

R的package安裝及基本操作

下載與安裝

<http://cran.csie.ntu.edu.tw/>



The Comprehensive R Archive Network

Frequently used pages

CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

Documentation

[Manuals](#)

[FAQs](#)

[Contributed](#)

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Linux](#)
- [MacOS X](#)
- [Windows](#)

Source Code for all Platforms

Windows and Mac users most likely want the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- **The latest release** (2009-06-26): [R-2.9.1.tar.gz](#) (read [what's new](#) in the latest version).
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

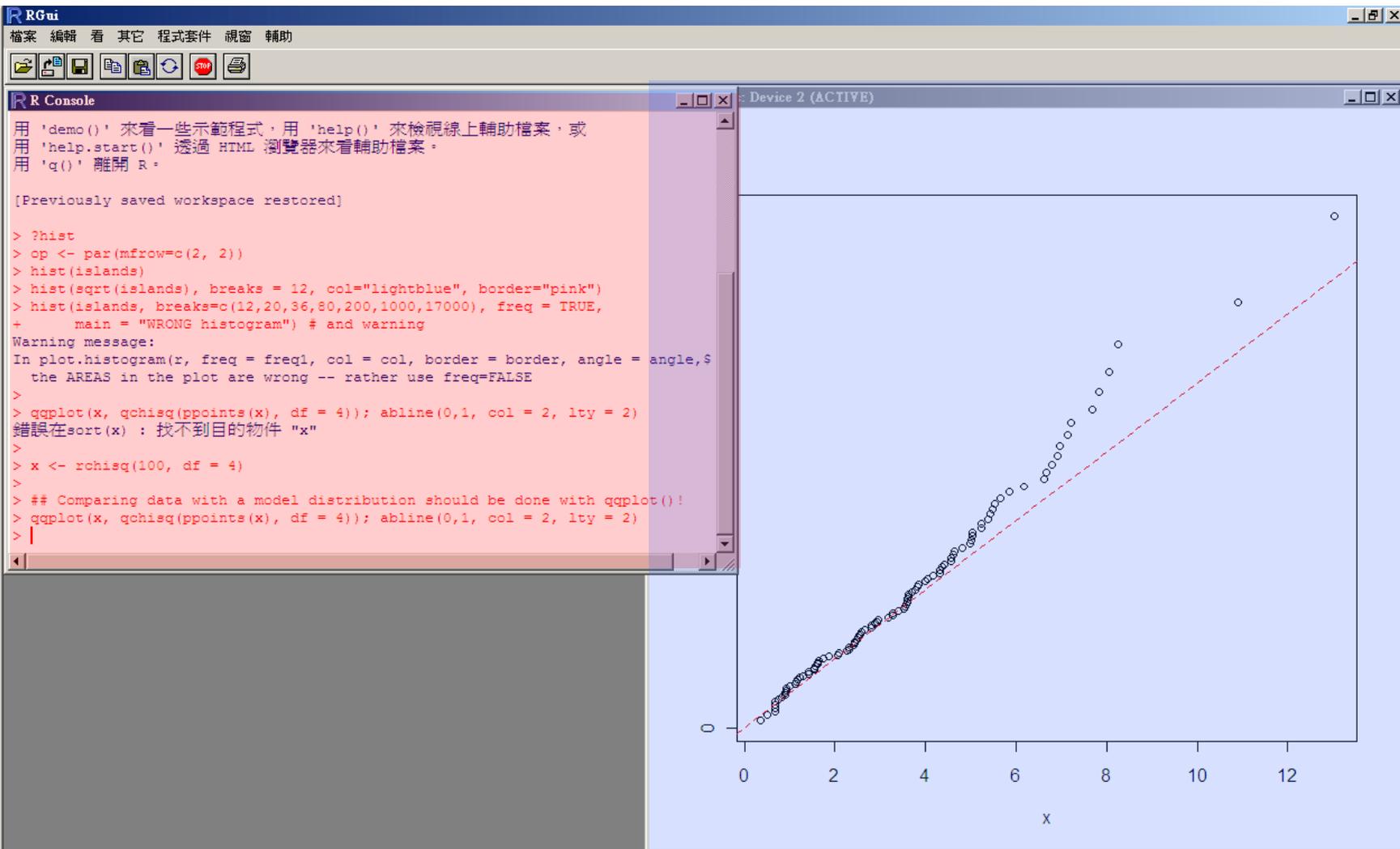
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

