

# R軟體入門

---

2015.01.17

沈彥廷

ytshen@tmu.edu.tw



# 大綱

---

- R軟體簡介
- R軟體及相關工具安裝
- 環境介紹、套件安裝
- 基本運算規則
- 變數型態
- 資料的輸入與輸出
- 程式流程控制
- 自訂函數
- 程式撰寫技巧

# What? Why? When? Who?

- R軟體是基於S語言所建構而成的一套自由軟體，最初由Ross Ihaka及Robert Gentleman所開發。
- 被廣泛用於資料處理、統計分析、圖形繪製等功能的平台。
- 目前最新版本：R 3.1.2；套件數量62xx(指數增加中)。
- R軟體可安裝於Windows、MacOS、Linux作業系統。

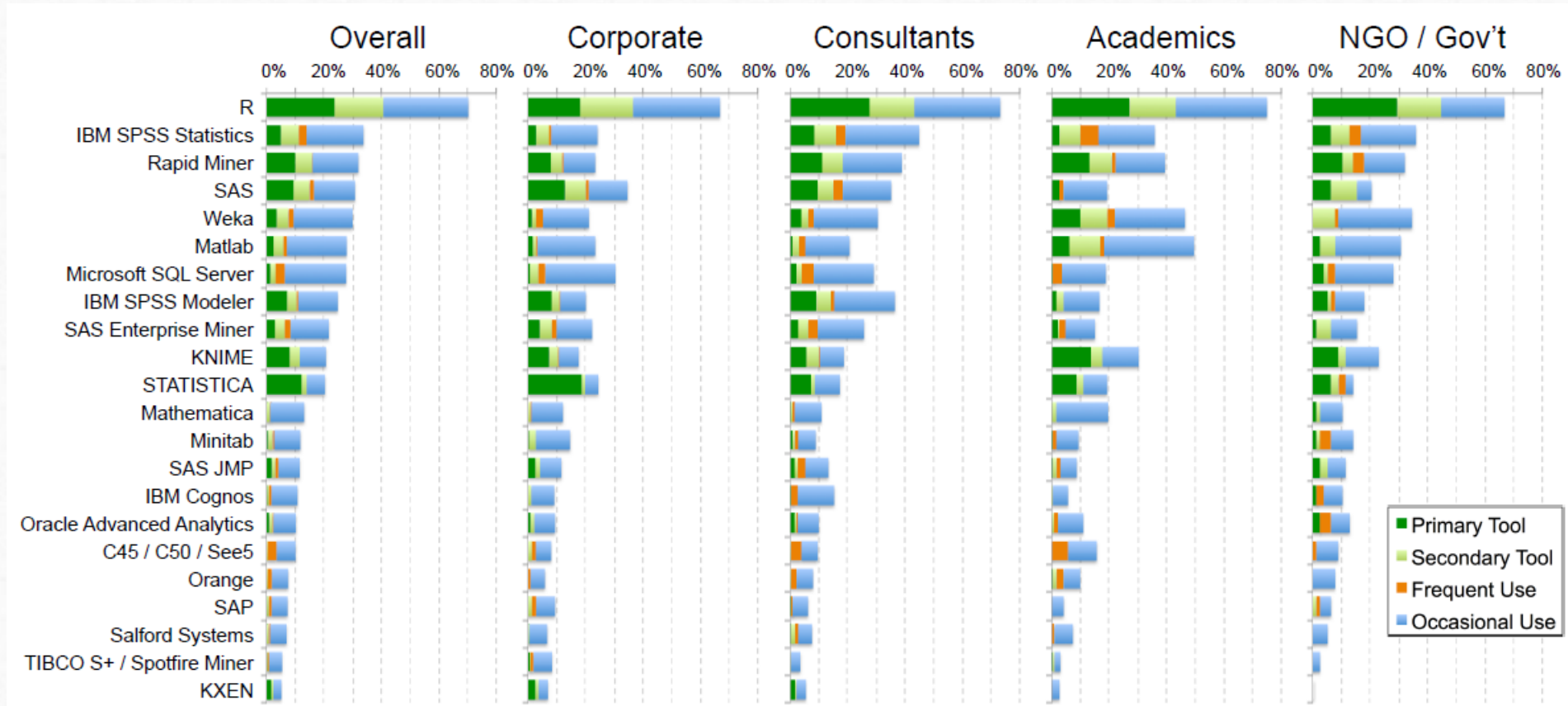
R軟體網站 – <http://www.r-project.org/>



# What? Why? When? Who?

- 程式功能完整、擴充性強
- 提供各主流作業系統的執行環境
- 豐富的討論及教學文件
- 強大的繪圖功能
- 快速銜接最新技術
- 支援其他程式語言如：C、Fortran、Python、Java
- 自由
- **免費！**

# What? Why? When? Who? (資料來源：r4stats.com)







# What? Why? When? Who?

貝氏推論	化學計量、計算物理	臨床試驗分析
集群分析	微分方程	機率分配
計量經濟	生態環境分析	實驗設計
財務金融	遺傳學	圖形繪製
平行運算	資料探勘	醫學圖像分析
綜合研究	多變量統計	自然語言處理
計算數學	調查方法	最佳化
藥物動力學	系統學	心理學
重複性研究	穩建統計方法	社會科學
空間統計	時空資料分析	存活分析
時間數列分析	網路技術及服務	圖形化

What? Why? When? Who?

```
if (  =>  ) {  
    probably not.  
} else {  
    go ahead !  
}
```

**print(“R只是一個工具！”)**



# R語言的十個等級 (by 淡江大學 統計系 陳景祥教授)

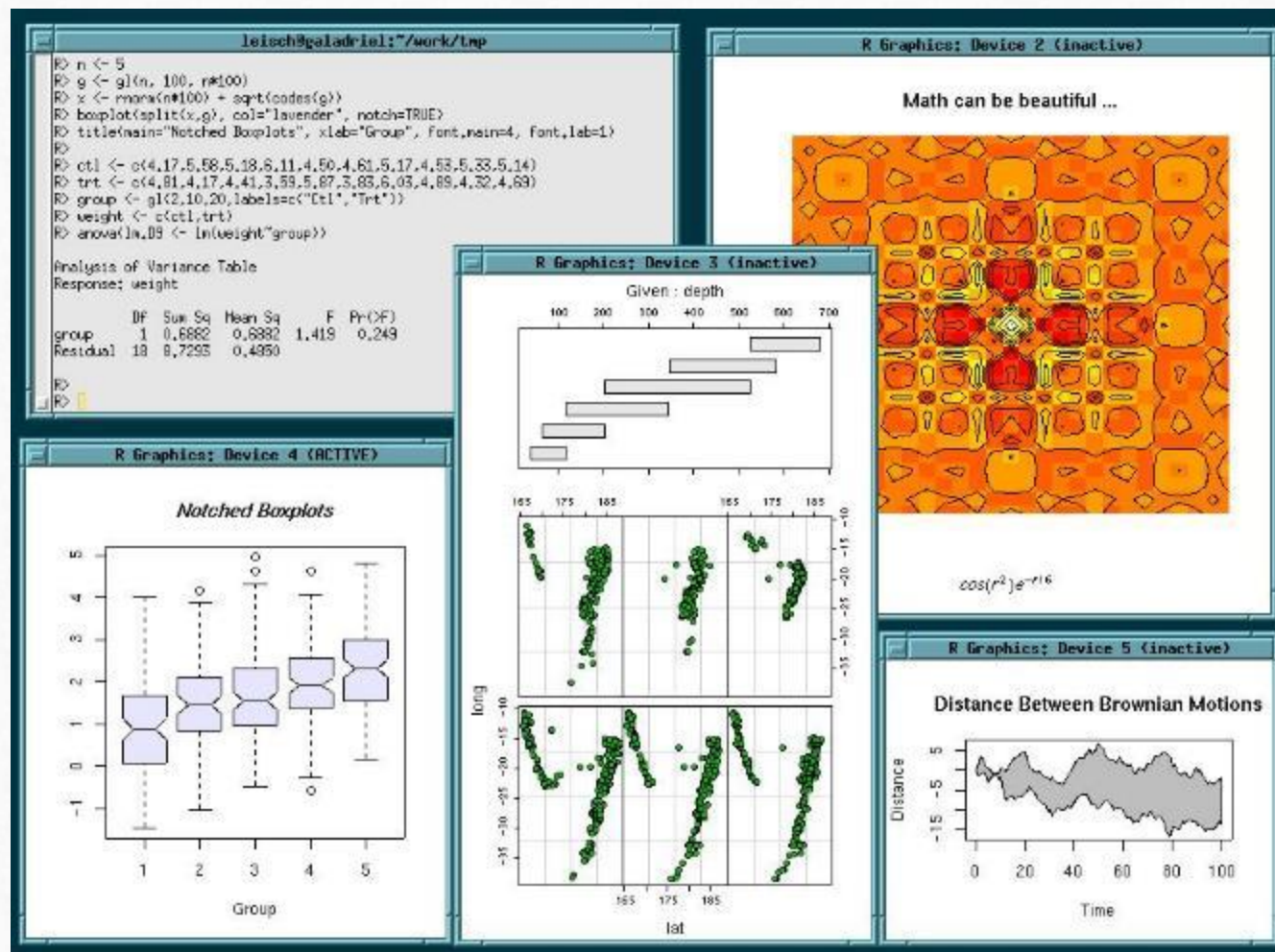
- (1) 寫程式引用適當函數來分析資料
- (2) 細緻化處理或美化 Output 與圖形
- (3) 動態變數替換(Variable-Replacement)
- (4) 在 R 程式中使用其他程式語言library
- (5) Package 包裝
- (6) 簡單 package 寫作(R programs)
- (7) Class 與 Methods
- (8) 進階 package 寫作(C, Fortran, Java)
- (9) 高階 package 寫作(GUI, HTML, LaTeX 處理)
- (10) 大型資料處理、多機平行運算



