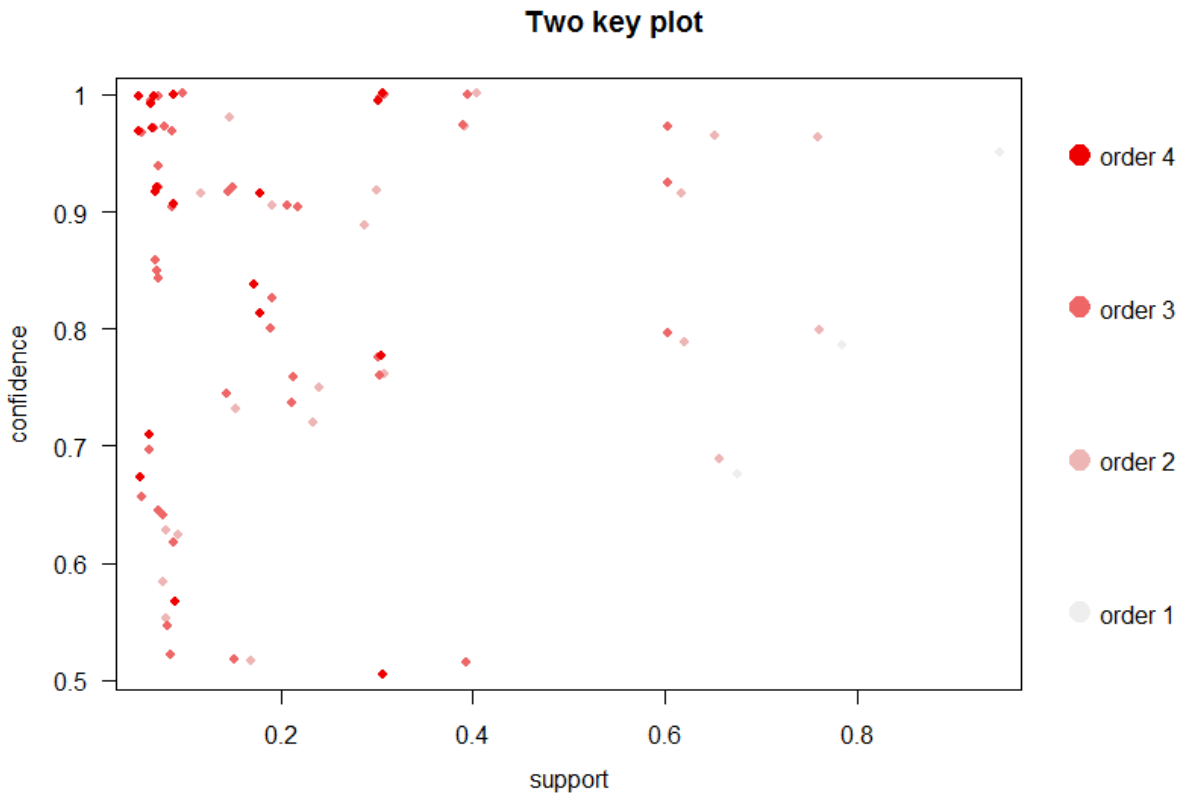
## 產生互動式散佈圖



# 將關聯規則視覺化

## Two-key point plot
## 顏色愈深代表規則的數量愈多

```
21  plot(ar, shading="order",control=list(main ="Two key plot"))
```



# ECLAT algorithm using R package (arules)

eclat(…)
#eclat 只能跑出 FP, 不能跑出 Rule

```
1  rm(list=ls())
2  library(arules)
3  library(arulesViz)
4  data("Groceries")
5
6  g_data<-Groceries
7
8  er <- eclat(g_data, parameter=list(supp=0.15),
9              control=NULL)
10
11 inspect(er)
```

```
Eclat

parameter specification:
 tidLists support minlen maxlen            target   ext
    FALSE    0.15      1     10 frequent itemsets FALSE

algorithmic control:
 sparse sort verbose
      7   -2    TRUE

Absolute minimum support count: 1475

create itemset ...
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating bit matrix ... [4 row(s), 9835 column(s)] done [0.00s].
writing  ... [4 set(s)] done [0.00s].
Creating S4 object  ... done [0.00s].
>
> inspect(er)
     items              support   count
[1] {whole milk}       0.2555160 2513
[2] {other vegetables} 0.1934926 1903
[3] {rolls/buns}       0.1839349 1809
[4] {soda}             0.1743772 1715
```