安裝方式與windows一般軟體安裝相同
在桌面會建立捷徑

R的操作環境

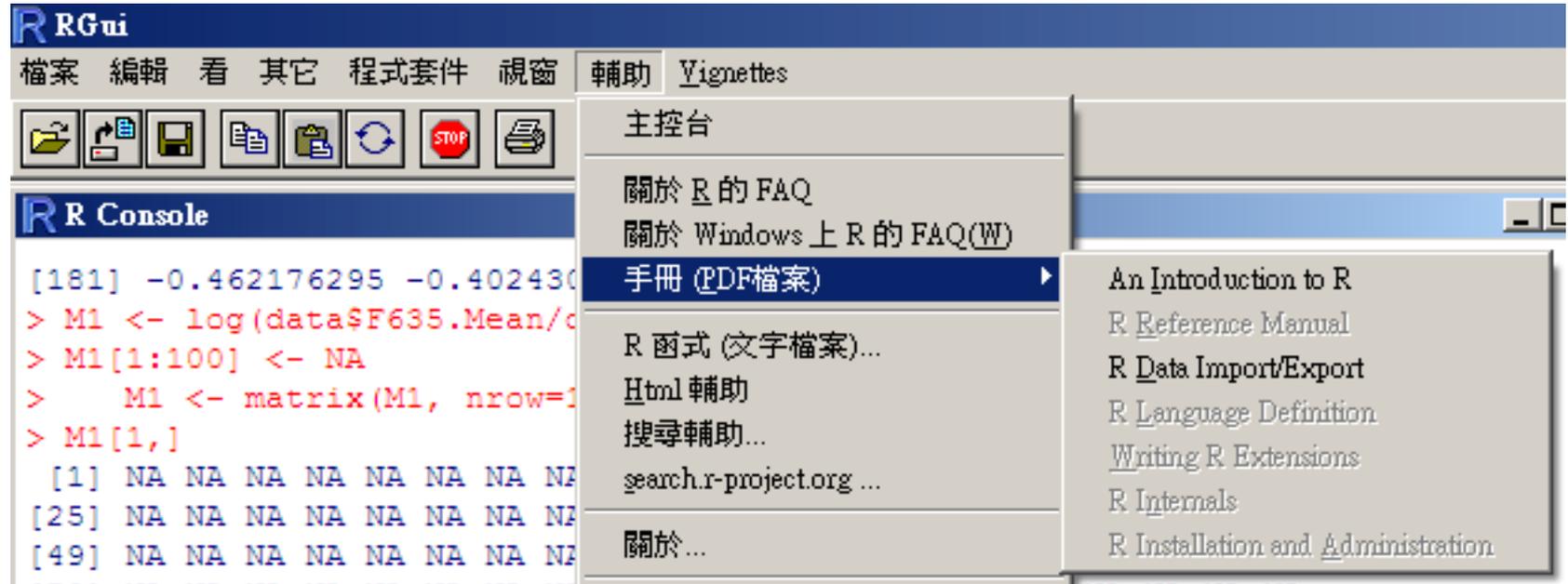
命令輸入視窗

圖形輸出視窗



R...Start

R上面的教學文件:



內容:

