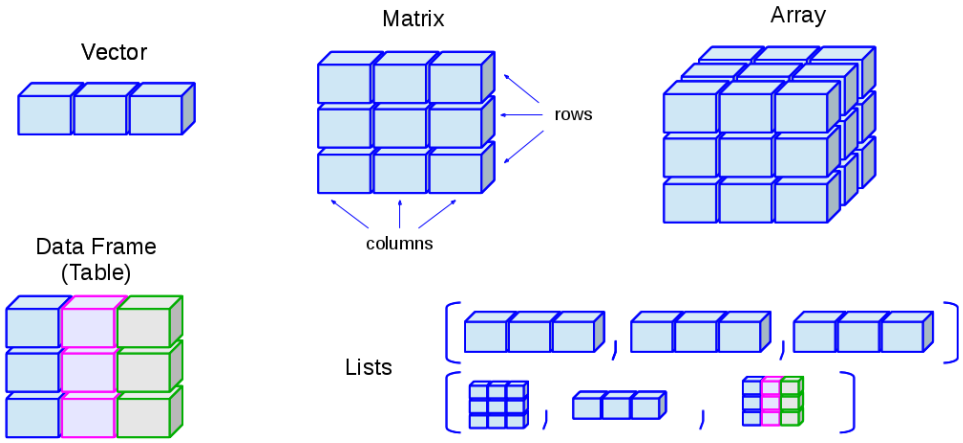
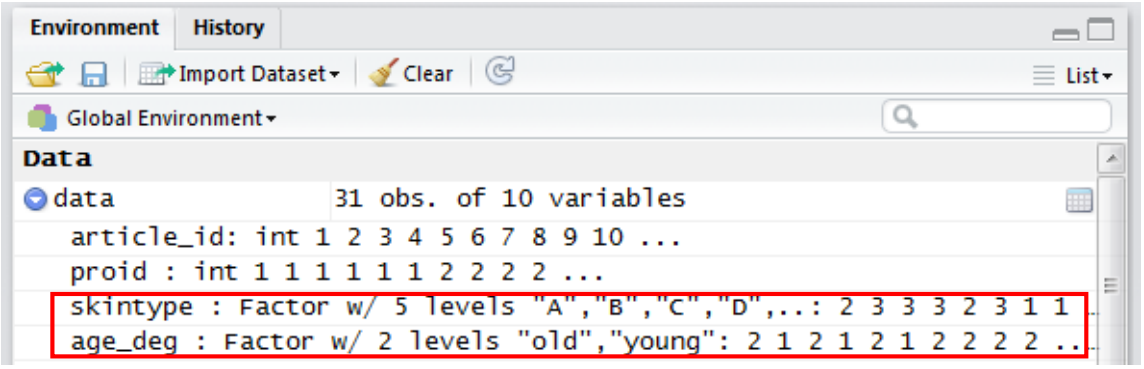
➢ R 的資料物件



Data Frame: 以 List 方式儲存, 但長度相同

List: 每個元素的資料類型及長度可以不相同

R 裡的 factor 是較特別的 vector, 用來儲存類別變數



若資料為類別資料, 匯入資料時 R 會自動將類別型資料轉成 Factor

ex: 膚質為 A,B,C,D,E 五類, 年齡層為 old, young 兩類

➢ matrix( ) : create matrix

建立一個 6 橫列,3 直行, element 為 1~18 的 matrix, 欄位取名為 ID, 年資, 年齡如下

ID	Num	Age
1	7	13
2	8	14
3	9	15
4	10	16
5	11	17
6	12	18

```
a = matrix( 1:18 , nrow=6 )
colnames(a) = c( 'ID' , 'Num' , 'Age' )
```

output:

```
     ID  Num  Age
[1,]  1    7   13
[2,]  2    8   14
[3,]  3    9   15
[4,]  4   10   16
[5,]  5   11   17
[6,]  6   12   18
```