# 程式語言的五個基本功能

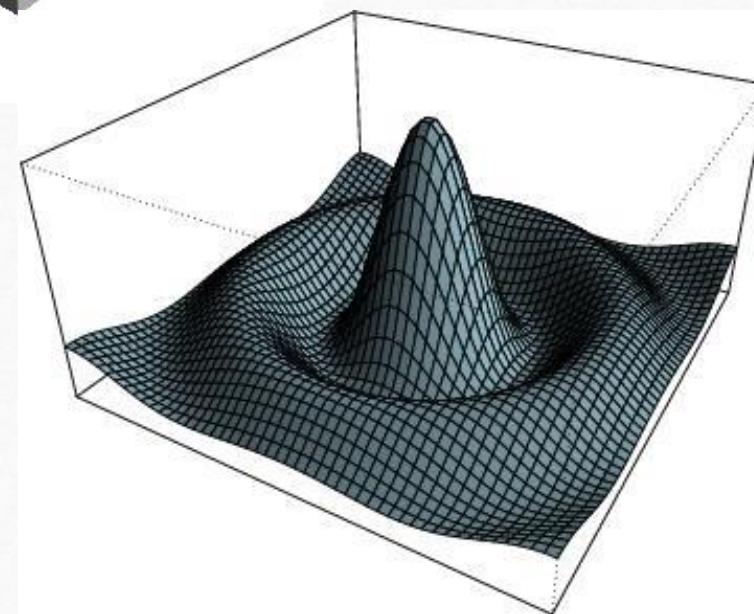
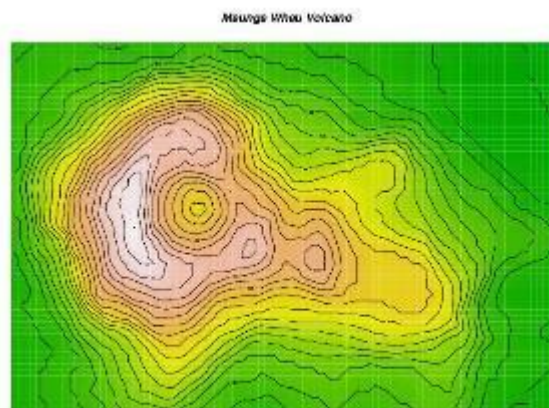
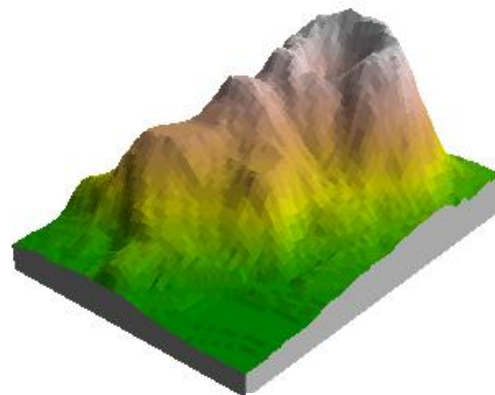
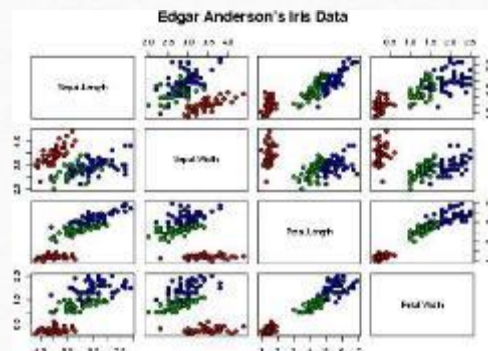
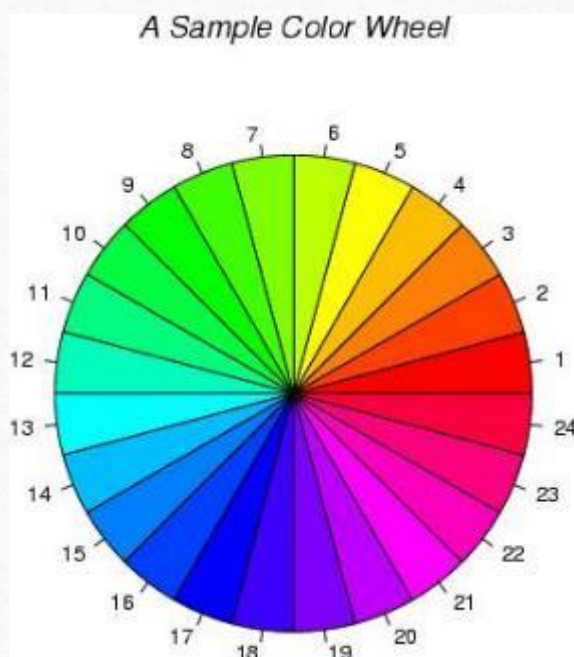
(1)	變數與常數	可以表達常數(constant)，一般變數(variable)，或陣列變數(array)
(2)	輸入與輸出	外部資料的輸入與輸出，有時候也包含遠端網站資料的存取
(3)	條件執行或邏輯判斷	Conditional execution、logical decision。例如if-else、switch等語法
(4)	迴圈	例如for、while、until等語法
(5)	獨立模組	函數或副程式(function、subroutine、procedure、或module)，允許使用者發展自己的獨立應用模組，以供多個不同程式或使用者呼叫使用

# 範例





# 範例



# 安裝R軟體

- (1) R軟體網站 - <http://www.r-project.org/>
- (2) 點選左側選單中的『Download, Packages → CRAN』
- (3) 選擇任一鏡像站(建議選擇位於臺灣的鏡像站)
- (4) 根據作業系統選擇合適的連結下載最新版的R軟體
- (5) 進行安裝 → always 『NEXT』
- (6) 開始使用R軟體



