抽樣設計與調查資料探勘
R軟體實作工作坊

淡江統計系 陳景祥 (Steve Chen)

◦「R軟體:應用統計方法」作者
◦NetStat 作者(http://netstat.stat.tku.edu.tw)
◦R-Web 團隊 (http://www.r-web.com.tw)
◦steve@stat.tku.edu.tw

課程大綱

- 第一天: R軟體一日上手
- 第二天: 抽樣設計理論與實作
- 第三天: 調查資料探勘與實作

補充教材網址：

http://steve-chen.net/SurveyR/

Day 1: R 軟體一日上手

- R 軟體與 R 程式簡介
- R 程式的特殊寫作要求
- 變數型態
- 條件執行與迴圈
- R 的 Function
- 資料處理
- 機率與統計應用

範例資料檔 iris

150 朵鳶尾花的測量資料, 每一朵花包含 5 個變數

- Sepal.Length : 花萼長度
- Sepal.Width : 花萼寬度
- Petal.Length : 花瓣長度
- Petal.Length : 花瓣寬度
- Species : 花的品種,
  setosa, versicolor, virginica

範例資料檔 babies

1236 個懷孕母親的測量資料, 7 個變數

- bwt : 出生嬰兒的體重（單位: once）
- gestation : 懷孕日數
- parity:出生胎序. 0=第一個小孩,1=其他
- age,height,weight:母親年齡、身高、體重
- smoke:母親懷孕時是否抽煙. 1=yes, 0=no

Simple R Operation

```
> 1 + 1
[1] 2
> scores = scan()
1: 99
2: 63
.....
> mean(scores)
[1] 71
> sd(scores)
[1] 20.73644
```

A Very Simple R Program

```
# 讀入文字資料檔(text file), 數值以空格隔開
# xdata 為 R 的資料框架變數 (data-frame)
xdata = read.table("c:/dir2/babies.txt",header=TRUE)
# 刪除所有包含遺失值的個體
xdata = na.exclude(xdata)
mean(xdata$age) ; median(xdata$age)
hist(xdata$bwt)
result = lm(bwt ~ age + weight + smoke, data=xdata)
summary(result)
```

程式說明

1. "c:/dir2/babies.txt" : R 軟體的檔案路徑寫法與 Windows 不同，但也可以寫成 "c:\\dir2\\babies"
2. header = TRUE : 資料檔第一個橫列是變數名稱(空格隔開)
3. xdata 是資料框架變數(data-frame), 可儲存整個資料檔
4. na.exclude(...) : 刪除包含遺失值(NA)的個體/橫列
5. mean( ) : 計算平均數 , median( ) : 計算中位數
6. hist( ) : 畫出直方圖 (histogram)
7. lm( ) : linear model 函數, 此例中是計算迴歸分析
8. bwt ~ age + weight + smoke : bwt 是被解釋變數, age、weight、smoke 是解釋變數
9. summary(result) : 顯示迴歸分析計算的彙整資訊

學習 R 軟體的可能障礙

1. Packages/functions 的英文說明比較難懂
2. 如何找出適合的 packages/functions？
3. 無法將自己的資料轉成需要的形式
4. 看不懂別人寫的 R 程式
5. 可用多種不同程式寫法解決同一個問題？
6. 不知道該用什麼程式技巧來完成自己需要的功能

R 軟體 Help/疑問求解

- ?關鍵字
- ??關鍵字　　# 模糊搜尋
- apropos("關鍵字")　# 模糊搜尋
- library(help=package名稱)
- ?package名稱::函數名稱
- demo(package名稱)
- example(函數名稱)　# 需先 library(package)
- CRAN 網站的 Task Views
- CRAN 網站的 search

程式語言的五大特色

(1)常數、一般變數、向量或陣列變數
(2) Input 與 Output 功能
(3) 條件執行 (例如 if , else)
(4) 迴圈功能 (例如 for, while)
(5) 獨立模組功能：例如 function, procedure, subroutine, module 等

程式 = 自由 (Freedom)！

R 程式 Example

```
babies =
    read.csv("d:/data/babies.csv")
babies = na.exclude(babies)
f1 = function(x)
{
  if (x >= 30) {
     y = "old"
  } else {
     y = "young"
  }
  return(y)
}
```

```
n = nrow(babies)
age2 = character(n)
for (i in 1:n)
{
  age2[i] = f1(babies$age[i])
}
babies$age2 = age2
write(babies,"d:/data/babies2.csv",
          row.names=FALSE)
# f1, 迴圈, 與 if-else 可濃縮成
# age2 = ifelse(babies$age >= 30,
                ,"old","young")
```

學習程式語言的階段

Step 1. 模仿/複製

找到一模一樣的程式，直接使用

Step 2. 修改

找到相似的程式，修改使用

Step 3. 獨創

自己寫出全新、獨特的程式

R 的學習目標

1. 讀入外部資料，直接使用別人寫好的 package / function 作分析
2. 讀入外部資料，作資料處理/轉換/彙整
3. 讀入外部資料，並作資料處理/轉換，再使用現有的 packages/functions 計算
4. 修改他人所寫的 functions/packages，以適用於自己的資料分析任務
5. 依照自己的特殊需求寫出全新的程式

R Programming Levels

(0)X = read.table("d:/dir2/mydata.txt",header=T)
(1)寫程式引用適當函數來分析資料
(2)細緻化處理或美化 Output 與圖形
(3)動態變數替換(Variable-Replacement)
(4)在 R 程式中使用其他程式語言library
(5)Package 包裝
(6)簡單 package 寫作(R programs)
(7) Class 與 Methods
(8)進階 package 寫作(C, Fortran, Java)
(9)高階 package 寫作(GUI, HTML, LaTeX 處理)
(10)大型資料處理、多機平行運算

R 軟體的特色

- Vector 與 Array 運算導向
- 與統計領域直接對應的變數型態
- 以函數(function)與套件(package)為模組
- 強大的繪圖功能
- 活躍的套件(package)發展與更新
- R程式可以使用 C, Fortran, Java 等程式
- 完整的程式語言功能
- 可執行平行運算(Parallel Computing)
- 樂高玩具特質：打照出自己的 R 環境

R vs. SAS and SPSS

功能	R	SAS	SPSS
程式語言功能	完整	不完整	不完整
繪圖功能	超強	普通	普通
應用最新研究	快	慢	慢
分析模組數目	5800多個 packages	普通	普通
GUI 客製化	容易	麻煩	麻煩
Output 客製化	容易	很難	很難
可使用其他語言或軟體的功能	強	稀少	稀少
價格	0	昂貴:每年付費	昂貴

商業統計軟體選擇與 R 共舞

以下商業統計軟體均已提供該軟體與 R 的溝通介面，可以使用 R 的 packages

- SAS
- SPSS
- Minitab
- Stata

R程式寫作規則

- X 與 x 不一樣：名稱的英文大小寫有差異
- 變數名稱不能包含空格、特殊符號、及 "–"
- 一個橫列(row)放好幾個運算式：使用 ";"

```
x = 3 ; y = 2 ; z = 10
```

- 一個較長的運算式寫成多列：運用","等符號切斷

```
result = lm( Y ~ X1+X2 , data = mydata)
```

- x = y = z = 3 是合法的 R 程式寫法
- (x = 3) 相當於 x = 3; x

R 的變數:一般變數

```
x = 12.5              # 一般數值
A1= "John"            #文字字串
z = FALSE             # 邏輯值 TRUE/FALSE 或 T/F
y = 2.4e3             # 2.4 x 10^3 = 2400
A2 = paste(A1,"Dow",sep="")    # A2="John Dow"
x^2                   # 平方, 也可寫成 x**2
x + 3  ; x - 3 ; x * 3 ; x / 3        # 加減乘除
x %/% 3               # 整除
x %% 3                # 餘數
```

R 的變數:向量(Vector)

```
x = c(1, 15.2, 33)
y = c("男","女","女")    # 文字向量
z = c(TRUE,FALSE, FALSE, TRUE)        # 或 c(T,F,F,T)
x2 = x + 1   #   x2 = c(2, 16.2, 14)
x[2]         #   15.2
x[c(1,3)]    # c(1, 33)
x[-2]        # c(1, 33)
y[3]         # "女"
y[1:2]       # c("男","女")
```

R 的變數：Factor 變數

因子變數(Factor)用來儲存資料中的分類變數

```
> gender = c("Boy","Girl","Girl","Boy","Girl")  # 文字向量
> gender = as.factor(gender)  ;   gender     # Factor 變數
  [1] Boy  Girl Girl Boy  Girl
  Levels: Boy Girl
> parttime = c(1,0,0,0,1)              # 數值向量
> parttime = as.factor(parttime)  ;  parttime  # Factor 變數
  [1] 1 0 0 0 1
  Levels: 0  1
> parttime[2]     # Factor 變數的指標使用跟向量變數一樣
>  result = lm(score ~ gender + parttime + IQ )    # 應用
```

R 的變數：矩陣(Matrix)[1]

```
> x1 = c(11,12,13)
> x2 = c(21,22,23)
> M1 = rbind(x1,x2)  #row bind
      [,1] [,2] [,3]
x1      11   12   13
x2      21   22   23
> M2 = cbind(x1,x2)
  #column bind
      x1 x2
[1,] 11 21
[2,] 12 22
[3,] 13 23
```

```
> M3 = matrix(11:16,2,3)
     [,1] [,2] [,3]
[1,]   11   13   15
[2,]   12   14   16
> M3 %*% M2        #矩陣相乘
     x1   x2
[1,] 472 862
[2,] 508 928
> solve(M3 %*% M2) #反矩陣
          [,1]      [,2]
x1   7.733333 -7.183333
x2 -4.233333  3.933333
```

R 的變數：矩陣(Matrix) [2]

```
> M3 = matrix(11:16,2,3)
     [,1] [,2] [,3]
[1,]   11   13   15
[2,]   12   14   16
> M3[2,3]
[1] 16
> M3[2, ]
[1] 12 14 16
> M3[ ,3]
[1] 15 16
```

```
> M3[ ,1:2]
     [,1] [,2]
[1,]   11   13
[2,]   12   14
> M3[ ,c(1,3)]
     [,1] [,2]
[1,]   11   15
[2,]   12   16
```

R 的變數：List（串列）

串列(List)：向量的擴充，可包含不同屬性的元素

```
> friend1 = list(fname="John",age=32,child.ages=c(2,5))
> friend1$fname
    [1] "John"
> friend1$age
    [1] 32
> friend1$child.ages
    [1] 2    5
> friend1$child.ages[2]
    [1] 5
```

List應用：R函數的Output

R 函數計算的結果，多數可儲存為 LIST(串列)變數，可自由挑選所要的部分資訊

```
> x = rnorm(20); y = rnorm(20)
> lm.result = lm(y ~ x)      # 簡單迴歸分析
>lm.result
Call:
lm(formula = y ~ x)
Coefficients:
(Intercept)            x
     0.2781      -0.2354
> names(lm.result)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
> lm.result$coefficients
(Intercept)            x
  0.2781229  -0.2353573
```

R 的變數：資料框架 [1]

資料框架(Data-Frame): 儲存整個資料檔

Example: iris 資料檔就是 R 內建的資料框架變數

```
> head(iris , 2)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
> mean(iris$Sepal.Length)
[1] 5.843333
> table(iris$Species)
 setosa versicolor  virginica
     50         50         50
```

R 的變數：資料框架 [2]

建立資料框架變數

(1)Xdata = read.table(...) 或 Xdata = read.csv(...)