# 隨堂練習 1

1. 讀取內建資料集 Groceries

```
> g_data<-Groceries
>
> summary(g_data)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns             soda           yogurt          (Other)
            2513             1903             1809             1715             1372            34055
```

# 隨堂練習 1

2. 觀測前 5 筆交易資料

```
> inspect (g_data[1:5])
    items
[1] {citrus fruit,semi-finished bread,margarine,ready soups}
[2] {tropical fruit,yogurt,coffee}
[3] {whole milk}
[4] {pip fruit,yogurt,cream cheese ,meat spreads}
[5] {other vegetables,whole milk,condensed milk,long life bakery product}
>
```

# 隨堂練習 1

3. 使用 Apriori 找出 Rule (supp = 0.01, conf =0.5)

| | lhs | | rhs | support | confidence | lift | count |
|---|---|---|---|---|---|---|---|
| [1] | {curd,yogurt} | => | {whole milk} | 0.01006609 | 0.5823529 | 2.279125 | 99 |
| [2] | {other vegetables,butter} | => | {whole milk} | 0.01148958 | 0.5736041 | 2.244885 | 113 |
| [3] | {other vegetables,domestic eggs} | => | {whole milk} | 0.01230300 | 0.5525114 | 2.162336 | 121 |
| [4] | {yogurt,whipped/sour cream} | => | {whole milk} | 0.01087951 | 0.5245098 | 2.052747 | 107 |
| [5] | {other vegetables,whipped/sour cream} | => | {whole milk} | 0.01464159 | 0.5070423 | 1.984385 | 144 |
| [6] | {pip fruit,other vegetables} | => | {whole milk} | 0.01352313 | 0.5175097 | 2.025351 | 133 |
| [7] | {citrus fruit,root vegetables} | => | {other vegetables} | 0.01037112 | 0.5862069 | 3.029608 | 102 |
| [8] | {tropical fruit,root vegetables} | => | {other vegetables} | 0.01230300 | 0.5845411 | 3.020999 | 121 |
| [9] | {tropical fruit,root vegetables} | => | {whole milk} | 0.01199797 | 0.5700483 | 2.230969 | 118 |
| [10] | {tropical fruit,yogurt} | => | {whole milk} | 0.01514997 | 0.5173611 | 2.024770 | 149 |
| [11] | {root vegetables,yogurt} | => | {other vegetables} | 0.01291307 | 0.5000000 | 2.584078 | 127 |
| [12] | {root vegetables,yogurt} | => | {whole milk} | 0.01453991 | 0.5629921 | 2.203354 | 143 |
| [13] | {root vegetables,rolls/buns} | => | {other vegetables} | 0.01220132 | 0.5020921 | 2.594890 | 120 |
| [14] | {root vegetables,rolls/buns} | => | {whole milk} | 0.01270971 | 0.5230126 | 2.046888 | 125 |
| [15] | {other vegetables,yogurt} | => | {whole milk} | 0.02226741 | 0.5128806 | 2.007235 | 219 |