# 安裝相關工具

R的整合開發環境：RStudio



<http://www.rstudio.com/>

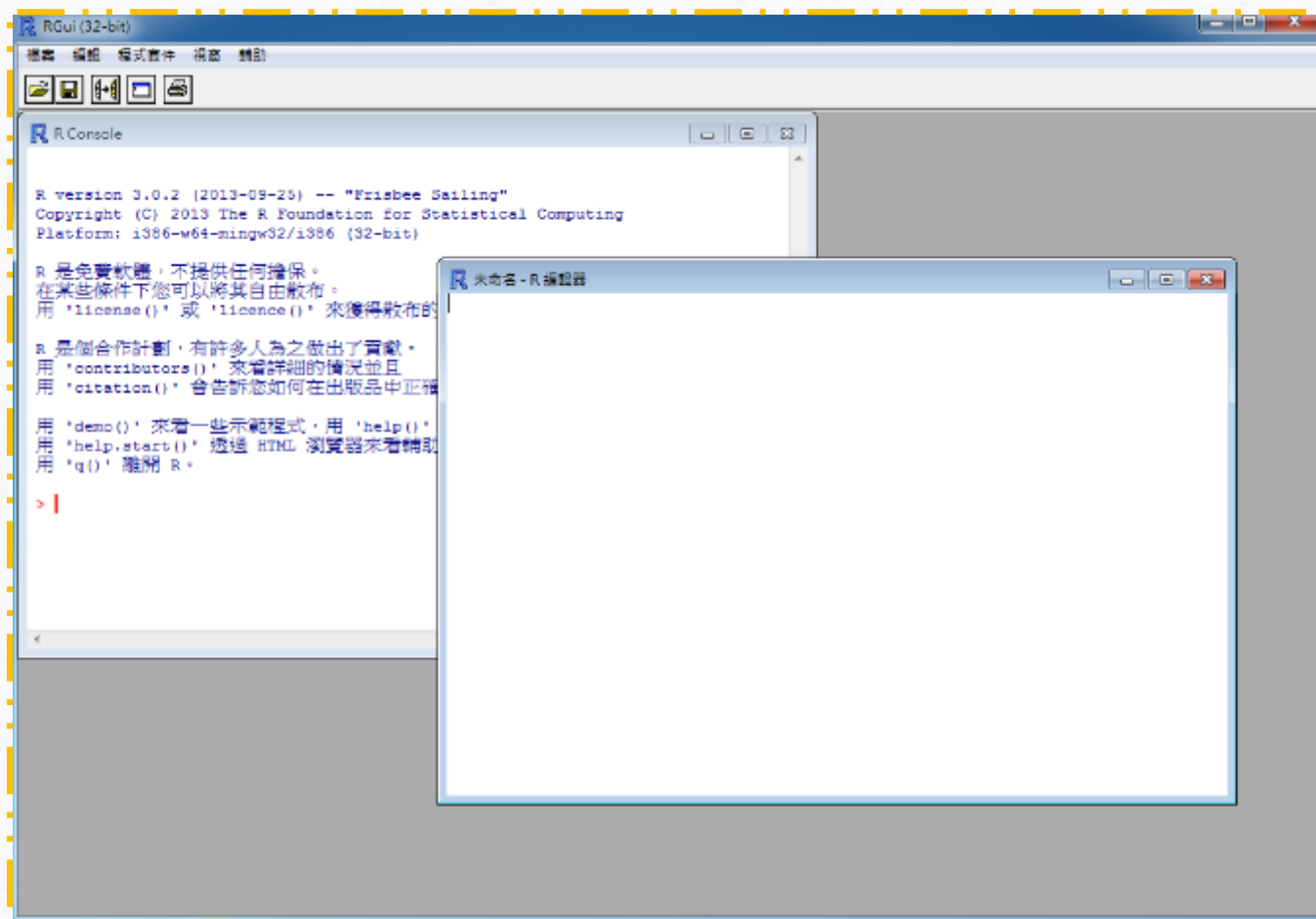
純文字代碼編輯器：Notepad++



<http://notepad-plus-plus.org/>

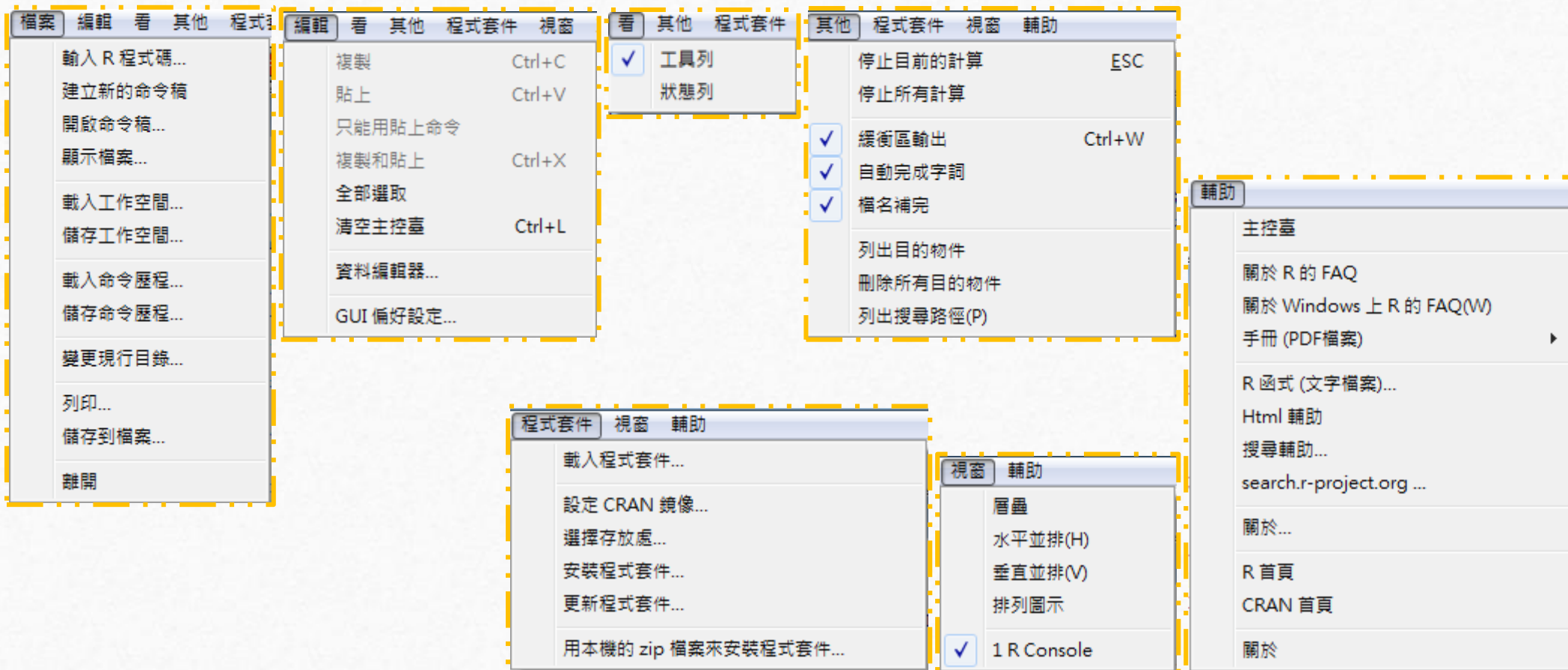
# RGui環境介紹

R的Windows圖形介面模式





# RGui環境介紹



# 基本操作環境 – 相關函數

help或?           # 檢視某函數的說明文件  
help.search或??   # 搜尋包含關鍵字的指令  
source             # 從指定的路徑或網址載入外部程式碼  
sink                # 紀錄運算的執行結果  
demo, example     # 指定套件或函數的範例展示  
install.packages   # 下載並安裝套件  
library             # 載入已安裝之套件  
getwd, setwd       # 查詢或設定工作目錄位置



# 牛刀小試

- 將工作目錄設定為”D:\R\_work\”
- 嘗試下載並安裝”rgl”套件
- 利用demo功能檢視rgl套件中的函數範例
- 利用help功能查看rgl套件中的函數說明
- 嘗試執行rgl套件中的函數範例