(2)使用 data.frame( ) 函數

```
score = c(60,54,36,72)
gender = c("男","女","女","男")
age = c(21,20,19,18)
students = data.frame(gender,age,score)
```

(3) 直接輸入

```
Xdata = data.frame() ;  Xdata = edit(Xdata)
```

R 的變數：table 變數

```
> gender = c("M","M","M","F","M","M","F","M","M","F")
> blood = c("B","O","A","B","AB","O","B","O","B","B")
> mytab = table(gender,blood)
> mytab                    # mytab 其實是一個 matrix 變數
        blood
gender  A  AB  B  O
     F  0   0  3  0
     M  1   1  2  3
> mytab[1,3]
[1] 3
> mytab[1,]
 A  AB  B  O
 0   0  3  0
```

奇怪的符號： =，<-，<<-

- x <- 3 相當於 x = 3
- <<- 在函數(function) 中改變**全域變數**的值

```
x = 10
f1 = function()
{
   x <<- 20
}
f1()
x
=> [1] 20
```

奇怪的符號：NA, NaN, NULL

- NA = Not Available : R 軟體的遺失值符號
- NaN = Not a Number

  例如：$\sqrt{-1}$, ∞, $-\infty$
- NULL : 空的 object 或未定義的計算結果

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
> iris[2] = NULL  ; head(iris)
  Sepal.Length Petal.Length Petal.Width Species
1          5.1          1.4         0.2  setosa
```

奇怪的程式：1:10, seq, rep

- `1:10` <=> `c(1,2,3,4,5,6,7,8,9,10)`
- `10:1` <=> `c(10,9,8,7,6,5,4,3,2,1)`
- `seq(1,10,2)` <=> `c(1,3,5,7,9)`
- `seq(2,10,2)` <=> `c(2,4,6,8,10)`
- `seq(10,1,-2)` <=> `c(10,8,6,4,2)`
- `seq(9,1,-2)` <=> `c(9,7,5,3,1)`
- `rep(c(1,2,3),times=2)` <=> `c(1,2,3,1,2,3)`
- `rep(c(1,2,3),times=2,each=3)`

  <=> `c(1,1,1,2,2,2,3,3,3,1,1,1,2,2,2,3,3,3)`

奇怪的程式: x[-3]? M[,-2]? M[-1,]?

```
> x = c(11,12,13,14,15)
> x2 = x[-3] ; x2
[1] 11 12 14 15
> x2 = x[-c(3,5)] ; x2
[1] 11 12 14
> M
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> M[-2, ]
[1] 1 3 5
> M[ ,-3]
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> M[ ,-c(1,3)]
[1] 3 4
```

R 軟體特殊的運算機制(1)

向量之間的加減乘除運算

```
> x = c(1,2,3,4,5)
> y = c(10,20,30,40,50)
> x + y
[1] 11 22 33 44 55
> x - y
[1]  -9 -18 -27 -36 -45
> x * y
[1]  10  40  90 160 250
> y ^ x
[1]      10     400   27000 2560000 312500000
> x + 100
[1] 101 102 103 104 105
```

R 軟體特殊的運算機制(2)

利用向量、矩陣、與陣列指標作快速資料篩選

```
> x = c(1,2,3,4,5)
> y = c(10,20,30,40,50)
> x >= 3
[1] FALSE FALSE  TRUE  TRUE  TRUE
> ( x.index = x >= 3 )
[1] FALSE FALSE  TRUE  TRUE  TRUE
> x[x >= 3]    # 或 x[ x.index ]
[1] 3 4 5
> x[c(F,F,T,T,T)]
[1] 3 4 5
> x[c(3,4,5)]
[1] 3 4 5
> x[y >= 20]
[1] 2 3 4 5
> z = c("boy","girl","boy","boy","girl")
> y[z == "boy"]
[1] 10 30 40
```

R軟體特殊的運算機制(3)

Function 中需使用適當的 cat 機制

```
> f2 = function(x) {
+   mean(x)
+   var(x)  }
> f2(y)
[1] 0.9221541
> f2 = function(x){
+   cat("mean of X = ",mean(x),"\n")
+   var(x)
+ }
> f2(y)
mean of X =  -0.01626653
[1] 0.9221541
```

R 軟體：Input 函數[1]

單一向量輸入：

X = scan(“c:/dir1/sample.txt”)

矩陣資料輸入：文字變數自動轉為 Factor

read.table(“c:/dir1/d1.txt”,header=T)

read.csv(“c:/dir1/d2.csv”,header=T)

read.table(“c:/dir1/d3.txt”,
col.names=c("x1","x2"))

R 軟體：Input 函數[2]

數值變數 input 後不會自動轉為分類變數，
但 colClasses 參數可以預先指定變數輸入後的型態（read.table 與 read.csv 都適用）

```
# bwt gestation parity age height weight smoke
babies = read.csv("l:/data/babies.csv",
   colClasses = c("numeric","numeric","factor",
                  "numeric","numeric","numeric",
                  "factor") )
```

R 軟體：Input 函數[3]

- Excel

```
library(gdata)
xdata = read.xls("c:/dir3/data.xls")
library(xlsx)      # Excel2007 格式
xdata2 = read.xls("c:/dir3/data.xlsx",sheetIndex=1)
```

- SAS

```
library(sas7bdat)
xdata3 = read.sas7bdat("c:/dir3/data.sas7bdat")
```

R 軟體：Output 函數

向量變數之輸出：

```
cat(scores, file="c:/dir2/scores.txt")
write(scores, file="c:/dir2/scores2.txt")
```

資料框架變數之輸出：

```
write.table(X, "c:/dir2/data.txt",row.names = FALSE)
write.csv(X, "c:/dir2/data.csv",row.names = FALSE )
library(xlsReadWrite)
write.xls(mydata, "c:/dir4/mydata.xls",sheet=1)
library(xlsx)    # Excel 2007 格式
write.xlsx(mydata, "c:/dir4/mydata2.xlsx",sheetName="Sheet 1")
```

條件執行：if / switch [1]

1. ifelse

```
y = ifelse( x < 3 , x+10, c(1,2,3))
```

2. if , else if , else

```
if (x < 3) {
  y = x + 10
} else if ( x >= 3 & x < 10) {
  y = x + 20
} else {
  y = x + 30
}
```

條件執行：if / switch [2]

3. switch

```
x = 2
y = switch(x, cos(10), c(1,2,3),list(a=10,b=20))
# y = c(1,2,3)

x = "AB"
y = switch(x, A=10,AB=20,B=30,O=40)
# y = 20
```

迴圈(Loop)：for , while [1]

- for (i *in* 1:10) { … }

  10 次迴圈，i 依序等於 1, 2, …, 9, 10

- range2 = c(1,3,5,11,19)

  for (i *in* range2) { … }

  5 次迴圈, i 依序等於 1,3,5,11,19

- for (xname *in* c("John","Mary","Joe")) { … }

  3 次迴圈, xname 依序等於 "John","Mary","Joe"

迴圈(Loop)：for , while [2]

```
# 1+2+…+ 10
total = 0
for (i in 1:10) {
  total = total + i
}

# 1+3+5+7+9
total = 0
for (i in c(1,3,5,7,9)) {
  total = total + i
}
```

```
fname =
    c("John","Mary","Joe")
for (x in fname) {
  cat("Name: ",x, "\n")
}
```

```
Name:  John
Name:  Joe
Name:  Mary
```

迴圈(Loop)：for ， while [3]

語法：while (留在迴圈的條件){ ... }

```
# 1 + 2 + ... + 9 + 10
total = 0  ; i = 1
while ( i  <= 10)
{
   total = total + i
   i = i + 1
}
```

迴圈 vs. apply 系列函數(1)

```
iris2 = iris[,1:4]      # 只含前 4 個數值變數
for (i in 1:4) {
    m = mean(iris2[ ,i])
    # iris2[,i]是 iris2 的第 i 個變數
    cat("Mean of ",names(iris2)[i],"= ",m,"\n")
}
# 使用 apply 系列函數: sapply
sapply(iris2,mean)
```

```
   Sepal.Length   Sepal.Width Petal.Length   Petal.Width
       5.843333      3.057333      3.758000      1.199333
```

迴圈 vs. apply 系列函數(2)

```
> M = matrix(1:6,2,3)   ;   M
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> apply(M,1,mean)      #每一個橫列(row) 的平均數
[1] 3 4
> apply(M,2,mean)      # 每一個直行(column)的平均數
[1] 1.5   3.5   5.5
```

R function:自訂函數[1]

```
f1 = function(X,Y) {
   z = X + Y
   return(z)
}
w = f1(3,5)
w = f1(X=3,Y=5)
w = f1(Y = 5, X = 3)
a1 = 3 ; a2 = 5
w = f1(X=a1, Y=a2)
# w = 3 + 5 = 8
```