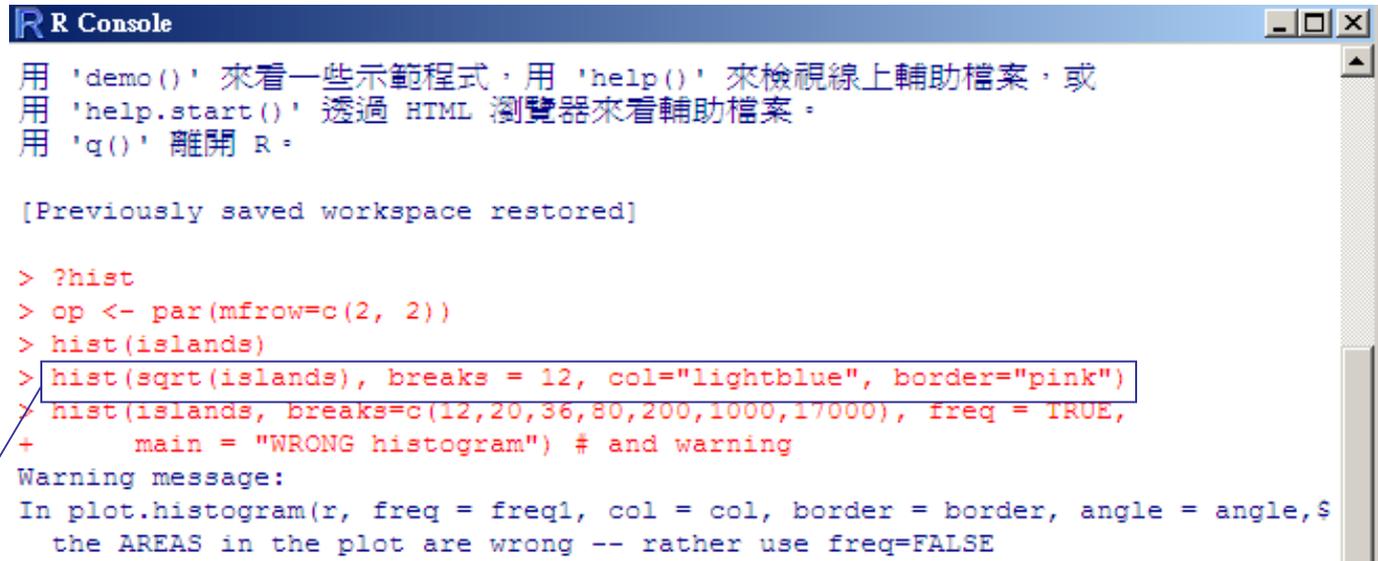
1. 介紹R上面所使用的變數格式及形態
2. 簡單的運算子使用及規則

R 的提示符號

R 的提示符號: > 與 +

“>” 為提示符號; 當提示符號出現時表示 R 正在待命中, 可以隨時鍵入下一個命令。

當提示符號為 “+” 時, 表示程式正在執行中, 或在等待未完成的指令。例如:



The screenshot shows an R Console window with the following text:

```
R Console
用 'demo()' 來看一些示範程式, 用 'help()' 來檢視線上輔助檔案, 或
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。

[Previously saved workspace restored]

> ?hist
> op <- par(mfrow=c(2, 2))
> hist(islands)
> hist(sqrt(islands), breaks = 12, col="lightblue", border="pink")
> hist(islands, breaks=c(12,20,36,80,200,1000,17000), freq = TRUE,
+      main = "WRONG histogram") # and warning
Warning message:
In plot.histogram(r, freq = freq1, col = col, border = border, angle = angle,$
  the AREAS in the plot are wrong -- rather use freq=FALSE
```

指令為逐行執行, 故在指令輸入後會馬上執行, 句尾不需加結束符號

按 “ESC” 可強制退出未完成的工作。

可利用鍵盤上下鍵重複輸入指令或做小幅度修改

How to use help?(1)

1. 查詢指令使用方法: ?+"指令名"(ex. ?write)

`write(base)`

()內為所屬package名

Write Data to a File

Description

描述此function的功能 The data (usually a matrix) `x` are written to file `file`. If `x` is a two-dimensional matrix you need to transpose it to get the columns in `file` the same as those in the internal representation.

Usage

使用語法

```
write(x, file = "data",
      ncolumns = if(is.character(x)) 1 else 5,
      append = FALSE, sep = " ")
```

Arguments

參數用法

- `x` the data to be written out.
- `file` A connection, or a character string naming the file to write to. If "", print to the standard output connection.
- `ncolumns` the number of columns to write the data in.
- `append` if TRUE the data `x` are appended to the connection.
- `sep` a string used to separate columns. Using `sep = "\t"` gives tab delimited output; default is " ".

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

`write` is a wrapper for [cat](#), which gives further details on the format used.

[save](#) for writing any R objects, [write.table](#) for data frames, and [scan](#) for reading data.