# 基本運算規則

## 簡單的數字與字串運算

1+2

3-4

5\*6

7/8

9^0 # 等同於9\*\*0

1+2\*(3/4)^5 # 先乘除後加減，有括號先運算

sqrt(2)

abs(-1)



# 基本運算規則

```
## 基本向量運算
x = c(1, 2, 3, 4, 5) # 等同於1:5
y = x+1
x*y
length(x) # 向量長度
sum(y) # 加總
prod(x) # 累乘
mean(y) # 平均
z = c(x, y)
```

# 基本運算規則

## 向量及矩陣的指標用法

`x[1]`

`y[c(2, 4)]`

`z[4:6]` # 等同於`z[c(4, 5, 6)]`

`x >= 3`

`y[x >= 3]`

`x[x <= 2 | y==5]` # 且(&, &&) , 或(|, ||)

`length(x[x < 4])`

`sum(y[y != 6])` # 不等號!=

`x[-1]`

`y[-(2:4)]`

`X = rbind(x, y)` # row bind

`Y = cbind(x, y)` # column bind

`X[1, 5]`

`X[, 2]`

`Y[2, ]`

`X[, c(1, 3)]`

`Y[2:4, -1]`

`Y[-1, -2]`



# 基本運算規則

```
## 多種指標用法
iris ## iris dataset
iris[, 5]
iris[, "Species"]
iris$Species
iris[["Species"]]
iris["Species"] # 等同於iris[5]
names(iris)
names(iris)=c("A","B","C","D","E")
iris$A
iris[18, c("B","D")]
iris[iris$E == "setosa", 1:4]
```

# 基本運算規則 – 相關函數

seq                   # 生成有序數列

rep                   # 重複指定元素

length               # 計算元素個數

sum, cumsum       # 元素加總及累積加總

prod, cumprod     # 元素乘積及累積乘積

sort, rank, order   # 排序、排序後各元素的順位、排序後各元素在原向量中的指標



# 牛刀小試

- 取出iris資料集中鳶尾花品種為setosa的資料
- 將上述資料依Sepal.Length變數進行排序



# R的變數與資料

```
## R軟體資料屬性: 邏輯真假值(logical,T,F), 整數(integer),  
## 雙倍精確度數字(double, real, numeric),  
## 複數(complex), 文字字串(character, string), 二進位資料(raw)
```

```
1; 20.0; 3e2 # 數值
```

```
class(1)
```

```
"stat" # 文字
```

```
class("stat")
```

```
TRUE; T; FALSE; F # 邏輯真假值
```

```
class(T)
```



# R的變數與資料

```
## R軟體變數種類: 向量(vector), 矩陣(matrix), 陣列(array),  
## 因子(factor), 資料框架(data-frame), 串列(list), 時間數列(ts)
```

```
## 向量(vector) ##
```

```
x = c(1, 3, 5, 7, 9) # 建立向量, 或聯結不同的向量
```

```
is.vector(x) # 查詢x是否為向量變數
```

```
y = c(2, "stat", T) # 元素屬性需相同
```

```
x[1]; y[-2] # 向量指標
```

```
x[c(1, 3, 5)] = c(2, 4, 6)
```

```
c(x, y[1])
```

```
length(x) # 算出元素個數
```

```
names(x) # 查詢或建立向量的元素名稱
```

```
names(x) = c("a", "b", "c", "d", "e")
```

```
x[c(3, 5)]; x[c("c", "e")]
```

# R的變數與資料

```
## 陣列(array), 矩陣(matrix) ##  
X = array(1:6, c(3, 2)) # matrix(1:6, 3, 2) # 建立陣列變數  
Y = array(1:12, c(2, 3, 2))  
is.array(X); is.matrix(X) # 查詢X是否為陣列及矩陣變數  
rbind(x, x); cbind(x, x) # 使用rbind及cbind來建立array變數  
nrow(X) # 查詢陣列的列數  
ncol(X) # 查詢陣列的行數  
dim(Y) # 查詢Y陣列的維度  
rownames(X) = c("R1", "R2", "R3")  
colnames(X) = c("C1", "C2")
```



# R的變數與資料

```
## 矩陣(Matrix) ##  
X = matrix(1:6, 3, 2)  
Y = t(X) # 轉置  
Z = X %*% Y # 矩陣相乘  
diag(Z) # 對角線函數  
det(Z) # 行列式  
A = matrix(1:4, 2, 2)  
b = c(2,2)  
solve(A) # 反矩陣  
solve(A, b) # 線性聯立方程式  
eigen(Z) # 特徵值與特徵向量
```

# R的變數與資料

```
## 因子(Factor) ##  
x = c(1, 1, 1, 2, 2, 2)  
y = factor(x) # 等同於 as.factor(x)  
y - 1  
levels(y) # 查詢或設定分類資料  
levels(y) = c("一", "二")  
nlevels(y) # 查詢分類數目
```



# R的變數與資料

```
## 串列(List) ##  
l=list(L1 = x, L2 = y, L3 = Z)  
names(l)  
l$L1 # 等同於 l[[1]] 或 l[["L1"]]  
l[2] # 等同於 l["L2"]  
l$L3[1, 2]  
l$L4 = 1:5  
c(l, list(L5 = 1:10))
```

# R的變數與資料

```
## 資料框架(Data-Frame) ##  
D = as.data.frame(Z) # 將變數類型轉為data-frame  
D[, 4] = c(T, F, T)  
names(D) = c("D1", "D2", "D3", "D4")
```



# R的變數與資料 – 相關函數

is, as 系列函數 # 判斷或轉換特定變數型態

attributes, attr # 查詢物件的屬性

class # 查詢物件的類別

str # 查詢物件的結構

which # 傳回符合條件的位置指標

match # 傳回第二個參數在第一個參數中出現的的位置指標

# 牛刀小試

已知 $x = 1:5$ 以及 $y = c(\text{“一”}, \text{“二”}, \text{“三”}, \text{“四”}, \text{“五”})$

- 將 $x$ 的元素名稱改為甲、乙、丙、丁、戊
- 查詢 $\text{rbind}(x, y)$ 的維度
- 將 $y$ 變數的型態轉變為因子(factor)變數
- 將 $\text{cbind}(x, y)$ 的型態轉變為資料框架(data-frame)
- 建立一個 $2 \times 3 \times 4$ 的三維陣列，其元素為 $1:24$
- 查詢 $\text{rbind}(x, y)$ 的型態是否為陣列(array)
- 將 $x, y$ 合併為一個串列(list)，並查詢其第一個元素





# 資料的輸入與輸出

```
setwd("D:/R_work/") # 設定工作目錄

## 文字檔輸入 ##
babies = read.table("babies.txt", header=T)
babies = na.exclude(babies) # 刪除具有遺失值的資料
Iris = read.table("iris_dataset.txt", header=F, sep=",")
IRIS = read.csv("iris.csv", header=F)
```

## 資料的輸入與輸出

```
## 文字檔輸出 ##  
cat(babies$smoke, file="smoke1.txt", sep="")  
write(babies$smoke, file="smoke2.txt", sep=",")  
weight = babies[babies$weight < 100,]  
height = babies[babies$height > 70,]  
write.table(weight, file="weight.txt", sep="," , row.names=F)  
write.csv(height, file="height.csv", row.names=F)
```



# 資料的輸入與輸出

```
## 存取其他軟體的資料檔 ##  
library(gdata)  
babies_xls = read.xls("babies.xls", sheet=1) # 讀取xls檔  
library(xlsx)  
babies_xlsx = read.xlsx("babies.xlsx", sheetIndex=2) # 讀取xlsx檔  
write.xlsx(iris, "iris.xlsx", sheetName="iris") # 匯出xlsx檔  
library(sas7bdat)  
babies_sas = read.sas7bdat("babies.sas7bdat") # 讀取sas資料檔  
library(foreign)  
babies_spss = read.spss("babies.sav", to.data.frame=T) # 讀取spss資料檔
```