```
f2 = function(X=10,Y=20)
     {
       z = X + Y
       return(z)
     }
w = f2()     # w=10+20=30
w = f2(100)
# 100+20 = 120
w=f2(3,5)    # 3 + 5 = 8
w=f2(Y=5, X=3) # 3+5 = 8
```

R function:自訂函數[2]

```
centre = function(X, type = "m") {
   y = switch(type, m = mean(X), med = median(X))
   return(y)
}
w = c(19, 5, 12, 3, 20)
centre(w,"m")      # 相當於 centre(w)
# [1] 11.8
centre(w,"med")
# [1] 12
```

R function:自訂函數[3]

函數名稱也可以當作另一個函數的參數

```
f1 = function(x, fname) {
  y = fname(x)
  return(y)
}
f1(1:10,mean)   # 傳回5.5 , 此時 fname = mean
f1(1:10,sum)    # 傳回 55, 此時 fname = sum
```

資料處理：條件篩選

```
age2 = age[age <= 18]    # 向量變數
age3 = age[ IQ > 120]    # 向量變數 IQ與age長度相同

iris2 = iris[, c(1,3,4)]  # 或 iris2=[ , -c(2,5)]  )
iris3 = iris[ iris$Sepal.Length < 12,  ]
iris4 = iris[ iris$Sepal.Length < 12, c(1,3,4)]

b2=subset(babies, age > 30, select =
   c(bwt,age,smoke))
b3=subset(babies, smoke == 1, select = -age)
b4=subset(babies, smoke == 1, select = -c(age,smoke))
```

資料處理：轉換 [1]

```
babies$bwt2 = babies$bwt + 10      #產生新變數bwt2
babies$bwt = log(babies$bwt) + 5   #覆蓋舊變數的值

b2 = within(babies, bwt <- bwt + 5)
b3 = within(babies,
            {  age <- age + 5
               new.var <- bwt + age }
)
# 註： within 內需用 <- ，不能使用 = 號
```

資料處理：轉換［2］

數值變數轉為分類變數：cut 函數

```
> score
 [1] 59 63 66 61 59 55 53 56 66 64 59 58 58 65 59 66 63 66 56 60
> score2 = cut(score,breaks=c(0,20,40,60,max(score)),
                    labels = c(1,2,3,4))
> score2
 [1] 3 4 4 4 3 3 2 3 4 4 3 3 3 4 3 4 4 4 3 3
Levels: 1 2 3 4
score3 = cut(score,breaks=4,label=c("A","B","C","D"))
# 等距切成 4 個分類
> score3
 [1] B D D C B A A A D D B B B D B D D D A C
Levels: A B C D
```

機率分配函數：d, p, q, r［1］

語法：

d|p|q|r + 機率分配的英文名稱(. . . , 分配參數)

d => density, p => probability (CDF)

q => quantile, r => random number

常用分配名稱：norm,unif, binom,t,chisq, f

Examples:

```
x1 = rnorm(100,0,1)          # 產生 100 個 N(0,1) 亂數
x2 = runif(50,0,1)           # 產生 50 個 U(0,1) 亂數
x3 = rgamma(50,0.1,0.2)      # 產生 50 個 Gamma(0.1,0.2)亂數
```

機率分配函數：d, p, q, r [2]

```
# Binomial(n=10,p=0.5)
# 計算 Pr(X = 3)=p(3)=f(3)
dbinom(3,10,0.5)
# 計算 Pr(X ≦ 3)=F(3)
pbinom(3,10,0.5)
# N(0,1) Pr(Z <= 1.96)=F(1.96)
pnorm(1.96,0,1)
# 0.9750021
```

```
# q 開頭函數用於查表值
qnorm(0.975,0,1)
# 1.959966 = 1.96
# t(20) 分配 Q97.5 查表值
qt(0.05,20,lower.tail=F)
# 1.724718
# F(5,7)分配 Q97.5 查表值
qf(0.05,5,7,lower.tail=F)
# 3.971523
```

R 的統計模式表達法

語法：被解釋變數 ~ 解釋變數組合

Examples:

假設 y, x1, x2 為數值變數, f1, f2 為分類變數

```
y ~ .                  # 所有其他變數皆為解釋變數
y ~ x1+x2+f1+f2        # 迴歸分析
y ~ f1                 # 一因子設計
y ~ f1 + f2            # 二因子設計
f1 ~ x1 + x2 + f2      # 分類預測模型
```

例：result = tree(Species ~ . , data=iris)

Simple 繪圖

假設 x , y 為向量變數

- plot(x) : Y 座標為 x 元素值, X 座標為 1,2,...,n
- plot(x,y) : X-Y 散佈圖
- hist(x) : 直方圖 (histogram)
- barplot(c(12,20,5),labels=c("A","B","C")) : 長條圖
- stem(x) : 枝葉圖 (stem-and-leaf)
- pie(c(0.2,0.3,0.5),c("A","B","C")) : 圓餅圖
- boxplot(x) ; boxplot(iris[, 1:4]) : 盒鬚圖

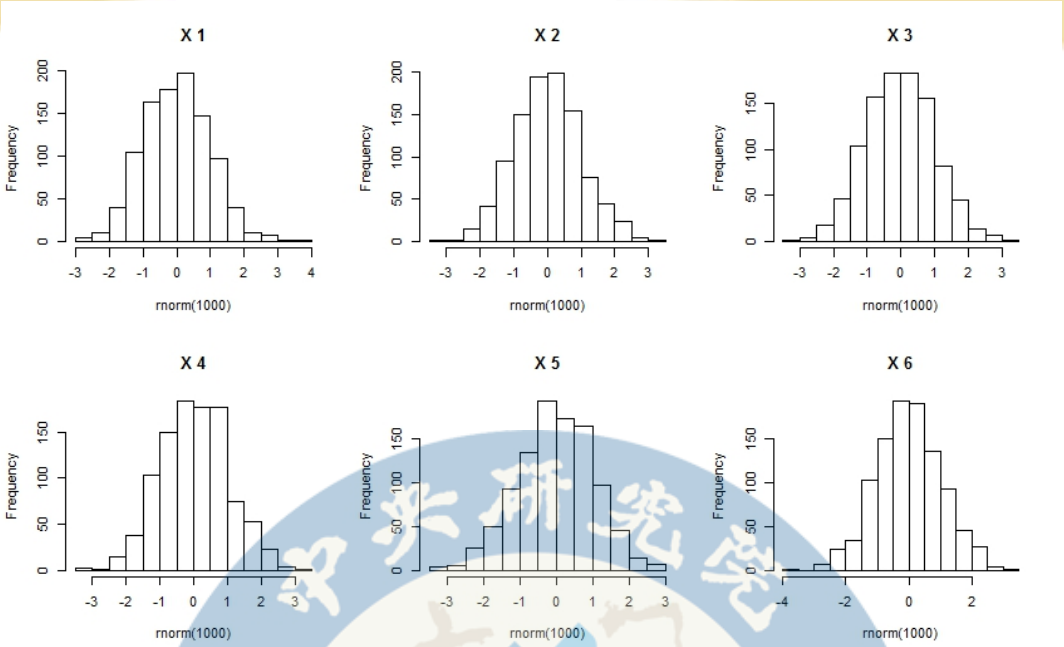
多張圖形放在同一頁[1]

- par(mfrow = c(圖形列數,圖形行數)) : 從第一列開始畫起
- par(mfcol = c(圖形列數,圖形行數)) : 從第一行開始畫起

Example:

```
oldpar = par()                    # 儲存舊的圖形設定
par(mfrow = c(2,3))
for (i in 1:6)
{
   hist(rnorm(1000),  main=paste("X",i)  )
}
par(oldpar)
```

多張圖形放在同一頁[2]



綜合練習:BostonHousing [1]

506 個波士頓人口普查區域的資料, 14變數 (mlbench 套件)

1. **crim** : 犯罪率
2. **zn** : 住宅區比例
3. **indus** : 工業區比例
4. **chas** : Charles River dummy variable
5. **nox**: 空氣中的一氧化氮比例 (ppm)
6. **rm** : 平均每戶的房間數目
7. **age** : 1940 年之前所建的自宅比例
8. **dis** : 距離上班商業中心地點的英哩數
9. **rad** : 可到最近的主要高速公路網路的便利指標
10. **tax** : 房屋與土地相關的收稅金額 (單位: 1 萬美金)
11. **pratio** : 平均 學生 vs. 教師比例
12. **b** : 與種族有關的數據, b=1000(B - 0.63)^2, B 為非洲裔人口比例
13. **lstat** : 低收入戶比例
14. **medv** : 自有住宅房價中位數

綜合練習[2]

```
library(mlbench) ; data(BostonHousing)
BH = BostonHousing ; BH = na.exclude(BH)
summary(BH)            #各變數的彙整敘述統計量
apply(BH,2,is.factor)  #檢查有哪些變數已是factor變數
for (i in 1:length(BH)) # length(BH) = 14
{   cat(names(BH)[i],":",is.factor(BH[i]),"\n")    }
# 將 crim 轉為分類變數 crim2 (factor 變數), 4 個分類 1,2,3,4
BH$crim2 = cut(BH$crim,breaks=4,labels=c(1,2,3,4) )
table(BH$crim2)
# 將 lstat 轉為 2 分類的分類變數 lstat2 : Low, High
BH$lstat2 = cut(BH$lstat,breaks=2,labels=c("Low","High"))
table(BH$lstat2)
```

綜合練習[3]

```
table(lstat2 = BH$lstat2, crim2 = BH$crim2)
                crim2
   lstat2   1   2   3   4
     Low  426   2   1   1
     High  65   8   1   2
tapply(BH$nox,BH$lstat2,mean)          # lstat2 兩個分類的
   平均 nox
tapply(BH$nox,BH$crim2,mean)           # crim2 四個分類的
   平均 nox
tab = tapply(BH$nox,list(BH$lstat2,BH$crim2),mean)
tab
                1       2     3     4
   Low  0.5343185 0.68600 0.597 0.671
   High 0.6603846 0.67825 0.693 0.686
```

綜合練習[4]