- **matrix 資料讀取**

```
a[1,]                    #取 matrix  a 的第一橫列
```

output:

```
[1]  1  7 13
```

```
a[,2]                    #取 matrix  a 的第二直行
```

output:

```
[1]  7  8  9 10 11 12
```

- **as.data.frame( ) 將資料轉換成 data frame 格式(類似 table 的格式)**

```
df = as.data.frame(a)        #將 matrix a 轉換成 data frame 格式
df$ID
#轉換成 data frame 格式就可以用$加欄位名稱輕易的將該欄位的值 print 出
```

output:

```
> df$ID
[1] 1 2 3 4 5 6
```

- **subset(data, select = c(欄位 1, …, 欄位 n), subset=(條件))**

資料篩選, 效果和 SQL 語法 "select 欄位 from data where 條件" 一樣

從 matrix a 中篩選 Num 欄位大於 9 的資料

```
#只 output column ID 和 Age 兩 column
w1 = subset(a,select=c(ID,Age),subset=('Num'>9))
```

output:

```
     ID  Age
[1,]  4  16
[2,]  5  17
[3,]  6  18
```

```
#要 output 全部欄位就不需再 select column
w2 = subset(a,subset=('Num'>9))      #條件可直接用欄位名稱
w3 = subset(a,subset=(a[,2]>9))      #也可指定欄位 index
```

output:

```
     ID Num Age
[1,]  4  10  16
[2,]  5  11  17
[3,]  6  12  18
```

- **資料分類: 將 matrix a 的 Age 欄位小於 15 的值歸類為類別 A, 介於 15~17 的歸類為 B, 大於 17 的歸類為 C**

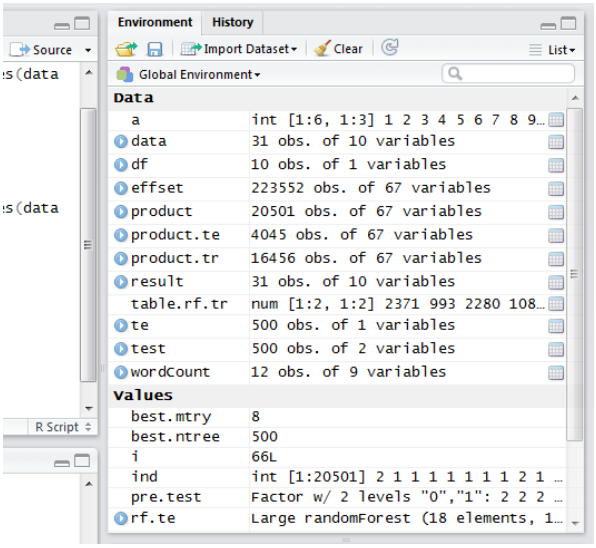
```
deg= length(a)              #建立與 a 橫列數一樣的空 vector (6x1)
deg[a[,3]<15] = 'A'         # a[,3]<15 output:true,true,false,...false
deg[a[,3]>=15 & a[,3]<18]= 'B'
deg[a[,3]>=18]= 'C'
deg                         # output: A A B B B C
cbind(a,deg)                #a 與 deg 做欄合併
```

output:

```
     ID  Num  Age  deg
[1,] "1" "7"  "13" "A"
[2,] "2" "8"  "14" "A"
[3,] "3" "9"  "15" "B"
[4,] "4" "10" "16" "B"
[5,] "5" "11" "17" "B"
[6,] "6" "12" "18" "C"
```

- **rm( ) : remove 變數、data frame、matrix…..等 objects**

跑 R 程式時所有 object 會顯示在 Global Environment 視窗裡, 點選 Clear 可以清除全部
但如果只想清除某一個物件, 就可以用 rm(物件名稱)來移除特定物件



```
rm(effset)                    #移除上圖中 effset 這個 data frame
rm(best.mtry)                 #移除上圖中 best.mtry 這個變數
```

output:

```
> best.mtry
[1] 8                         ←原本 best.mtry = 8
> rm(best.mtry)               ←移除變數 best.mtry
> best.mtry                   ←移除後再看看 best.mtry 這個變數是什麼
Error: object 'best.mtry' not found   ←顯示沒有 best.mtry 這個變數
```

- aggregate(x, by, FUN, ...) :對資料 x 做分群後各別運算 (檔案:income.csv)

```
#計算 Taipei,Taichung, Kaohsiung 男女性的平均收入
income_mat = read.table('D:/income.csv',sep=',',header = TRUE)
income_avg= with(income_mat,aggregate(income_mat[,4], by = list(city,g
ender),FUN=mean))
# income_mat[,4] → 收入欄位
# by = list(city,gender) → 對 city 和 gender 欄位做分群
# FUN=mean → 分群後取平均
```

	Group.1	Group.2	x
1	female	Kaohsiung	37315.00
2	male	Kaohsiung	42443.50
3	female	Taichung	34793.75
4	male	Taichung	34467.20
5	female	Taipei	35167.00
6	male	Taipei	31275.67