## 資料的輸入與輸出

```
## 存取R物件 ##  
save(weight, height, file="babies.RData")  
save.image() # 儲存工作空間  
load("babies.RData")
```



# 資料的輸入與輸出 – 相關函數

`read.csv2` # 輸入以分號分隔的csv檔

`read.delim` # 輸入以tab分隔的文字檔

`read.fwf` # 輸入固定寬度格式文字檔

`write.csv2` # 輸出以分號分隔的csv檔

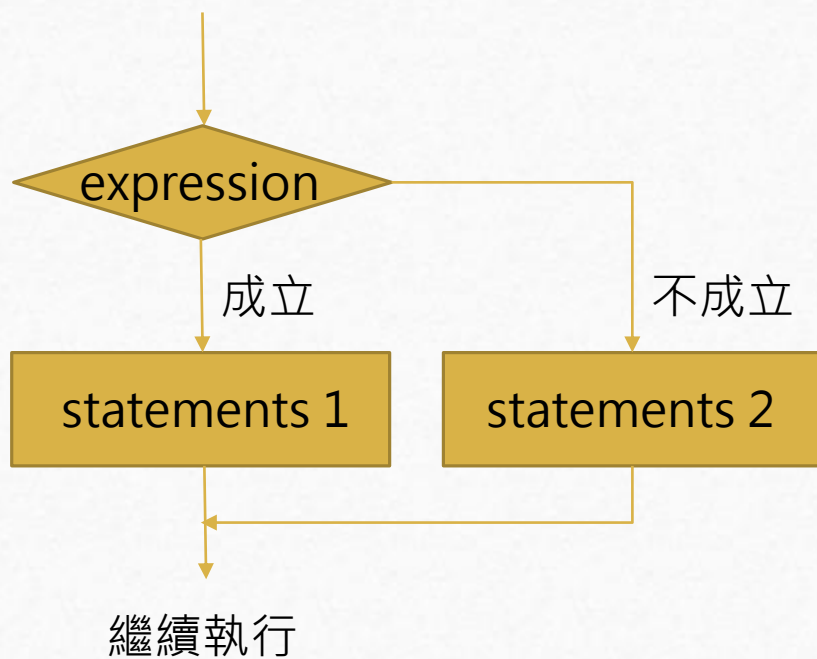
# 牛刀小試

- 讀入外部資料檔”babies.txt”
- 分別將babies及iris匯出成csv檔且不包含列名稱
- 將babies及iris儲存至同一個Rdata檔

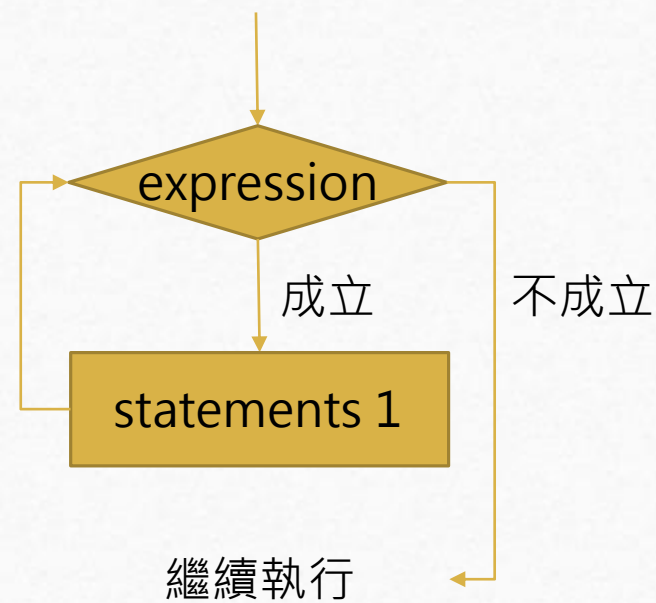




# 程式流程控制



條件執行



迴圈結構

# 程式流程控制

## 邏輯判斷式 ##

## 運算子優先性：

## 括弧 => 乘除 => 加減 => 比較 => 邏輯 => 指派

x = 1

x == 3

x != 1 + 2

!(x <= 3)

x %in% 1:5

x < 0 || x > 5

(is.matrix(x) || x >= 0) & (1 < 2)



# 程式流程控制

```
## 條件執行 ##  
x = 1  
if (x == 3) y = 10 else y = 20  
if (x >= 5) {  
    y = 15  
} else {  
    y=0  
} # 建議寫法  
if (x < 0) {  
    y = x - 1  
} else if (x > 0) {  
    y = x + 1  
} else {  
    y = x  
}
```

# 程式流程控制

```
## for迴圈 ##  
y = vector() # 宣告變數  
for (x in 1:5) {  
    y[x] = sqrt(x)  
}  
z = 1  
for (i in c(2,4,6,8,10)) {  
    z = z * i  
} # 2*4*6*8*10
```



# 程式流程控制

```
## while迴圈 ##  
x = 1; y = vector()  
while (x <= 5) {  
    y[x] = sqrt(x)  
    x = x + 1  
}  
z = 1; i = 2  
while (i <= 10) {  
    z = z * i  
    i = i + 2  
}
```

# 程式流程控制

```
## repeat迴圈 ##  
x = 1; y = vector()  
repeat {  
    y[x] = sqrt(x)  
    x = x + 1  
    if (x > 5) break # 跳離迴圈  
}  
z = 1; i = 1  
repeat {  
    i = i + 1  
    if (i > 10) {break} else if (i %% 2 != 0) {next}  
    z = z * i  
} # next: 跳過一次迴圈
```



# 程式流程控制 – 相關函數

**ifelse** # 二分類邏輯判斷

**switch** # 依照第一個參數決定執行的運算式

# 牛刀小試

有一物件  $x = \text{matrix}(\text{NA}, 5, 5)$

- 以迴圈逐一判斷每個元素的行列指標和是否可被3整除
- 若可以整除則該元素值為1；反之為0
- 例：因 $4+2=6$ 可被3整除，故第4列第2行的元素值為1





# 自訂函數

```
# 函數的定義語法：  
# 自訂function名稱 = function(參數1, 參數2, ...)  
# {  
#   完整運算式...  
# }  
  
func1 = function(a, b)  
{  
  x = 1+2*3/4  
  y <- a + b  
  return(y) # 預設傳回最後一個運算值，或使用return函數  
}  
func1(7, 6)  
func1(b=3, a=2)
```

# 自訂函數

```
## 參數的預設值 ##  
func2 = function(x=0)  
{  
    sum(x)/length(x)  
}  
func2(1:5)  
func2() # 參數x的預設值為0
```



# 自訂函數

```
## ... 參數 ##  
func3 = function(x, ...)  
{  
  y = mean(x, ...) + 1  
  return(y)  
}  
x = c(2,4,6,NA,10)  
func3(x, trim=0.1)  
func3(x, trim=0, na.rm=TRUE)
```

# 自訂函數

```
## 二元運算子 ##  
"%p%" = function(a, b)  
{  
  factorial(a)/factorial(a-b)  
}  
5 %p% 2
```



# 自訂函數 – 相關函數

`invisible` # 回傳值不顯示計算結果

`stop` # 停止函數執行並顯示錯誤訊息

`warning` # 顯示警告訊息

# 牛刀小試

- 定義一個可計算階層數的函數
- Hint：利用遞迴呼叫， $0! = 1$
- 定義一個可進行組合數計算的二元運算子
- 註： $n$ 取 $r$ 的組和數  $= n! / r! / (n-r)!$





# 程式撰寫技巧

- 善用apply系列函數
- 盡量利用向量、指標特性
- 減少迴圈的使用
  
- 程式碼要排版
- 程式碼要加註解
- 變數名稱不要隨便取
  
- 善用help功能
- trial and error

# 程式撰寫技巧

`apply(X, MARGIN, FUN, ...)`

說明：將FUN作用到X的每個MARGIN目標上

輸入：陣列(array)、資料框架(data.frame)