```
# 一些交叉表格運算
margin.table(tab,1)   ;  margin.table(tab,2)
prop.table(tab,1)     ;  prop.table(tab,2)
colSums(tab)          ;  rowSums(tab)
colMeans(tab)                ;  rowMeans(tab)
# 統計檢定
t.test(BH$crim,mu=3.5)                    # H0 : μ = 3.5
nox1 = BH$nox[BH$lstat2 == "Low"]
nox2 = BH$nox[BH$lstat2 == "High"]
t.test(nox1,nox2)                         # 雙樣本 t 檢定
chisq.test(table(BH$lstat2,BH$crim2))     # 卡方獨立性檢定
```

綜合練習[5]

```
BH$crim2 =
   cut(BH$crim,breaks=c(0,0.2565,100),labels=c(1,2))
which(names(BH) == "crim")       #  1
which(names(BH) == "crim2")      # 15
BH1 = BH[ , -15]       # BH1: 刪除 crim2, 保留 crim
BH2 = BH[ , -1]        # BH2: 刪除 crim, 保留 crim2
#  迴歸分析
result = lm(crim ~ . , data=BH1)   ;   summary(result)
# 決策樹
result2 = tree(crim2 ~ . , data=BH2 )
result2 ; summary(result2)  ;  plot(result2)  ;  text(result2)
```

[補充]R的應用領域(1)

- Bayesian Inference 貝氏統計方法
- Chemometrics and Computational Physics 化學與物理
- Clinical Trial Design, Monitoring, and Analysis 臨床實驗分析
- Cluster Analysis & Finite Mixture Models 集群分析
- Probability Distributions 機率分配
- Computational Econometrics 計量經濟
- Analysis of Ecological and Environmental Data 生態與環境分析
- Design of Experiments (DoE) & Analysis of Experimental Data 實驗設計
- Empirical Finance 財政實務分析
- Statistical Genetics 基因統計
- Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization 圖形分析
- gRaphical Models in R 圖形模組
- High-Performance and Parallel Computing 高效率運算與平行運算
- Machine Learning & Statistical Learning 機器學習、資料探勘

[補充] R軟體的應用領域(2)

- Medical Image Analysis 醫學影像分析
- Multivariate Statistics 多變量分析
- Natural Language Processing 自然語言分析
- Official Statistics & Survey Methodology 政府統計與調查
- Optimization and Mathematical Programming 函數最佳化
- Analysis of Pharmacokinetic Data 藥物動力學分析
- Phylogenetics 系統發生學
- Psychometric Models and Methods 心理學測量分析
- Reproducible Research 實驗複製分析
- Robust Statistical Methods 強韌統計方法
- Statistics for the Social Sciences 社會科學統計
- Analysis of Spatial Data 空間統計
- Survival Analysis 存活分析、可靠度分析
- Time Series Analysis 時間數列

[補充]R 物件的記憶體大小

R物件記憶體大小估算公式：1個 Vector/變數

資料筆數	佔用記憶體
1 萬	0.0763 Mb
10 萬	0.763 Mb
100 萬	7.63 Mb
1000 萬	76.3 Mb
10000 萬 = 1 億	763 Mb = 0.75 Gb

Example:

1億筆資料、100 變數：0.75 Gb x 100 = 75 Gb


中央研究院
ACADEMIA SINICA
1928

Day2:抽樣設計理論與實作

- 抽樣設計概念與名詞說明
- 抽樣設計類型
- R 軟體的抽樣套件
- 樣本數的決定
- 抽樣
- 抽樣後分析
- 權重修正

抽樣定義

抽樣 (sampling)：從母群體中選取部分元素/基本單位為樣本，並且從選取的樣本推估母體的特徵。

隨機抽樣：藉由機率原則抽取母體元素

非隨機抽樣：由主觀或方便性抽選母體元素

抽樣相關名詞

- 母體：欲調查的對象所成的集合
- 樣本：母體的子集合
- 抽樣單位(Sampling Unit)：抽樣時可被選取的基本單位。如果是簡單隨機抽樣，則抽樣單位可為「人」，如果是分群抽樣，則抽樣單位為「群」，例如台北市的行政區
- 抽樣框架(Sampling Frame): 某抽樣階段的所有抽樣單位所形成的集合或名冊

隨機抽樣的分類

- 簡單隨機抽樣(Simple random Sampling)
- 系統抽樣(Systematic Sampling)
- 分層抽樣(Stratified Sampling)
- 集群抽樣(Cluster Sampling)

簡單隨機抽樣

從N 個基本單位的母體中，抽取出n 個基本單位為樣本，稱為簡單隨機抽樣(Simple random sampling, SRS)、Random sampling或單純隨機抽樣法

所有N 個基本單位被選為樣本的機率(機會)皆必須相同

系統抽樣/等距抽樣

從抽樣名單中，有系統地每間隔若干個抽樣單位，就抽取一個樣本，如此一直等間隔抽樣，即稱為系統抽樣(Systematic sampling)、等距抽樣法、間隔抽樣法(Interval sampling)

各種隨機抽樣方法中最接近簡單隨機抽樣的一種方法，故又稱為**準隨機抽樣**

例如：從學號或電話簿中，每間隔20 名就抽一位。

分層抽樣/分類抽樣

根據某個特定研究變數，將母體中所有基本單位分為數個子集合(subset)，每個子集合可稱之為一層(strata, subset)或一個分類(category)。然後，在每一個子集合中皆採用「簡單隨機抽樣」再抽出數個基本單位，此法稱為分層抽樣(Stratified Sampling)

每一層/分類之內個體的同質性高

各層/分類之間的異質性高

例如: 以學歷、收入、會員等級。。。作為分層變數

分群/集群抽樣

將母體中之所有基本單位依據特定變數(如年齡層、班級、學校、學歷、職業、地區、...等)區分為數個群體(cluster)，針對所有群體進行「簡單隨機抽樣」(抽樣單位為「群」)抽出少數群體，並調查被抽出群體的「所有」成員。

此法稱為集群抽樣(Cluster Sampling)

每個群體內的個體之異質性高

群體間的同質性高(各群體的特性相似)

E.g. 以縣市、行政區、鄰、里、大樓作分群

地區抽樣(Area Sampling)

地區抽樣為集群抽樣之特例

將母體中所有基本單位依據特定地理性變數(如行政區、區域、地區、街道、鄰里、...等)區分為數個群體，每個群體都包含很多基本單位。

分層抽樣 vs. 分群抽樣

	分層抽樣(Stratified)	分群抽樣(Cluster)
同群/同層內的個體	同質性高(特性相似)	異質性高
各層/各群之間	異質性高	同質性高(特性相似)
層/群的抽選	所有各層都被選取	只選少數幾群
各層/各群內個體的選取	只有部分個體被選取	所有個體都被選取
範例變數	收入(低、中、高) 性別、年級、會員等級	縣、市、區、里、鄰、大樓、街道區塊(block)

R軟體的抽樣套件

- R 核心的 sample 函數
- sampling 套件：執行抽樣動作
- survey套件：分析抽樣「後」的資料
- EVER套件：計算估計量的變異數
- simFrame套件：抽樣模擬、不同抽樣計畫的模擬比較
- 抽樣校正/修正(Calibration)

    sampling::calib

    laeken::calibWeights

    survey套件

    reweight套件

其他套件

- TeachingSampling
- samplesize4surveys
- samplingbook
- pps
- mapStats

決定樣本數：簡單隨機抽樣

```
library(samplingbook)

# 目標：估計母體平均數
sample.size.mean(e=0.1,S=0.5,N=Inf,level=0.95)
# S = (estimate of )standard deviation in population
   Sample size needed: 97

# 目標：估計母體比例
sample.size.prop(e=0.03,P=0.5,N = Inf,level = 0.95)
   Sample size needed: 1068
```

決定樣本數：系統抽樣(1)

- 目標：估計 μ

$$n = \frac{N\rho^2}{(N-1)D + \rho^2} \quad , D = \frac{B^2}{4}$$

- 目標：估計 p

$$n = \frac{Npq}{(N-1)D + pq} \quad , D = \frac{B^2}{4}$$

決定樣本數：系統抽樣(2)

目標：估計 μ

```
syssp.mu.size = function(N,B,svar) {
    D = B^2/4
    return(N*svar/( (N-1)*D + svar))
}
# svar = (estimate of) variance of the population
syssp.mu.size(N=2500,B=2,svar=100)
[1] 96.19084
syssp.mu.size(N=150,B=0.5,svar=var(iris$Sepal.Length))
[1] 10.28726
syssp.mu.size(N=150,B=0.1,svar=var(iris$Sepal.Length))
[1] 97.19775
```

決定樣本數：系統抽樣(3)

目標：估計 p

```
syssp.p.size = function(N,B,p=0.5) {
   D = B^2/4   ;   q = 1 - p
   return( N*p*q/( (N-1)*D + p*q) )
}
syssp.p.size(N=5000,B=0.03)
[1] 909.2397
syssp.p.size(5000,0.03,p=0.2)
[1] 622.6771
```

決定樣本數：分層抽樣(1)

```
library(samplingbook)
stratasize(e=0.1, Nh=c(100000,300000,600000), Sh=c(1,2,3))
stratamean object: Stratified sample size determination
type of sample: prop
total sample size determinated: 2568

#random optimal stratified sample
stratasize(e=0.1, Nh=c(100000,300000,600000),
           Sh=c(1,2,3), type="opt")
stratamean object: Stratified sample size determination
type of sample: opt
total sample size determinated: 2395
```

決定樣本數：分層抽樣(2)

```
library(samplingbook)
stratasamp(n=500, Nh=c(5234,2586,649,157))
  Stratum   1   2  3 4
  Size    303 150 38 9

stratasamp(n=500, Nh=c(5234,2586,649,157),
     Sh=c(251,1165,8035,24725), type='opt')
  Stratum  1   2   3   4
  Size    49 112 194 145
```

決定樣本數：分群抽樣(1)