Examples

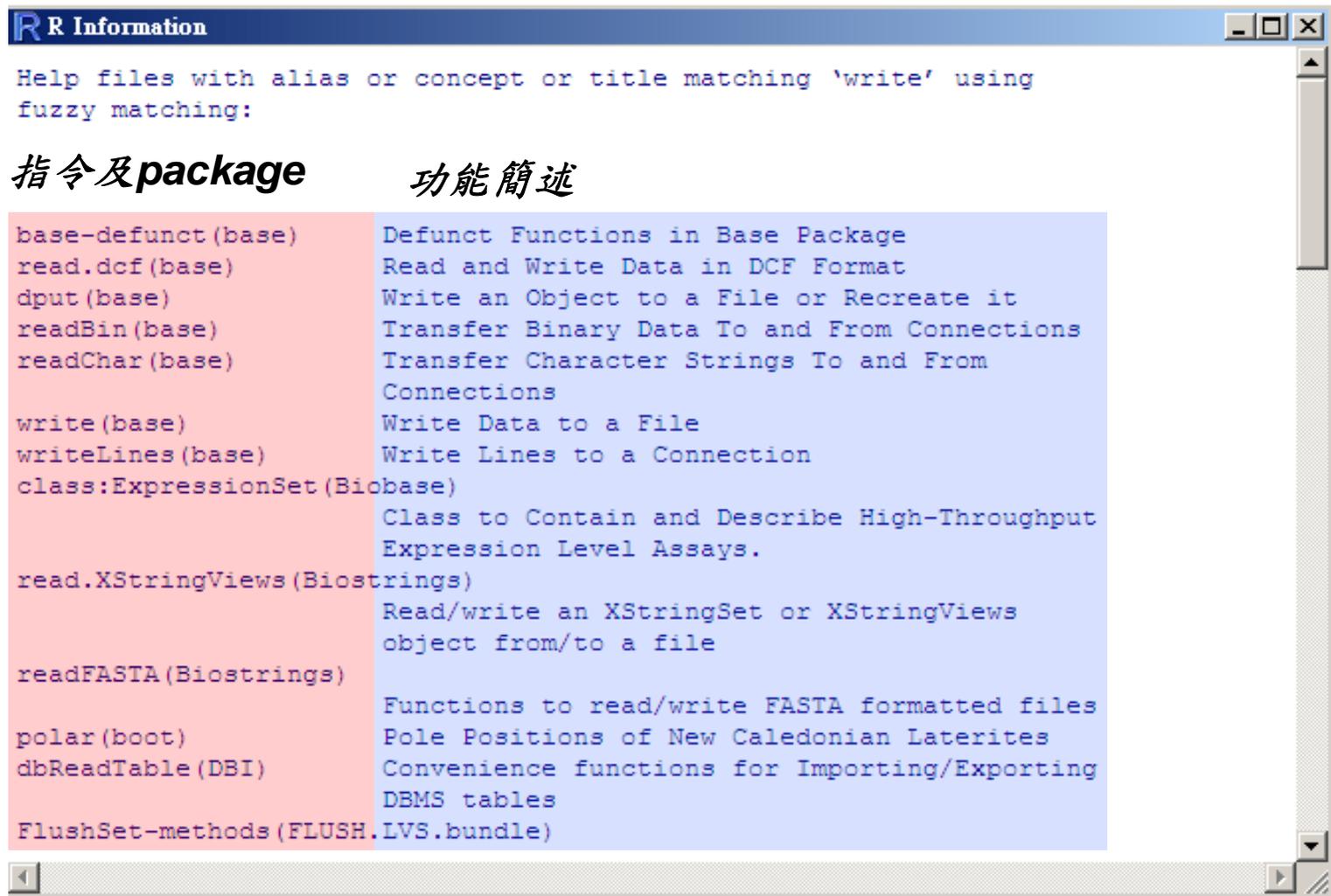
範例

```
# create a 2 by 5 matrix
x <- matrix(1:10, ncol=5)

# the file data contains x, two rows, five cols
# 1 3 5 7 9 will form the first row
write(t(x))
```

How to use help?(2)

2. 如果連指令都不知道: `help.search(“keyword”)` (ex. `help.search(“write”)`)



The screenshot shows an R Information window titled "R Information" with a search result for the keyword "write". The window displays a list of functions and packages that match the search criteria, along with their descriptions. The text is presented in a table-like format with two columns: "指令及package" (Command and package) and "功能簡述" (Brief description).