輸出：若FUN傳回值為向量，則輸出向量或矩陣，否則輸出串列(list)

```
apply(iris[-5], 2, max)
func = function(x) x[x < mean(x)]
apply(iris[,1:4], 2, func)
```



# 程式撰寫技巧

`tapply(X, INDEX, FUN, ...)`

說明：以INDEX作為分類指標，將FUN作用到X的每個目標上

輸入：R物件，通常是一個向量(vector)

輸出：依據INDEX的維度及FUN傳回值，有可能為陣列(array)或串列(list)

```
tapply(iris[,1], iris[,5], min)
index2 = rep(1:2, length=150)
tapply(iris[,2], list(iris[,5],index2), median)
```

# 程式撰寫技巧

`sapply(X, FUN, ...); lapply(X, FUN, ...)`

說明：將FUN作用到X的每個目標上

輸入：向量(vector)、串列(list)

輸出：sapply預設優先傳回向量(vector)或陣列(array)；lapply傳回串列(list)

```
sapply(iris, length)
lapply(iris, length)
sapply(iris[-5], function(x) { which(x > mean(x)) })
```



# 程式撰寫技巧

## 各方法計算時間比較 ##

x = rnorm(50000) # 以標準常態分配生成隨機樣本

y1 = y2 = y3 = y4 = vector()

t0 = proc.time() # 起始時間

for (i in 1:length(x)) {

    if (x[i] <= 0) y1[i] = -1 else y1[i] = 1

}

t1 = proc.time() - t0 # y1的計算時間

y2 = ifelse(x <= 0, -1, 1)

t2 = proc.time() - t0 - t1 # y2的計算時間

y3[x <= 0] = -1; y3[x > 0] = 1

t3 = proc.time() - t0 - t1 - t2 # y3的計算時間

y4 = sapply(x, function(x) {if (x <= 0) -1 else 1})

t4 = proc.time() - t0 - t1 - t2 - t3 # y4的計算時間

## 程式撰寫技巧

### Which is better ?

```
aa=read.table("babies.txt",header=TRUE)
bb=na.exclude(aa$smoke);cc=vector()
for(i in 1:length(bb)){if(bb[i]==1) cc[i]="是" else cc[i]="否"}
```



## 程式撰寫技巧

```
## 讀入babies資料檔、宣告smoke及new_var變數 ##
babies = read.table("babies.txt", header=TRUE)
smoke = na.exclude(babies$smoke)
new_var = vector()

## 使用迴圈將smoke變數重新編碼並存入new_var變數 ##
for (i in 1:length(smoke)) {
  if (smoke[i] == 1) {
    new_var[i] = "是"
  } else {
    new_var[i] = "否"
  }
}
```

# 程式撰寫技巧 – 相關函數

# table系列相關函數

tabulate, table, margin.table, prop.table, xtabs, ftable

# 遺失值相關函數

na.fail, na.omit, na.exclude, is.na

# 字串處理相關函數

cat, sprint, paste, substr, strtrim, strsplit, sub, gsub, grep



# 牛刀小試

- 計算babies資料集中每一個變數的遺失值個數
- 以smoke為分組變數，繪製parity變數的次數表



# R的圖形化使用者介面

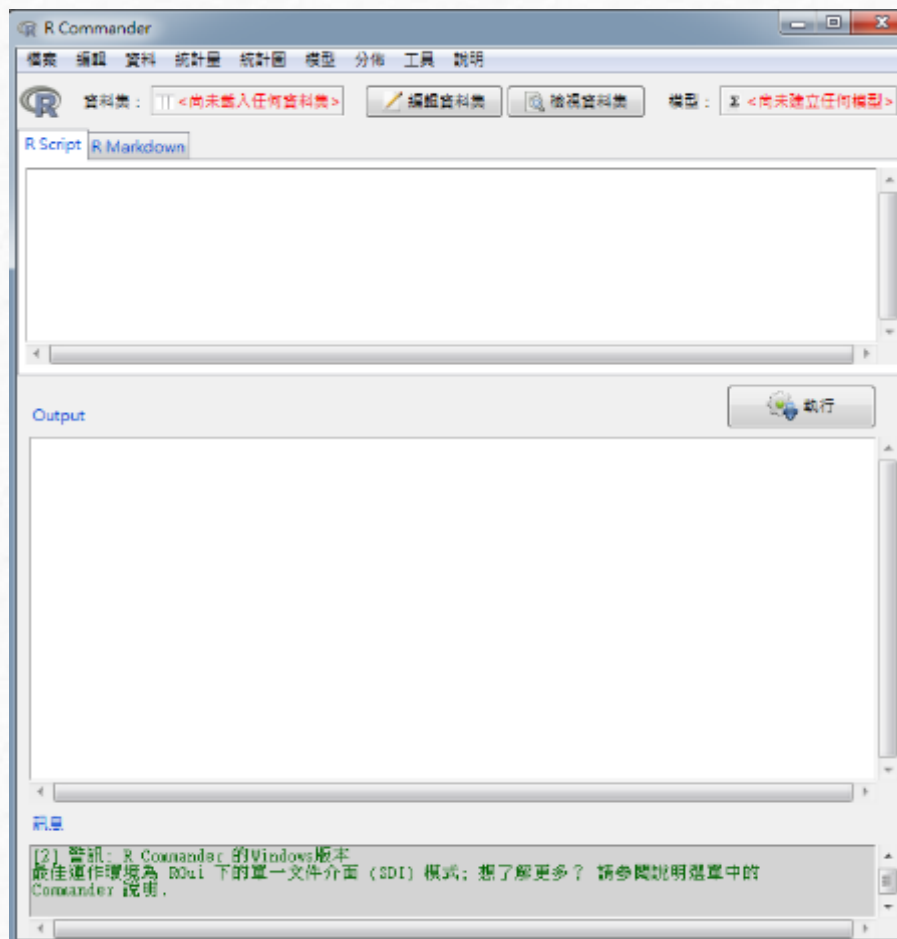
GUI	Website
R Commander	<a href="http://www.rcommander.com/">http://www.rcommander.com/</a>
SciViews-R	<a href="http://www.sciviews.org/SciViews-R/">http://www.sciviews.org/SciViews-R/</a>
JGR	<a href="http://rforge.net/JGR/">http://rforge.net/JGR/</a>
Deducer	<a href="http://www.deducer.org/">http://www.deducer.org/</a>
RStudio	<a href="http://www.rstudio.com/">http://www.rstudio.com/</a>
RKward	<a href="http://rkward.sourceforge.net/">http://rkward.sourceforge.net/</a>
Brodgar	<a href="http://www.brodgar.com/">http://www.brodgar.com/</a>
Rattle	<a href="http://rattle.togaware.com/">http://rattle.togaware.com/</a>
DAGS · C (R-web)	<a href="http://www.r-web.com.tw/">http://www.r-web.com.tw/</a>