```
library(samplesize4surveys)
N = 1000000 # 母體觀察值數目
nc = 1000   # cluster 數目
M = N/nc    # 平均每個 cluster 包含的觀察值個數
y = c(1:N)  # 欲估計平均數的數值變數
cl = rep(1:nc, length.out=N)    # 分群變數
# intraclass correlation coefficient
rho = ICC(y,cl)$ICC
ss2s4m(N,mu=mean(y),sigma=sd(y),conf=0.95,
       rme=0.03,M=M, by=10, rho)
```

決定樣本數：分群抽樣(2)

```
ss2s4p(N, p=0.5, conf=0.95, me=0.03, M=M,
            by=10, rho)
```

	Deff	NI	m	n2s
2	0.990000000	96	11	1056
3	0.980000000	50	21	1045
4	0.970000000	34	31	1034
5	0.960000000	25	41	1024
6	0.950000000	20	51	1013
7	0.940000000	17	61	1003

.....

Design Effect(設計效果)

- Design Effect (Kish,1965)

假設 z 為某母體參數的估計量, 則設計效果 為 $D_{eff}=D^2(z)$

$$D^2(z)=\frac{\text{Variance of } z \text{ with the complex design}}{\text{Variance of } z \text{ with an unrestricted sample of the same size}}=\frac{V_c(z)}{V_u(z)}$$

則此複雜設計所需的樣本數 $n_c=n\times D^2(z)$

註：若 $D_{eff}<1$, 則此設計比 SRS 有效率

系統抽樣(1)

```
sys.sample = function(N,n) {
   k = ceiling(N/n)
   r = sample(1:k, 1)
   indices = seq(r, r + k*(n-1), k)
      return(indices)
}
( indices = sys.sample(50, 5) )
    [1]  6 16 26 36 46
(indices= sys.sample(1000, 10) )
    [1]  85 185 285 385 485 585 685 785 885 985
sample = mydata[indices , ]
```

系統抽樣(2)

```
library(TeachingSampling)

as.vector(S.SY(100,30))
[1] 26 56 86
as.vector(S.SY(100,30))
[1]  1 31 61 91

indices = S.SY(100,30)
sample = mydata[indices, ]
```

分層抽樣(1)

使用 sampling 套件 strata 函數

```
library(sampling)  ;  data(swissmunicipalities)
xdata=swissmunicipalities
table(xdata$REG)
  1   2   3   4   5   6   7
589 913 321 171 471 186 245
xdata=xdata[order(xdata$REG),]
st=strata(xdata,stratanames=c("REG"),
      size=c(30,20,45,15,20,11,44), method="srswor")
sample = getdata(xdata, st)
```

分層抽樣 (2)

使用 pps 套件的 stratsrs 函數

```
library(pps)
# xdata = xdata[order(xdata$strata), ]
# strata = xdata$strata
strata = c(1,1,1,1,1,2,2,2,3,3,3,3,3,3,3)
# 第 1 層抽2個，第2層抽1個，第 3 層抽 3 個
nh = c(2,1,3)
( indices = stratsrs(strata,nh) )
      [1]  5  2  6 10 12 11
sample = xdata[indices, ]
```

分群抽樣

使用 sampling 套件 cluster 函數

```
library(sampling) ;  data(swissmunicipalities)
xdata =swissmunicipalities
# 抽出 3 群
cl = cluster(xdata,clustername=c("REG"),
                    size=3,method="srswor")
sample = getdata(xdata, cl)
table(sample$REG)
  4   6   7
171 186 245
```

抽樣後的資料分析：survey 套件

survey 套件包含以下主要函數：

- svydesign: 定義抽樣類型
- summary: 說明 survey.design 物件
- svymean, svytotal: 平均數/總數估計
- svyvar: 變異數估計資訊
- svytable: 列聯表與卡方檢定
- svyratio, svyquantile: 比例與百分位數估計
- svyby: 子群體統計量估計

survey 套件：api 資料檔(1)

- data(api) #Academic Performance Index
- apipop, apisrs, apiclus1, apiclus2, apistrat

重要變數：

cds	ID 變數
stype	學校類型(Elementary/Middle/High School)
name	School name (學校名稱, 15 字元)
sname	School name (學校名稱, 40 字元)
snum	School number (學校編號)
dname	District name (地區名稱)
dnum	District number (地區編號)

survey 套件：api 資料檔(2)

emer	percent teachers with emergency qualifications
api00	API in 2000
api99	API in 1999
enroll	學校註冊人數
api.stu	參與考試的人數
fpc	每一個學校所屬層或群的學校總數
pw	每筆記錄的抽樣權重(sampling weights)

survey 套件的執行步驟

Step 1. 使用 svydesign 函數定義抽樣設計類型，並且儲存成屬性為 survey.design 的 R 物件

Step 2. 使用 survey 套件內的其他函數進行相關估計或計算

svydesign:簡單隨機抽樣

```
library (survey)
data(api)
dsrs = svydesign(id = ~1, fpc = ~fpc,
                                  data=apisrs )
# apisrs 所有資料的 fpc 值均為 6194,視為只有一群
dsrs
   Independent Sampling design
   svydesign(id = ~1, fpc = ~fpc, data = apisrs)
summary(dsrs)
```

svydesign:分層抽樣

```
# 使用學校類型(stype)作為分層的依據, 共有 3 層
d.strat = svydesign(id=~1,strata=~stype,fpc=~fpc,
            weights=~pw, data=apistrat)
d.strat
Stratified Independent Sampling design
svydesign(id = ~1, strata = ~stype, weights = ~pw, data =
   apistrat, fpc = ~fpc)
summary(dstrat)
table(apistrat$fpc)  # 3 層各含 755,1018,4421 所學校
        755 1018 4421
         50   50  100
```

svydesign:一階段分群抽樣

```
library(survey) ; data(api)
# 使用地區編號(dnum)作為分群的依據, 共 15 群
dclus1 = svydesign(id=~dnum, weights=~pw,fpc=~fpc,
                   data=apiclus1)
dclus1
1 - level Cluster Sampling design
With (15) clusters.
svydesign(id = ~dnum, weights = ~pw, fpc = ~fpc, data =
   apiclus1)
table(apiclus1$dnum)
 61 135 178 197 255 406 413 437 448 510 568 637 716 778 815
 13  34   4  13  16   2   1   4  12  21   9  11  37   2   4
```

svydesign:兩階段分群抽樣

```
# 先抽出 40 個地區 (dnum), 每個地區再抽出 5 所學校
dclus2 = svydesign(id=~dnum+snum, fpc=~fpc1+fpc2,
                   data=apiclus2)
dclus2        # 母體共有 757 個地區
2 - level Cluster Sampling design
With (40, 126) clusters.
svydesign(id = ~dnum + snum, fpc = ~fpc1 + fpc2, data =
   apiclus2)

length(unique(apiclus2$dnum))
[1] 40
```

svymean

```
svymean(~api00, dsrs, deff=TRUE)
          mean     SE DEff
api00 656.5850 9.2497    1
svymean(~api00, dstrat, deff=TRUE)
          mean     SE   DEff
api00 662.2874 9.4089 1.2045
svymean(~api00, dclus1, deff=TRUE)
         mean     SE   DEff
api00 644.169 23.542 9.3459
svymean(~api00, dclus2, deff=TRUE)
         mean     SE   DEff
api00 670.812 30.099 6.2505
```

只分析符合條件的資料

```
# emer 變數 = 學校中受過急救訓練的教師人數
demer1 = subset(dsrs,emer == 0)
demer2 = subset(dsrs,emer >= 20)

svymean(~api00,demer1)
# 或 svymean(~api000,subset(dsrs,emer == 0) )
         mean     SE
api00 734.29 19.104
svymean(~api00,demer2)
         mean     SE
api00 555.53 18.441
```

信賴區間計算(1)

```
# 母體參數的信賴區間:confint(…) 函數
confint(svymean(~api00, dclus1),level=0.9)
              5 %      95 %
api00 605.4459 682.8929

confint(svytotal(~api00, dsrs))
         2.5 %  97.5 %
api00 3954596 4179179


confint(svyratio(~api.stu, ~enroll,dsrs))
                     2.5 %    97.5 %
api.stu/enroll 0.8056191 0.8450882
```

信賴區間計算(2)

```
svyquantile(~api00, dclus1, c(.25,.5),ci=TRUE)
$quantiles
         0.25 0.5
api00 551.75 652
$CIs
, , api00
            0.25       0.5
(lower 493.2835 564.3250
upper) 622.6495 710.8375
```

一次分析多個統計量(1)

```
svymean(~api.stu+api00+api99,dsrs)
          mean     SE
api.stu 482.51 22.1949
api00   656.59  9.2497
api99   624.68  9.5003

confint(svymean(~api.stu + api00 + api99,dsrs))
            2.5 %   97.5 %
api.stu 439.0088 526.0112
api00   638.4559 674.7141
api99   606.0647 643.3053
```

一次分析多個統計量(2)

```
vars = names(apisrs)[c(12:13,16:18,27)]
svymean(make.formula(vars),dsrs,na.rm=TRUE)
                mean     SE
api00        656.585 9.2497
api99        624.685 9.5003
sch.wideNo     0.185 0.0271
sch.wideYes    0.815 0.0271
comp.impNo     0.335 0.0329
comp.impYes    0.665 0.0329
bothNo         0.345 0.0331
bothYes        0.655 0.0331
pct.resp      77.600 1.8322
```

其他估計函數範例(1)

```
( sep = svyratio(~api.stu, ~enroll, dstrat,
                 separate=TRUE) )
Stratified ratio estimate: svyratio.survey.design2(~api.stu, ~enroll,
   dstrat, separate = TRUE)
Ratio estimator: Stratum == 1L
Ratios=
          enroll
api.stu 0.8518163
SEs=
            enroll
api.stu 0.00703236
Ratio estimator: Stratum == 2L
.............................
```

其他估計函數範例(2)

```
( combined = svyratio(~api.stu, ~enroll, dstrat) )
Ratio estimator: svyratio.survey.design2(~api.stu, ~enroll,
   dstrat)
Ratios=
           enroll
api.stu 0.8369569
SEs=
              enroll
api.stu 0.007757103
```

svyby: 分組/分群彙整(1)