指令及package	功能簡述
<code>base-defunct(base)</code>	Defunct Functions in Base Package
<code>read.dcf(base)</code>	Read and Write Data in DCF Format
<code>dput(base)</code>	Write an Object to a File or Recreate it
<code>readBin(base)</code>	Transfer Binary Data To and From Connections
<code>readChar(base)</code>	Transfer Character Strings To and From Connections
<code>write(base)</code>	Write Data to a File
<code>writeLines(base)</code>	Write Lines to a Connection
<code>class:ExpressionSet(Biobase)</code>	Class to Contain and Describe High-Throughput Expression Level Assays.
<code>read.XStringViews(Biostrings)</code>	Read/write an XStringSet or XStringViews object from/to a file
<code>readFASTA(Biostrings)</code>	Functions to read/write FASTA formatted files
<code>polar(boot)</code>	Pole Positions of New Caledonian Laterites
<code>dbReadTable(DBI)</code>	Convenience functions for Importing/Exporting DBMS tables
<code>FlushSet-methods(FLUSH.LVS.bundle)</code>	

*R*的*package*(1)

起始套件

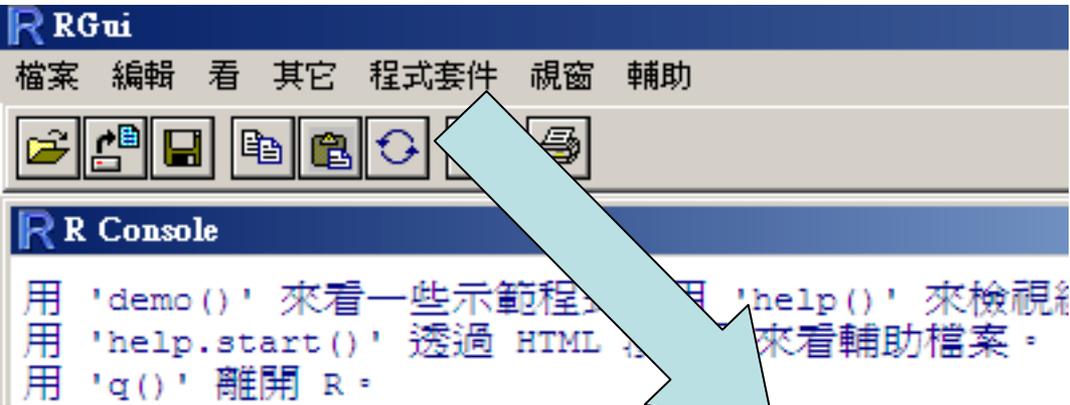
- 當R 啟動時, 有7 個常用之**packages** 會自動載入:
 - **base**: 基本函式(**IO**, 敘述統計, etc.)
 - **stats**: 常用統計分析(**t.test**, **anova**, etc.)
 - **methods**: 定義**classes of objects**
 - **utils**: 基本程式編寫工具
 - **graphics**: 基本繪圖工具(**boxplot**,**plot**,**line**)
 - **grDevices**: 基本繪圖介面
 - **datasets**: 數據範例

*R*的*package*(2)

常用的package: (建議安裝)

1. **limma(biocLite)**: 提供許多分析microarray data的工具，主要包括image的繪製，normalization的function，及許多統計功能
2. **marray(biocLite)**: 提供許多方便的microarray data的輸入格式，使用者可以容易的將各種格式的microarray輸入R做許多常用的處理，可以方便的用簡單的指令快速處理
3. **RMySQL**: 連接MySQL資料庫，可直接在R及資料庫間作存取動作(另外需要CRAN的DBI package)