- describeBy(y, x, mat=TRUE) :以 x 做分群做 y 的敘述統計分析

```
install.packages("psych")
library("psych")
des.mat = describeBy(income_mat$income,income_mat$city,mat=TRUE)
```

row.names	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
11	1	Kaohsiung	1	6	39024.50	5530.188	39858.5	39024.50	6765.104	31514	46118	14604	-0.13623769	-1.807453	2257.690
12	2	Taichung	1	9	34612.33	9878.924	37914.0	34612.33	14092.113	23300	47419	24119	-0.01216647	-1.989130	3292.975
13	3	Taipei	1	5	32832.20	5119.949	31525.0	32832.20	5596.815	27002	40274	13272	0.30614565	-1.742240	2289.711

➢ grep("欄位名稱", colnames(data)) : 找特定欄位名稱在 data 中的第幾欄

eff.csv 為不同膚質(skintype),不同年齡層(age_deg)對各個產品(proid)的心得斷詞, 如下表

第一橫列表示膚質 B, 年齡 young 的人用產品 1 有 expensive, good, whitening, smell_good 的感受

example 1: 匯入 eff.csv

example 2: 欄位 expensive 和 smell_good 在第幾欄?

example 3: 想知道每個產品在不同膚質/年齡層提到的詞頻 (ex: skintype 為 C, age_deg 為 young, proid 為 2 的所有心得 expensive/cheap/good/bad/whitening/smell_good 提到幾次?)

article_id	proid	skintype	age_deg	expensive	cheap	good	bad	whitening	smell_good
1	1	B	young	1	0	1	0	1	1
2	1	C	old	0	1	1	0	1	0
3	1	C	young	0	1	1	0	0	1
4	1	C	old	0	1	1	0	0	1
5	1	B	young	0	1	1	0	0	0
6	1	C	old	1	0	0	1	1	1
7	2	A	young	1	0	1	0	1	0
8	2	A	young	1	0	0	1	0	0
9	2	C	young	0	1	0	1	1	1
10	2	C	young	1	0	0	1	0	1
11	2	C	young	0	1	1	0	1	1
12	2	E	old	0	1	1	0	0	0
13	2	D	young	0	1	0	1	0	1
14	2	C	old	1	0	1	0	1	0
15	2	C	old	1	0	1	0	1	1
16	2	D	young	1	0	1	0	1	1
17	2	A	young	0	1	0	1	0	0
18	3	A	old	0	1	1	0	1	0
19	3	A	old	1	0	1	0	0	0
20	3	A	old	1	0	0	1	1	1

```
#example 1
data = read.table('D:/eff.csv', sep=',' , header = TRUE)
# 檔案存放路徑: D:/eff.csv , 欄位用,分隔, 資料第一橫列為欄位名稱

#example 2
grep("expensive", colnames(data))          #expensive 欄位在 data 中第幾欄?
grep("smell_good", colnames(data))         #smell_good 欄位在 data 中第幾欄?
```

example 2 output:

```
> grep("expensive", colnames(data))
[1] 5
> grep("smell_good", colnames(data))
[1] 10
```

```
#example 3 透過 aggregate, 對 proid/skintype/age_deg 做 gorup by 後再做欄加總
wordCount= with(data,aggregate(data[,grep("expensive", colnames(data)):
grep("smell_good", colnames(data))], by = list(proid,skintype,age_deg),
FUN=sum))
#對 data 中的 expensive~smell_good 欄位依 proid,skintype,age_deg 做分群後做加總
```

output:

	Group.1	Group.2	Group.3	expensive	cheap	good	bad	whitening	smell_good
1	3	A	old	2	2	2	2	3	2
2	1	C	old	1	2	2	1	2	2
3	2	C	old	2	0	2	0	2	1
4	3	C	old	0	1	1	0	1	1
5	2	E	old	0	1	1	0	0	0
6	2	A	young	2	1	1	2	1	0
7	1	B	young	1	1	2	0	1	1
8	3	B	young	4	2	4	2	2	3
9	1	C	young	0	1	1	0	0	1
10	2	C	young	1	2	1	2	2	3
11	3	C	young	2	1	1	1	1	1
12	2	D	young	1	1	1	1	1	2

- **merge(x , y): join 兩個 data frame**

```
emp = read.table('D:/employee.csv',sep=',',header = TRUE)
dep = read.table('D:/department.csv',sep=',',header = TRUE)
merge(emp,dep)              #只輸入兩 table 名稱會自動找兩 table 相同欄位名稱當做 key
merge(emp,dep, by='department')     #兩 table 依 department 欄位 join
```

如果想用某欄位做 join, 但兩個 table 欄位命名不一樣的時候可用下面的方法

```
emp = read.table('D:/employee.csv',sep=',',header = TRUE)
dep2 = read.table('D:/department2.csv',sep=',',header = TRUE)
merge(emp,dep2, by.x= 'department', by.y='dep')
#emp 裡部門欄位名稱為 department, dep2 裡部門欄位名稱為 dep
```