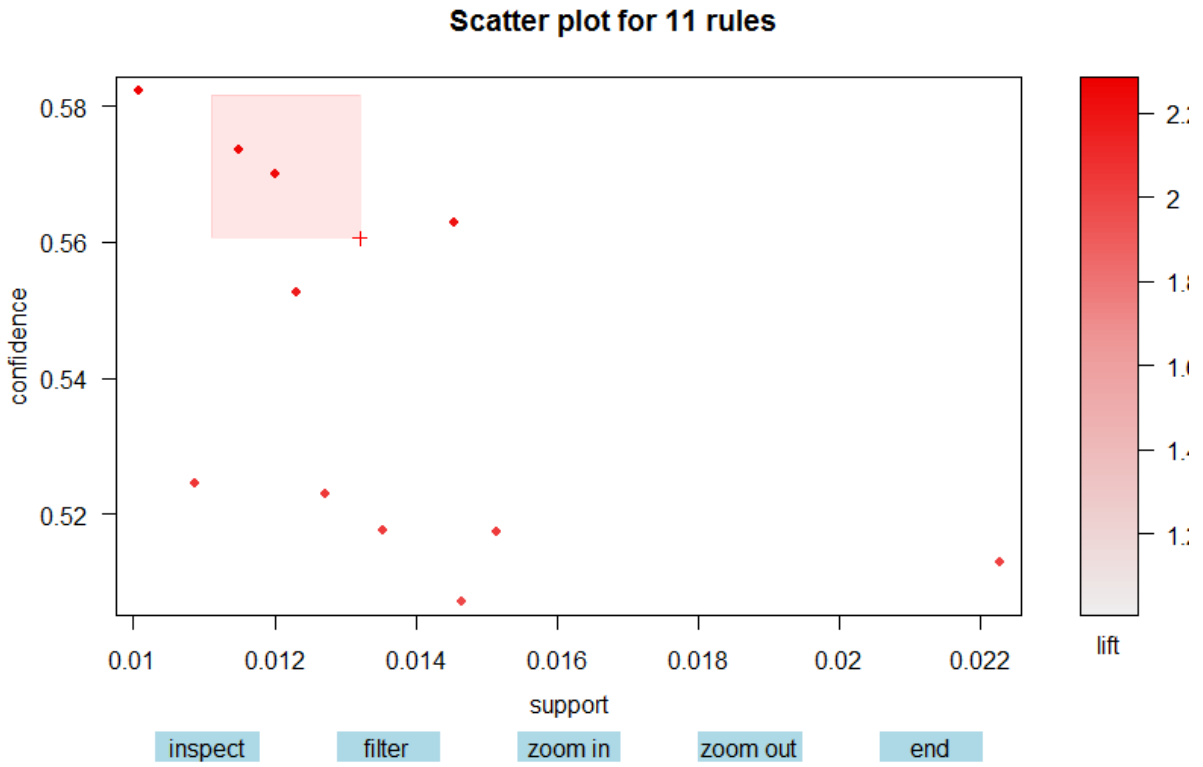
# 隨堂練習 1

4. 限制 Rule 的 rhs 只能是 whole milk

```
     lhs                                    rhs           support    confidence lift     count
[1]  {curd,yogurt}                       => {whole milk} 0.01006609 0.5823529  2.279125  99
[2]  {other vegetables,butter}           => {whole milk} 0.01148958 0.5736041  2.244885 113
[3]  {other vegetables,domestic eggs}    => {whole milk} 0.01230300 0.5525114  2.162336 121
[4]  {yogurt,whipped/sour cream}         => {whole milk} 0.01087951 0.5245098  2.052747 107
[5]  {other vegetables,whipped/sour cream} => {whole milk} 0.01464159 0.5070423  1.984385 144
[6]  {pip fruit,other vegetables}        => {whole milk} 0.01352313 0.5175097  2.025351 133
[7]  {tropical fruit,root vegetables}    => {whole milk} 0.01199797 0.5700483  2.230969 118
[8]  {tropical fruit,yogurt}             => {whole milk} 0.01514997 0.5173611  2.024770 149
[9]  {root vegetables,yogurt}            => {whole milk} 0.01453991 0.5629921  2.203354 143
[10] {root vegetables,rolls/buns}        => {whole milk} 0.01270971 0.5230126  2.046888 125
[11] {other vegetables,yogurt}           => {whole milk} 0.02226741 0.5128806  2.007235 219
```

# 隨堂練習 1

5. 產生互動散佈圖



# Any Questions !?