```
svyby(~api99, by=~stype, dsrs, svymean)
  stype    api99       se
E     E 627.3662 11.68752
H     H 592.6400 22.70430
M     M 637.4242 21.48135
svyby(~api99, by=~stype, dstrat, svyratio,
                                  denominator=~enroll)
# denominator 是 svyratio 所需的參數
  stype api99/enroll se.api99/enroll
E     E    1.5256730       0.07705448
H     H    0.4674491       0.03543824
M     M    0.7329906       0.05398206
```

svyby: 分組/分群彙整(2)

```
# 多個分類變數的分組組合
dclus1<-svydesign(id=~dnum, weights=~pw,
            data=apiclus1, fpc=~fpc)
svyby(~api99+api00, ~stype+sch.wide, dclus1, svymean,
            keep.var=FALSE)
      stype sch.wide statistic.api99 statistic.api00
E.No      E       No        601.6667        596.3333
H.No      H       No        662.0000        659.3333
M.No      M       No        611.3750        606.3750
E.Yes     E      Yes        608.3485        653.6439
H.Yes     H      Yes        577.6364        607.4545
M.Yes     M      Yes        607.2941        643.2353
```

補充：系統抽樣的估計(1)

- 平均數估計量 $\hat{\mu}_{sy}$ = 樣本平均數(與 SRS 相同)
- 平均數估計量的樣本變異數

$$\hat{V}(\hat{\mu}_{sy}) = \frac{S^2}{n}\left(\frac{N-n}{N}\right)$$

若 N 未知，則等於 S^2/n

- 母體比例估計量 $\hat{p}_{sy}$ = SRS 的樣本比例 = X/n
- 樣本比例的樣本變異數

$$\hat{V}(\hat{p}_{sy}) = \frac{\hat{p}_{sy}(1-\hat{p}_{sy})}{n-1}\left(\frac{N-n}{N}\right)$$

補充：系統抽樣的估計(2)

- Example:

```
N = 1000000  ; n = 1000
mu.hat = mean(xdata$age)
var.mu.hat = (var(xdata$age)/n)*((N-n)/N)
mu.CI = c(mu.hat-1.96*sqrt(var.mu.hat),
          mu.hat+1.96*sqrt(var.mu.hat) )
p.hat = 300/1000 ;  q.hat = 1 - p.hat
var.p.hat = (p.hat*q.hat/(n-1))*((N-n)/N)
p.CI = c(p.hat-1.96*sqrt(var.p.hat),
          mu.hat+1.96*sqrt(var.p.hat) )
```

抽樣校正：事後分層 (Post-Stratification)(1)

- 根據某個變數的已知母體比例來校正

```
data(api)
# 兩階段分群抽樣
dclus2 = svydesign(id=~dnum+snum, fpc=~fpc1+fpc2,
                   data=apiclus2)
# 根據 stype (學校型態) 的母體比例資訊來校正
pop.types = data.frame(stype=c("E","H","M"),
                       Freq=c(4421,755,1018))
ps.dclus2 = postStratify(dclus2,strata=~stype,
                         population=pop.types)
```

樣本代表性檢定(1)

- 使用卡方適合度檢定(chisq.test)

Example:性別(gender)與地區(area)變數

母體：

性別\地區	北	中	南
男	50000	48000	50000
女	49000	51000	50000

樣本：

性別\地區	北	中	南
男	200	200	200
女	180	190	220

樣本代表性檢定(2)

```
# 北/男, 北/女, 中/男, 中/女, 南/男, 南/女
pop = c(50000,49000,48000,51000,50000,50000)
sample = c(200,180,200,190,200,220)
chisq.test(sample,p=pop,rescale.p=TRUE)

   Chi-squared test for given probabilities

data:  sample
X-squared = 4.6046, df = 5, p-value = 0.466

# Do not reject H0:資料與假設相符
```

抽樣校正：事後分層 (Post-Stratification)(2)

```
svytotal(~enroll, dclus2,na.rm=T)
          total     SE
enroll 2639273 799638
svytotal(~enroll, ps.dclus2,na.rm=T)  # 校正後
          total     SE
enroll 3074076 292584
svymean(~api00,dclus2)
         mean     SE
api00 670.81 30.099
svymean(~api00,ps.dclus2)  # 校正後
      mean     SE
api00  673 28.832
```

抽樣校正：Raking(1)

範例資料檔：FRS (Family Resources Survey)

http://www.restore.ac.uk/PEAS/ex1datafiles/data/ex1.RData

HHINC	家庭每週淨收入
DEPCHLDH	家中孩童數目
ADULTH	家中成人數目
PSU	郵遞區號(主要抽樣單位)
CTBAND	房屋稅(council tax)等級 (1~9)
TENURE	住屋類型(1~4)
GROSS2	抽樣權重

抽樣校正：Raking(2)

- Raking = 多變數反覆加權

```
frs.des = svydesign(id=~PSU,weights=~GROSS2,data=frs)
pop.ctband = data.frame(CTBAND=1:9,
                        Freq=c(515672,547548,351599,291425,
                         266257,147851,87767,9190,19670) )
pop.tenure = data.frame(TENURE=1:4,
                    Freq=c(1459205,493237,128189,156348))
frs.raked =
      rake(frs.des,sample=list(~CTBAND,~TENURE),
      population=list(pop.ctband,pop.tenure))
```

抽樣校正：Raking(3)

```
svymean(~HHINC,frs.raked)
        mean     SE
HHINC 483.09 7.5781
svymean(~HHINC,subset(frs.raked, DEPCHLDH > 0))
        mean     SE
HHINC 611.21 12.541
svymean(~HHINC,subset(frs.raked, DEPCHLDH > 0 &
              ADULTH==1))
        mean     SE
HHINC 276.56 8.4417
```


中央研究院
ACADEMIA SINICA
1928


中央研究院
ACADEMIA SINICA
1928

Day 3: 調查資料探勘與實作

- 資料探勘：運用自動或半自動化的方式，從(大量)資料中挖掘出有用的資訊/規則/模式。

- 資料探勘的特色：
  1. 自動化或半自動化
  2. 運算結果很容易詮釋
  3. 可與傳統統計技術互補

傳統調查資料分析 vs. 資料探勘

- 計算母體參數估計量: 平均數、比例等
- 計算參數的信賴區間
- 計算變數之間的相關性(e.g. 卡方檢定)
- 迴歸分析、Logistic 迴歸分析、ANOVA
- 多變量分析：包含集群分析、主成份分析、因素分析、典型相關分析
- SEM 結構方程式模型

+ Data Mining

- 決策樹、類神經網路、SVM、關連規則分析等技術

資料探勘常用功能分類

- 分類(Classification)
- 分群(Clustering)或分割(Segmentation)
- 規則(Association)或序列(Sequence)
- 預測(Prediction)或估計(Estimation)

分類(Classification)常用方法

- Logistic 迴歸模式, Polytomous 模式, Ordinal Logistic 模式
- 判別分析(Discriminant Analysis)
- 分類樹(Classification Tree)
- 隨機森林(Random Forest)
- 類神經網路(Artificial Neural Network)
- 支持向量機(SVM:Support Vector Machine)
- 貝式分類器(Bayesian Classifier)

集群(Clustering)

Clustering：事前不知道詳細的分類

- K-Means 集群分析
- Hierarchical 集群分析
- Density-based 集群分析
- SOM (Self-Organizing Map)

規則(Association)或序列(Sequence)

- 關聯分析(Association Analysis)
- 序列探索(Sequence Discovery)

預測(Prediction)或估計(Estimation)

- 一般統計估計與檢定
- 迴歸分析
- 時間數列
- Logistic 迴歸模型
- 迴歸樹 (Regression Tree)
- 類神經網路(ANN)

訓練樣本 vs. 測試樣本

```
# 使用Splitdata 函數 (在輔助教材 Sample Codes 中)
out = Splitdata(iris, 0.9) # 訓練樣本 90%,測試樣本 10%
Dtrain = out$train        # 訓練樣本
Dtest= out$test    # 測試樣本

Xtrain = notY(Dtrain,"Species")]
Ytrain = Dtrain[,"Species"]

Xtest = notY(Dtest,"Species")
Ytest = Dtest[,"Species"]
```

Confusion Matrix: 混淆矩陣

```
confmatrix = function(Y,Ypred)
{
  t1 = table(Y,Ypredict=Ypred)
  print(t1)
  p = sum(diag(t1))/sum(t1)*100
  cat("\n\n預測正確率 = ",p,"% \n")
}

 # Example:
library(tree)
 result = tree(Species ~ ., data=iris)
 p1 = predict(result,type="class")
confmatrix(iris$Species, p1)
```

計算結果：

Y	Ypredict setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	47	3
virginica	0	1	49

預測正確率 = 97.33333 %

決策樹(Decision Trees)

依照功能區分：

1. 分類樹(Classification Tree)
2. 迴歸樹(Regression Tree)

常用的決策樹：

1. CART（分類/迴歸）
2. CHAID（分類）
3. QUEST(分類/迴歸)
4. C4.5（分類/迴歸）
5. Random Forest（分類/迴歸）
6. Mob, Cubist（迴歸）

決策樹建構法則

- 從上而下、從左而右，任意一個節點

Step 1. 找出此節點的最佳分割「變數」
Step 2. 找出此最佳變數的最佳分割點
Step 3. 產生左右兩個分支或更多分支
Step 4. 若滿足停止條件，則停止分支動作
Step 5. 必要時，刪剪決策樹以求最佳化

四種決策樹的比較

特色	QUEST	CART	CHAID	C4.5
變數型態	連續/分類	連續/分類	分類	連續/分類
分支數目	2	2	2 以上	連續: 2 以上 分類: 2
分支變數	單/多變數	單/多變數	單變數	單變數
分割規則	卡方/F 檢定	Gain ratio	卡方檢定	Gain Ratio
可設定分類先驗機率	O	O	X	X
樹的修剪	測試樣本 或 交叉驗證	測試樣本 或 交叉驗證	Stopping Rules	同時分支與刪減
遺失值	內插法 或代理變數	代理變數	分出遺失值的支幹	使用機率加權

R軟體中的決策樹函數

- CART：tree 套件, rpart 套件
- C4.5：RWeka 套件(J48 分類法)
- CHAID：CHAID 套件(chaid 函數)，
  R-Forge 網站(尚未正式發佈)
- QUEST：無
- Random Forest：randomForest 套件
- mob 與 Cubist：model-based trees

決策樹 function 通用語法

result = 函數名稱(Y ~ X_1+X_2+...+X_k, 其他選項)
# Y 若為分類變數，需為 Factor 型態
pred = predict(result , new_X_data
,type="class")

# 混淆矩陣 (confusion matrix)
ctable = table(Y 真值向量, pred)

# 預測正確率
sum(diag(ctable))/sum(ctable)

R分類方法通用的預測形式

1. 直接輸出分類值