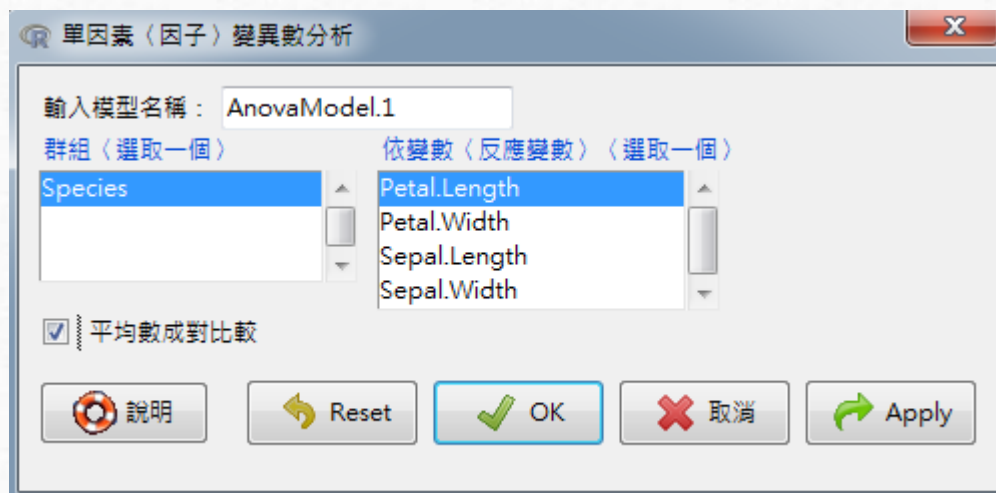
# R的圖形化使用者介面 – R commander

`install.packages("Rcmdr")`

`library(Rcmdr)`



# R的圖形化使用者介面 – R commander



```
Species      Df Sum Sq Mean Sq F value Pr(>F)
Residuals    147   27.2    0.19
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> numSummary(iris$Petal.Length, groups=iris$Species, statistics=c("mean",
+ "sd"))
      mean      sd data:n
setosa   1.462 0.1736640    50
versicolor 4.260 0.4699110    50
virginica 5.552 0.5518947    50

> .Pairs <- glht(AnovaModel.1, linfct = mcp(Species = "Tukey"))
> summary(.Pairs) # pairwise tests

Simultaneous Tests for General Linear Hypotheses

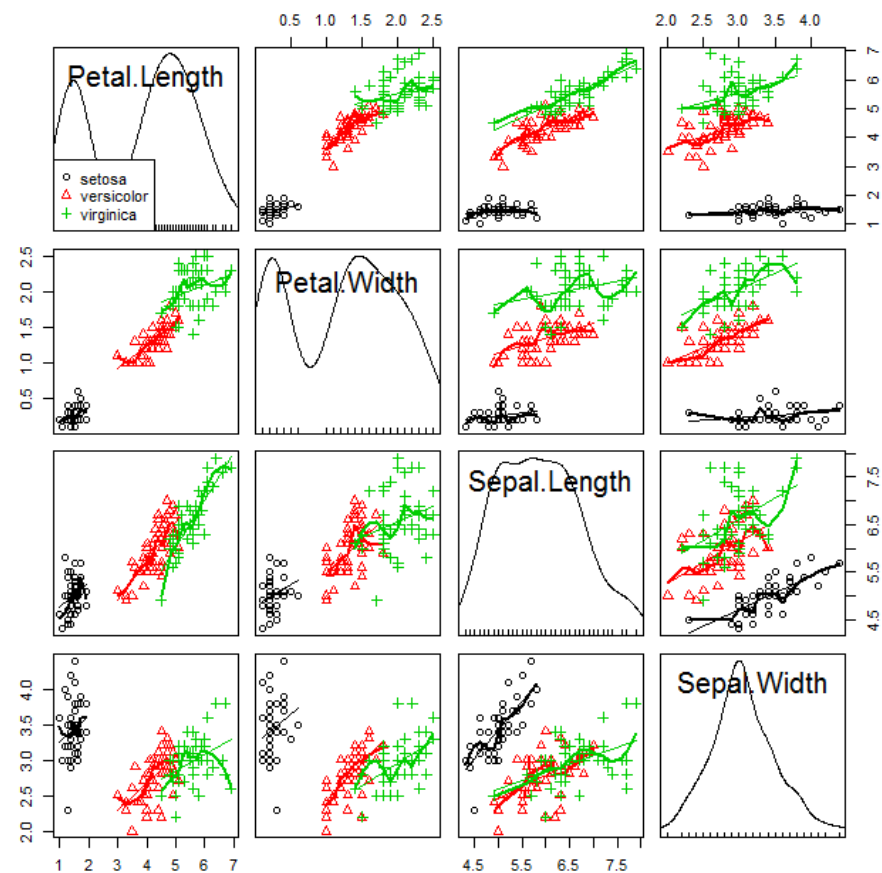
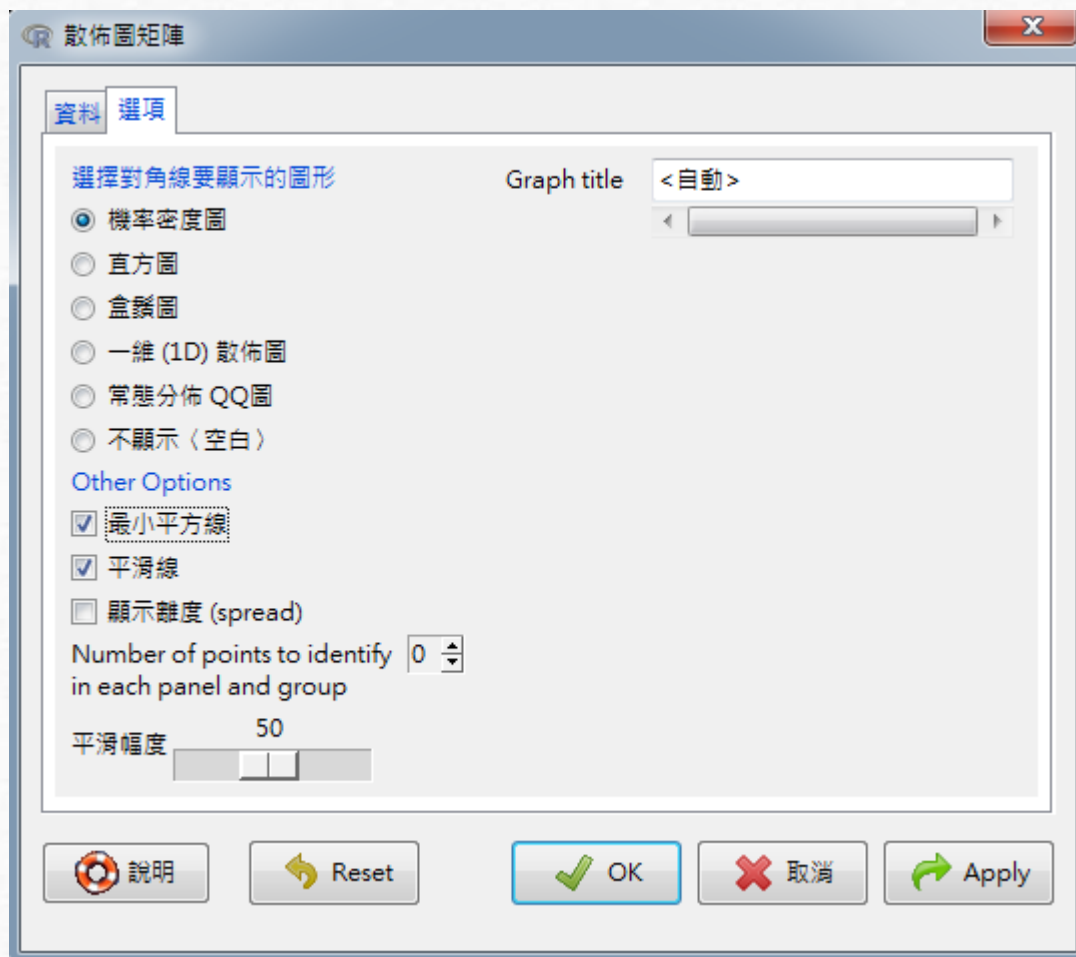
Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = Petal.Length ~ Species, data = iris)

Linear Hypotheses:
              Estimate Std. Error t value Pr(>|t|)
versicolor - setosa == 0    2.79800    0.08607   32.51 <2e-16 ***
virginica - setosa == 0     4.09000    0.08607   47.52 <2e-16 ***
virginica - versicolor == 0  1.29200    0.08607   15.01 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```




# R的圖形化使用者介面 – R commander



# R的圖形化使用者介面 – DAGS · C (R-web)

<http://www.r-web.com.tw/>



## 雲端資料分析暨導引系統

Data Analysis & Guiding System - Cloud

初階導引 資料處理 分析方法 圖表繪製 機率分配 輸出結果

首頁 | 網站導覽 | 會員登入 | 初階使用者 ▾

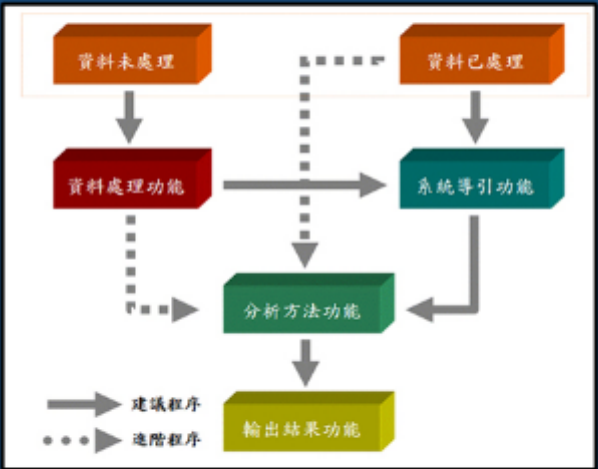
fb 讚 分享

8+1

### 使用流程簡介

此流程圖為提供新手/初階使用者  
使用本系統之建議順序，若為專家  
使用者可直接選取所需功能進行分析

## DAGS·C