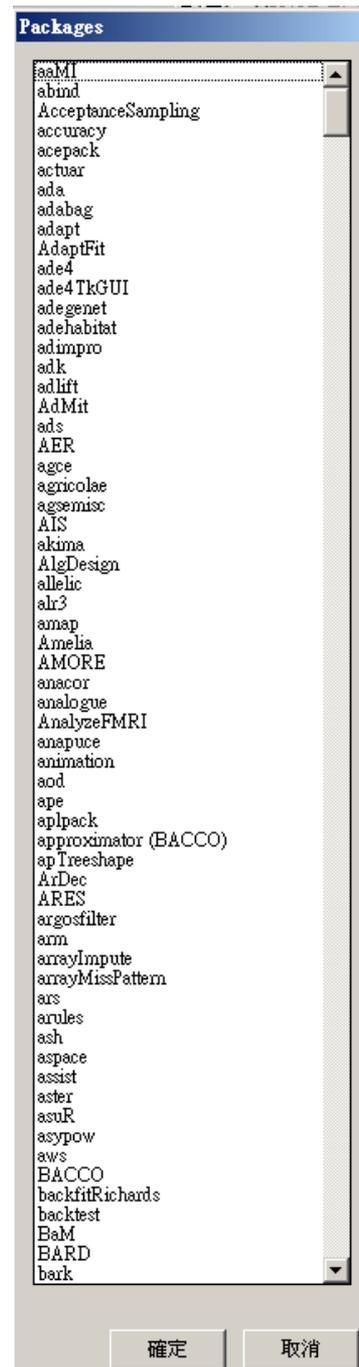
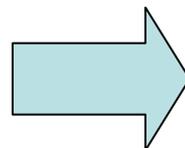
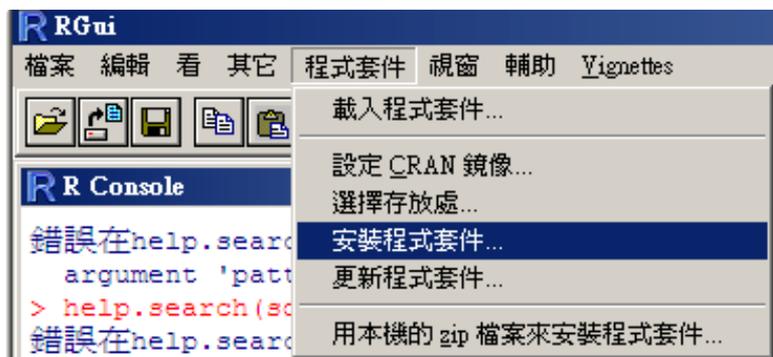
如何新增其他package(在CRAN)



使用介面 - Packages

- | | | |
|-----------------------|---|---|
| 載入程式套件... | — | 在目前工作環境中載入套件→ <code>library(package)</code> |
| 設定 CRAN 鏡像... | — | 設定下載 package 的網站(Taipei, etc.) |
| 選擇存放處... | — | 設定下載 package 的搜尋範圍 (cran, etc.) |
| 安裝程式套件... | — | package 網路安裝 |
| 更新程式套件... | — | package 更新 (已安裝之 package) |
| 用本機的 zip 檔案來安裝程式套件... | — | package 本機安裝(已下載) |

如何新增package



Start Coding...

1. 前面的步驟準備好了，R就有分析**microarray data**的基本功能
2. 以下會介紹一些基本的運算子，物件格式，基本的統計**function**
3. 以下是以邊講解邊練習的方式進行，操作有問題請隨時提出...

基本運算子

運算基本指令

- +, -, *, / (加減乘除)
- %(餘數), %/% (相除的商), %*%(矩陣相乘), t(轉置)
- ^ (次方, 2^3)

如要先行計算可用()規範

練習:

1. $6+4*7-5*2$ (符合先乘除後加減)
2. $4*2^2-2*10$ (次方先於加減乘除)
3. $(6+4)*7-5*2$

基本運算函數

運算基本指令

- **sqrt(x)**
- **exp(x), log(x), log10(x), log2(x)**
- **abs(x), round(x)**

練習:

1. **sqrt(36)+4**
2. **log2(4), log(1), log(0), log(-5)**
3. **abs(-10+7), round(5/3,2)**

常用物件(object)格式

1. 單一變數格式:

種類	範例	特性
數字	5, -7, 1.001, 1e+14	可以用運算子計算，可以轉成字串使用
字串	"NTU"	無法使用運算子
其他	NA, NULL	為空值
邏輯	FALSE, TRUE	使用運算子時， FALSE 轉成0， TRUE 轉成1