```
Ypred = predict(result,newx,type="class")
```

2. 輸出小數：需四捨五入

```
Ypred = round(predict(result,newx))
```

3. 輸出觀察值落到各分類的機率：特別處理

e.g.

	逃稅	不會逃稅
1.	0.88	0.12
2.	0.04	0.96
.....		

CART 範例：tree/party 函數

```
library(tree)
head(iris)
result = tree(Species ~ . , data=iris)
result
names(result)
plot(result)
text(result)
Ypred = predict(result,type="class")
confmatrix(iris$Species, Ypred)

library(rpart)
result2 = rpart(Species ~ . ,data=iris); plot(result2); text(result2)
Ypred2 = predict(result2,type="class")
confmatrix(iris$Species, Ypred2)
result2$variable.importance
```

CHAID 決策樹：安裝

1.請至
http://r-forge.r-project.org/R/?group_id=343
http://steve-chen.net/RTM2014/ 下載
CHAID_0.1-1.zip

2. 先在R軟體安裝 partykit 套件
3. 請到 R 軟體上端的選單選擇「程式套件」
=>「用本機的 zip 檔案來安裝程式套件」
，安裝 CHAID_0.1-1.zip

CHAID 決策樹：程式範例

```
data(iris)

SepL = cut(iris$Sepal.Length,breaks = 10)
SepW = cut(iris$Sepal.Width, breaks = 10)
PetL = cut(iris$Petal.Length,breaks = 10)
PetW = cut(iris$Petal.Width, breaks = 10)

SepL = ordered(SepL)
SepW = ordered(SepW)
PetL = ordered(PetL)
PetW = ordered(PetW)

iris2=data.frame(SepL,SepW,PetL,PetW,Species=iris$Species)
head(iris2)
```

CHAID 決策樹：程式範例(續)

```
library(CHAID)
result = chaid(Species ~ ., data = iris2)

plot(result)
print(result)

Ypred = predict(iris.chaid, newdata=iris2)
confmatrix(iris2$Speices, Ypred)
```

C4.5（J48 in Rweka）範例

```
#J48: 請先安裝 party 與 Rweka 套件
library(RWeka)
result = J48(Species ~ . , data = iris)
result
summary(result)
plot(result)
Ypred = predict(result)
confmatrix(iris$Species, Ypred)
```

RandomForest 範例

```
library(randomForest)
set.seed(71)
result <- randomForest(Species ~ . , data=iris ,
                       importance=TRUE, proximity=TRUE)
print(result)

round(importance(result), 2)

names(result)
(  t = result$confusion  )

sum(diag(t))/sum(t)
```

類神經網路

- 前饋式 ANN
- 倒傳遞 ANN
- MLP 多層感知機

類神經網路結構

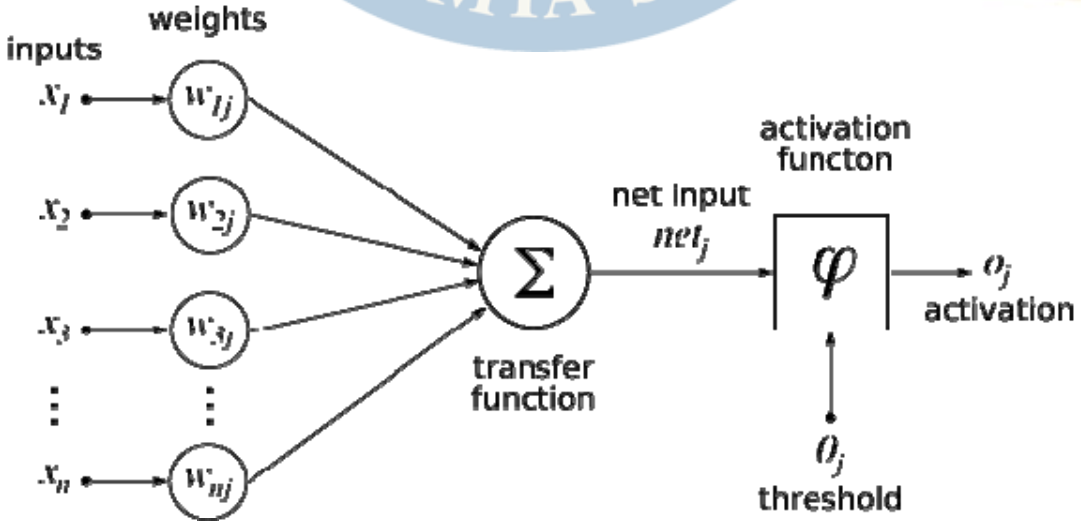
- Input: $X_1, X_2, ..., X_k$ ， Output: Y
- 一個 ANN 具有輸入層、隱藏層、輸出層
- Y 的估計量 $= f(\sum w_i X_i - \theta)$

  其中 w_i 為各 X_i 的權重

  θ 為閥值 (threshold)

  $f(.)$ 或 $\phi(.)$為某個非線性函數

類神經網路範例圖



前饋式網路：nnet 範例

```
library(nnet)
result = nnet(Species ~ . , data = iris, size=3)

Ypred = predict(result,iris[,1:4],type="class")

confmatrix(iris$Species, Ypred)

#( t = table(Y = iris$Species,Ypred ) )
#cat('正確分類比例 = ',100*sum(diag(t))/sum(t),' % \n')
```

倒傳遞網路：neuralnet 範例

```
library(neuralnet)
y=as.numeric(iris$Species)
iris2=data.frame(iris[,1:4],y)
result = neuralnet(y~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,
    hidden=c(3,2),data=iris2)

result

cf=compute(result,iris2[,1:4])
Ypred = round(cf$net.result)
confmatrix(iris$Species, Ypred)

# (t=table(Y=iris2$y,Ypred))
# cat("預測正確率 = ", sum(diag(t))/sum(t), "\n" )
```

SVM 與其他分類方法

- SVM
- Naïve Bayes

e1071 套件: SVM 範例

```
out = Splitdata(iris)  ; iris.Train = out$train  ; iris.Test =
    out$test

library(e1071)
result = svm(Species ~ . ,data=iris.Train)
print(result)
summary(result)

Ypred = predict(result, iris.Train)
confmatrix(iris.Train$Species,Ypred)

Ypred = predict(result, iris.Test)
confmatrix(iris.Test$Species,Ypred)
```

e1071套件: Naïve Bayes 範例

```
out = Splitdata(iris) ; iris.Train = out$train ; iris.Test = out$test

library(e1071)

result = naiveBayes(Species ~ . ,data=iris.Train)
print(result)
summary(result)

Ypred = predict(result, iris.Train)
confmatrix(iris.Train$Species, Ypred)

Ypred = predict(result, iris.Test)
confmatrix(iris.Test$Species, Ypred)
```

集群分析(Cluster Analysis)

找出觀察值之間潛在的群體

- K-Means 集群分析
- 階層式集群分析
- 模糊集群分析
- SOM 類神經網路

集群分析：擷取數值變數

```
NumVars = function(data) {
  nc = ncol(data)
  keep = numeric(nc)
  j = 0
  for (i in 1:nc) {
   if (is.numeric(data[,i])) {
     j = j + 1
     keep[j] = i
   }
  }
  return(as.matrix(data[,keep]) )
}
```

```
iris2 = NumVars(iris)
head(iris2)
```

```
 Sepal.Length Sepal.Width
    Petal.Length Petal.Width
[1,]          5.1         3.5          1.4         0.2
[2,]          4.9         3.0          1.4         0.2
.....................................................
```

K-Means 集群分析

```
library(mlbench)
data(BostonHousing2)
B2 = NumVars(BostonHousing2)
B2 = na.exclude(B2)  # 去除遺失值
ncluster = 5
result = kmeans(B2,centers=ncluster,nstart=10)
result
result$cluster      # 顯示各觀察值的分群
table(result$cluster)     # 計算各群的數目
```

K-Means: iris 資料檔

```
# iris data
iris2 = NumVars(iris)
ncluster = 3
result = kmeans(iris2,centers=ncluster,nstart=10)
result
result$cluster  # 顯示各觀察值的分群
table(result$cluster)  # 計算各群的數目

library(car)
Y = as.numeric(iris$Species)
Y2 = recode(Y,"1=2;2=1;3=3")  # 使用 recode 轉碼
confmatrix(Y2,result$cluster)
```

階層式集群分析（ Hierarchical ）

(1) 使用 hclust 函數

```
# 使用 BostonHousing2 資料
result = hclust(dist(B2)^2, method = "complete")

# 另外還有 "average" 與 "centroid" 兩種方法
result
plot(result)
rect.hclust(result, k=2, border="red")
```

階層式集群分析(Hierarchical)

(2) Agglomerative Hierarchical Clustering (凝聚階層集群)：
cluster 套件的 agnes 函數

```
# 使用 BostonHousing2 資料
library(cluster)
result = agnes(B2)
summary(rsult)
pltree(result)
```

fpc 套件 pamk 決定最佳分群數目