```
graph TD
    A[資料未處理] --> B[資料處理功能]
    B --> C[分析方法功能]
    C --> D[輸出結果功能]
    E[資料已處理] --> F[系統導引功能]
    F --> C
    C --> D
    A -.-> F
    B -.-> C
    F -.-> D
```

→ 建議程序  
... 進階程序

雲端資料分析暨導引系統(Data Analysis & Guiding System-Cloud, DAGS-C)

在網路上以視窗點選方式操作，資料處理及分析零負擔。



# R的圖形化使用者介面 – DAGS · C (R-web)

步驟一：資料匯入

選擇要進行分析的資料檔或上傳檔案

選擇資料檔 (分析方式)

檢索資料型別(幫助說明)

資料型別 (分析方式, 分析模型)

資料型別 (關聯模型)

資料型別 (觀察變性模型)

資料型別 (主、成分分析)

資料型別 (其他)

您選擇的資料檔是： 範例F6

步驟二：資料型態設定

選擇欲進行檢定的資料型態

資料型態

檢定變數

分析模型

(說明)

步驟三：參數設定

選擇要進行分析的變數

Sepal.Width

Petal.Length

Petal.Width

檢定變數

Sepal.Length

分析模型

Species

## 變異數分析<sup>1</sup>：

虛無假設：各母體的平均數相等						
$H_0: \mu_1 = \dots = \mu_3$						
來源	平方和	自由度	均方和	F檢定統計量	臨界值	p-值 <sup>II</sup>
source	sum of squares	d.f.	mean square	F-statistics	F(d.f.1,d.f.2,1- $\alpha$ )	p-value
處理	63.2121	2	31.6061	119.2645	3.0576	< 2.22e-16 ***
誤差	38.9562	147	0.265			
總和	102.1683	149				
total						

I：分組變數為Species

II：顯著性代碼： '\*\*\*' : < 0.001, '\*\*' : < 0.01, '\*' : < 0.05, '.' : < 0.1

• 分析結果建議：由於檢定結果P-值(< 2.22e-16) < 顯著水準0.05，因此可拒絕虛無假設。

• 兩兩(pairwise)多重比較(分組變數為Species)：

◦ Scheffe兩兩比較<sup>1</sup>：

	差異 Difference	95% 信賴區間		修正P-值 Adj. p-value
		下界 Lower	上界 Upper	
setosa-versicolor	-0.93	-1.1846	-0.6754	< 2.22e-16 ***
setosa-virginica	-1.582	-1.8366	-1.3274	< 2.22e-16 ***
versicolor-virginica	-0.652	-0.9066	-0.3974	< 2.22e-16 ***

I：顯著性代碼： '\*\*\*' : < 0.001, '\*\*' : < 0.01, '\*' : < 0.05, '.' : < 0.1

67

# R的圖形化使用者介面 – DAGS · C (R-web)

步驟一：資料匯入

選擇要進行分析的資料檔或上傳檔案

選擇資料檔 (分析方法)

選擇資料型態 (圖形視覺)

選擇 2 (時間系列)

選擇 3 (廣義線性模型)

選擇 4 (廣義線性模型)

選擇 5 (結構方程模型)

選擇 6 (分類模型 - 分析模型)

您選擇的資料檔案： **資料 F-6**

步驟二：參數設定

選擇邊數

是否在圖形中加上密度估計曲線

☒ 是 ☐ 否

是否在圖形中標示密度資料位置

☒ 是 ☐ 否

進階選項設定：

圖形顏色

#FFCC99

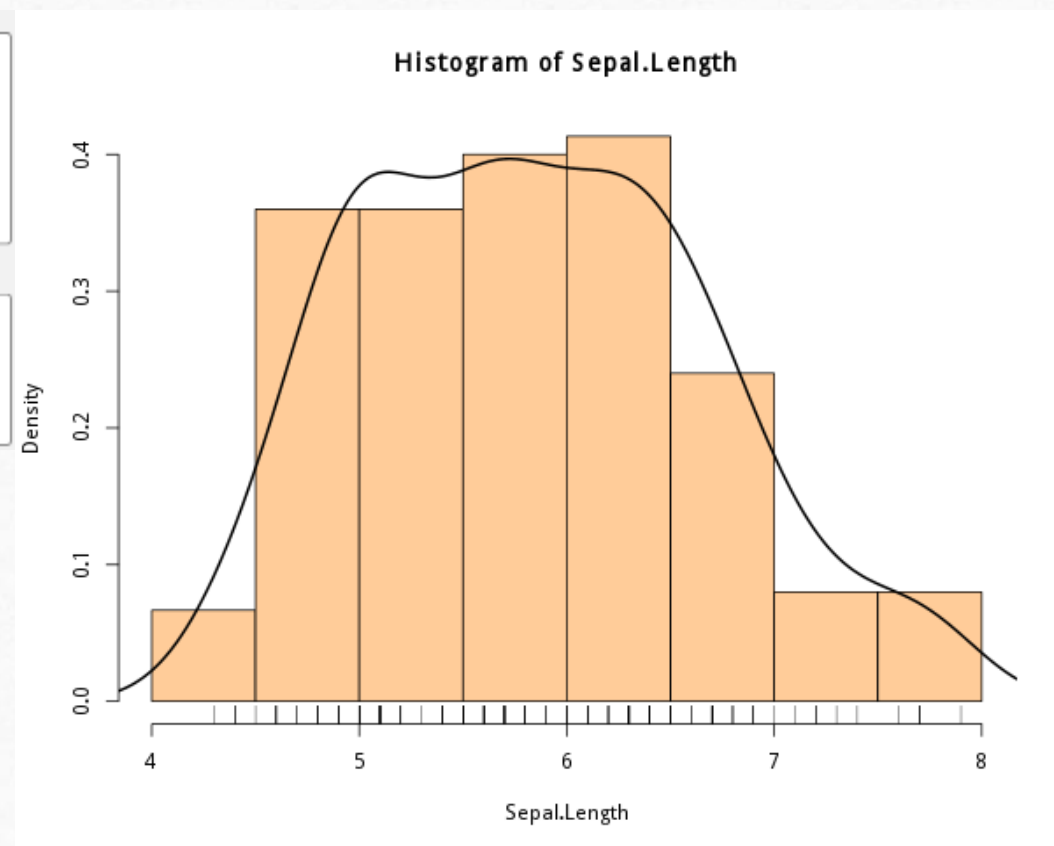


自訂分組組數

主標題

儲存設定

關閉視窗





# 參考資料

- 中文 –

- 淡江大學統計系陳景祥老師『R軟體：應用統計方法』
- 台北大學統計系林建甫老師『醫學統計與R』
- 金門大學資工系陳鍾誠老師『R統計軟體』
- 網友Jah Tsai提供『R commands』

- 英文 –

- 『The R Manuals』 - <http://cran.r-project.org/manuals.html>

# Thank you !!

---

Practice makes perfect.