其他:

```
merge(dataset1, dataset2, by="countryID", all.x=TRUE)  →left join
merge(dataset1, dataset2, by="countryID", all.y=TRUE)  →right join
```

➢ randomForest(x,y, ntree = ?, mtry = ?,importance=TRUE,…) 隨機森林演算法

y 為要預測的項目(response), x 為會影響預測項目的因子(factor), ntree 和 mtry 要自己調整, importance=TRUE 會算出各個因子對預測項目的重要程度

```
install.packages("randomForest")
library(randomForest)

ind = sample(2, nrow(product), replace = TRUE, prob=c(0.8, 0.2))
product.tr = product[ind == 1,]          #training set
product.te = product[ind == 2,]          #testing  set

# factor: product.tr[,c(7:65)], response: product.tr[,'rebuy']
rf.tr = randomForest(product.tr[,c(7:65)], product.tr[,'rebuy'], ntree
=500, mtry=8,importance=TRUE)
table.rf.tr = rf.tr$confusion[,1:2]
sum(diag(table.rf.tr))/sum(table.rf.tr)     #計算預測正確率
rf.tr$confusion
varImpPlot(rf.tr,cex=0.7)                   #越有影響的因子越靠近圖的右上角
print(rf.tr)
rf.tr$importance                            #print 各因子重要程度

#testing set 套入 training 出來的 model 看預測結果
pre.test = predict(rf.tr, product.te)
table.test = table(product.te$rebuy,pre.test)
sum(diag(table.test))/sum(table.test)        #testing set 預測正確率
```

➢ **ts( ) : time series object**

```
# Time-series
data.ts <- ts(c(2,5,4,6,3,7,9,8),start=c(2009,2),frequency=4) #4 季
data.ts
start(data.ts)
end(data.ts)
plot(data.ts, type="b")
# ts(c(1:30),start=c(2009,2),frequency=12)   #12 個月
```

output:

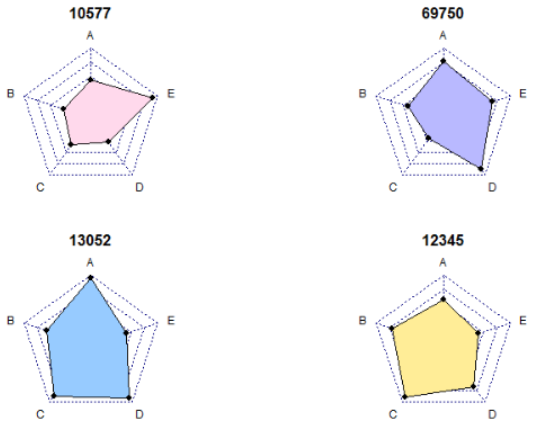
```
> data.ts
     Qtr1 Qtr2 Qtr3 Qtr4
2009         2    5    4
2010    6    3    7    9
2011    8
```

➢ **radarchart( ) : 畫雷達圖　(檔案: radar_data.csv)**

```
install.packages("fmsb")          # R 畫 radar chart 的套件
library("fmsb")

par(mar=c(1, 2, 2, 1))            # decrease default margin
color=c('#FFD9EC','#97CBFF','#B9B9FF','#FFED97')  #create color vector
layout(matrix(1:4, ncol=2))       #draw 4 plots to device
lapply(1:4, function(i) {
  radarchart(rbind(rep(100,4), rep(0,4), radar_data[i,2:6]),title=rada
r_data[i,1],vlcex=0.9,pfcol=color[i])
})
# radar_data[i,2:6]  → 第 i 編號 A~E 面向的分數
# rep(100,4) → 雷達圖各面向的滿分是 100，資料總共有四筆，所以有 4 個 100
# rep(0,4)  →雷達圖各面向最低 0 分，資料總共有四筆，所以有 4 個 0
# title = radar_data[i,1] → 圖的標題為 radar_data 第一直航的第 i 個 element
# pfcol=color[i] → 指定第 i 個雷達圖的顏色
```

proid	A	B	C	D	E
10577	43	27	34	27	91
13052	94	58	87	91	41
69750	76	42	20	88	67
12345	56	71	90	67	41



➢ **最後畫一下台灣地圖療癒一下^^**

```
install.packages("ggmap")
library(ggmap)
map.taiwan <- get_map(location="Taiwan", zoom=8)
ggmap(map.taiwan)
```