```
library(fpc)
# 試探 2 ~ 6 群
pamk(B2,2:6)
```

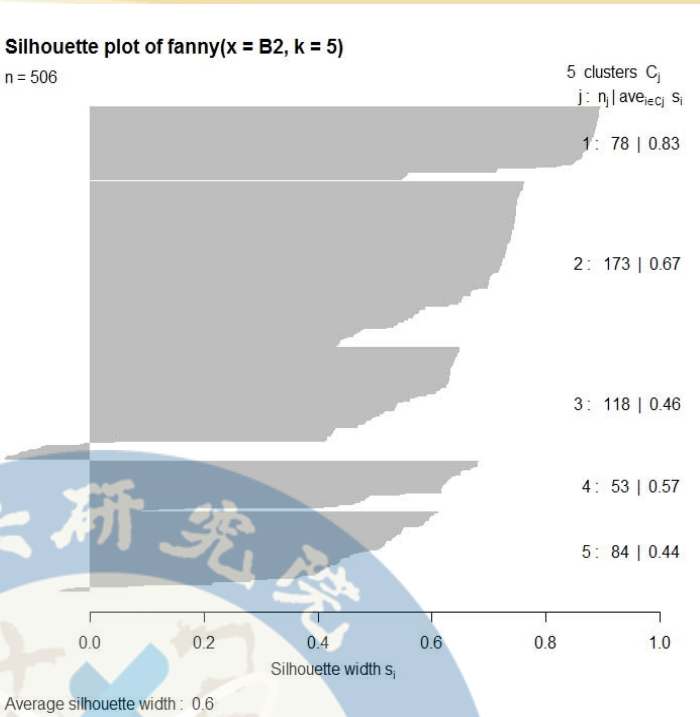
Output:

```
$pamobject
Medoids:
     ID tract     lon      lat medv cmedv
    crim zn indus   nox    rm  age
468 468  1101 -71.066 42.1780 19.1  19.1
    4.42228  0  18.1 0.584 6.003 94.5
223 223  3733 -71.125 42.2134 27.5  27.5
    0.62356  0   6.2 0.507 6.879 77.7
       dis rad tax ptratio      b lstat
468 2.5403  24 666    20.2 331.29 21.32
223 3.2721   8 307    17.4 390.39  9.93
```

```
Clustering vector:
  1   2   3   4   5   6   7   8   9  10
    11  12  13  14  15  16  17  18  19
  1   1   1   1   1   1   1   1   1   1
    1   1   1   1   1   1   1   1   1
 20  21  22  23  24  25  26  27  28  29
    30  31  32  33  34  35  36  37  38
  1   1   1   1   1   1   1   1   1   1
    1   1   1   1   1   1   1   1   1
.....................
495 496 497 498 499 500 501 502 503 504
    505 506
  1   1   1   1   1   1   1   1   1   1
    1   1


.....................

$nc
[1] 2
# 最佳 cluster 數目：2
```

模糊集群分析(1)

```
library(cluster)
result = fanny(B2,5)
result
summary(result)
plot(result)
```



模糊集群分析(2)

Silhouette 指標/圖形

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Which can be written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

From the above definition it is clear that

$$-1 \leq s(i) \leq 1$$

a(i) = 第 i 個觀察值跟同一個 cluster 內其他觀察值的距離

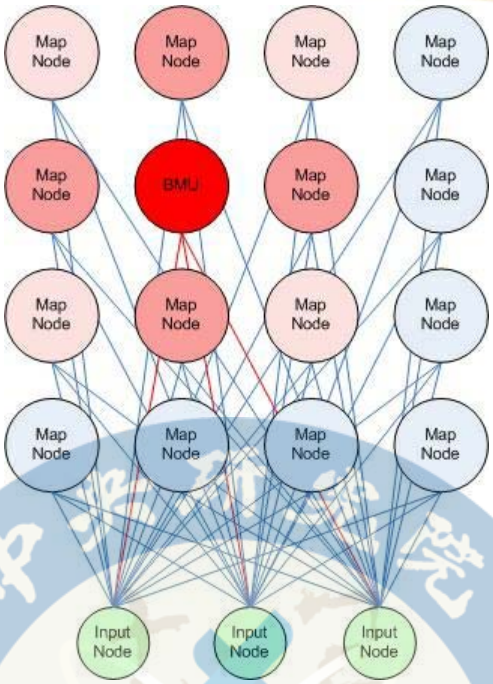
b(i) = 第 i 個觀察值跟「最靠近」的其他 cluster 的距離

若 s(i) 靠近 1.0 => 第 i 個觀察值被分群的成效很好

若 s(i) 靠近 -1.0 =>第 i 個觀察值的分群成效很差，應該被分在鄰近那個 cluster

若 s(i) 靠近 0 => 第 i 個觀察值落在兩個 cluster 的中間

SOM: Self-Organizing Map

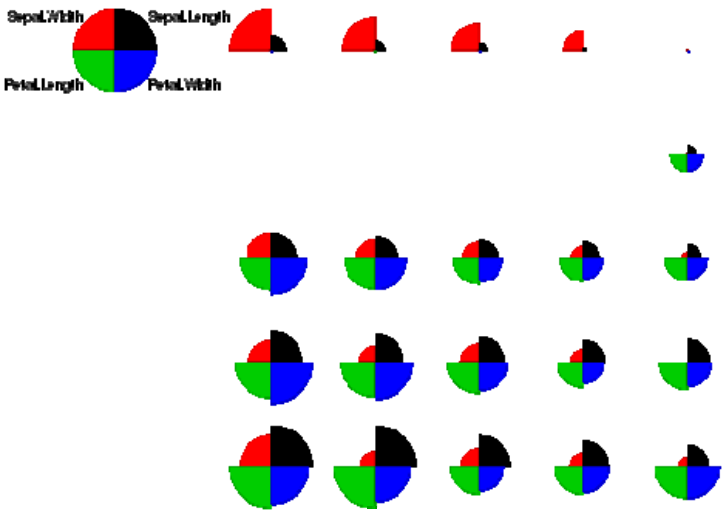


SOMbrero 範例

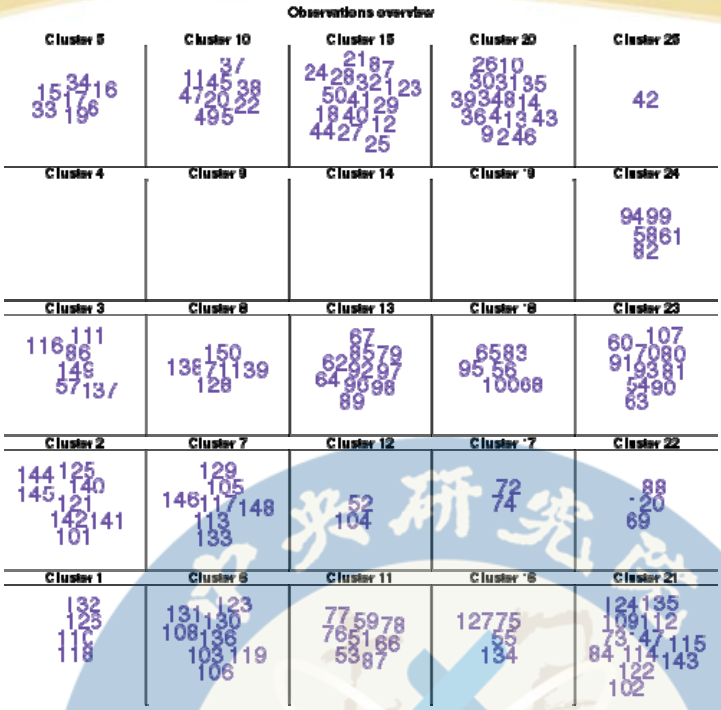
```
# 先安裝 slam, knitr, igraph 套件
install.packages("SOMbrero", repos="http://R-Forge.R-project.org")
library(SOMbrero)
set.seed(4031730) # run the SOM algorithm with verbose set to TRUE
result = trainSOM(x.data = iris[, 1:4], verbose = TRUE, nb.save = 5)
result$clustering
table(result$clustering)
summary(result)
oldpar=par()
par(mfrow = c(2, 2))
plot(result, what = "obs", type = "color", variable = 1, print.title = TRUE, main = "Sepal length")
plot(result, what = "obs", type = "color", variable = 2, print.title = TRUE, main = "Sepal width")
plot(result, what = "obs", type = "color", variable = 3, print.title = TRUE, main = "Petal length")
plot(result, what = "obs", type = "color", variable = 4, print.title = TRUE, main = "Petal width")
par(oldpar)

plot(result, what = "obs", type = "boxplot", print.title = TRUE)
plot(result, what = "obs", type = "names", print.title = TRUE)
predict(result, iris[1, 1:4])
result$clustering[1]
```

SOMbrero Plot (1)

SOMbrero Plot (2)



SOMbrero Plot (3)



關聯規則分析

- 原理：根據 support 跟 confidence 挑選規則
- 資料格式：

	牛奶	餅乾	可樂	蛋	啤酒
t1	1	1	0	0	1
t2	1	0	1	1	1

.....

- 規則：X -> Y （其中 X 與 Y 為物件的集合）

  例如：{ 牛奶, 餅乾 } -> { 啤酒 }

- **support** = X 與 Y 同時出現的次數 / 所有交易數
- **confidence** = X 與 Y 同時出現的次數 / X 出現次數

關聯分析： arules 套件

```
library(arules)
data("Adult")  # Adult 已經是 0/1 格式的資料檔
inspect(Adult[1:4, ]) # 檢視前4筆交易
rules = apriori(Adult,
parameter = list(supp = 0.5, conf = 0.9,
target = "rules"))
summary(rules)
inspect(head(sort(rules, by = "support"), n = 100))
```

arules: transaction 資料轉換

```
library(arules)
# iris data
# 所有數值變數需先轉成分類變數 (factor 或 ordered factor)
SepL = ordered(cut(iris$Sepal.Length,breaks = 4))
SepW = ordered(cut(iris$Sepal.Width, breaks = 4))
PetL = ordered(cut(iris$Petal.Length,breaks = 4))
PetW = ordered(cut(iris$Petal.Width, breaks = 4))

iris2=data.frame(SepL,SepW,PetL,PetW,Species=iris$Species)
iris3 = as(iris2, "transactions")

rules = apriori(iris3,parameter = list(supp = 0.2, conf = 0.6, target = "rules"))
summary(rules)
inspect(head(sort(rules, by = "support"), n = 100))
```

應用與實作

- 法國成人資料檔
- marketing 資料檔
- credict card 資料檔