練習:

1. $1.2e+5+300$

2. `paste("NTU",5)`

3. $2+NA$

4. $(TRUE+TRUE)/3+FALSE*2$

2. 複合物件:

為單一物件的組合

常用複合物件格式

種類	範例	特性
向量	<code>c(10,2,4,9)</code>	一定順序的數字或文字
<code>matrix</code>	<code>matrix(1:6,n row=2, ncol=3)</code>	具行列位置的向量
<code>Data.frame</code>	<pre>X1 X2 1 1 3 2 2 4</pre>	具表格屬性的資料

練習：

1. `c(2,5,7,9)[3]`
2. `1:6[4]`
3. `seq(1,6,0.5)[4]`
4. `c(2,5,7,9,"NTU")`
5. `matrix(1:12,nrow=4, ncol=3)`
6. `matrix(1:12,nrow=4, ncol=3)[2,3]`
7. `as.data.frame(matrix(1:12,nrow=4, ncol=3))`
8. `as.data.frame(matrix(1:12,nrow=4, ncol=3))$V2`

變數指定

在R的data操作中，我們習慣用指定名稱的物件來操作data

方法: 名稱= 物件 or 名稱← 物件

物件名稱規則:

- 可由英文字母、數字、英文句點(.)組成。
- 英文大小寫有所區別(ab 與Ab 可分別代表不同變數)。
- 變數名稱須起始於英文字母。
- 有些英文單字具有特殊意義，不能當做變數名稱: return, break, if, TRUE, FALSE, (T, F).

練習:

1. `n1 <- 3`
2. `v1 <- 1:6`
3. `t1 <- as.data.frame(matrix(7:12,nrow=2,ncol=3))`
4. `n1*v1[3]+t1`

Matrix and Data.frame

```
table <- as.data.frame(matrix(1:18,ncol=6))
```

`table[2,4]`

`table[2:3,4:5]`

`table[,6]`

`table$V2`

	V1	V2	V3	V4	V5	V6
1	1	4	7	10	13	16
2	2	5	8	11	14	17
3	3	6	9	12	15	18

!

比較運算子

可用來比較數值，設定篩選條件...，回傳結果為FALSE/TRUE

比較運算基本指令

- $<$, $>$, $==$, $<=$, $>=$, $!=$
- $\&$, $|$

練習:

1. $1*9<5+2$ (比較指令兩邊先運算)
2. $v1==n1$
3. $t1<9$
4. $t1[t1>=9]$ (格式不再為data.frame)
5. $v1[v1!=n1 \ \& \ v1!=2*n1]$

Data Input(1)···vector format

在輸入的方式上，除了可以在R上面直接指定變數外，也可以從已有的資料作匯入

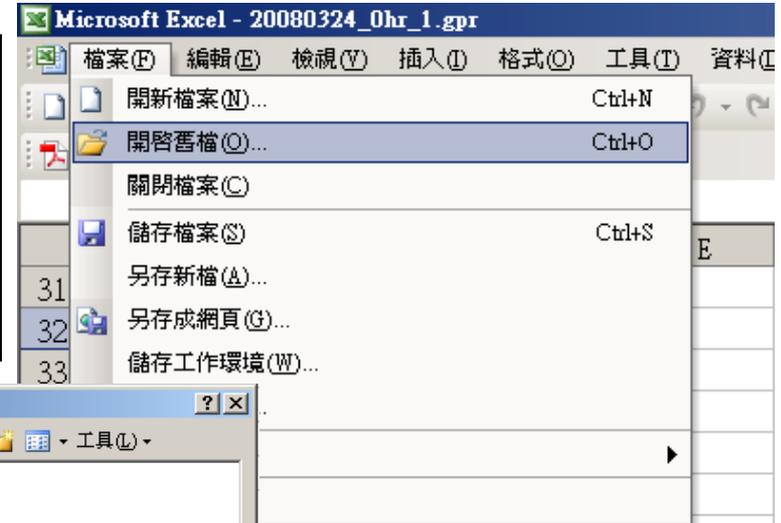


`v2 <- scan("c:/test.txt", sep = " ")`
檔名 以"空白"分隔

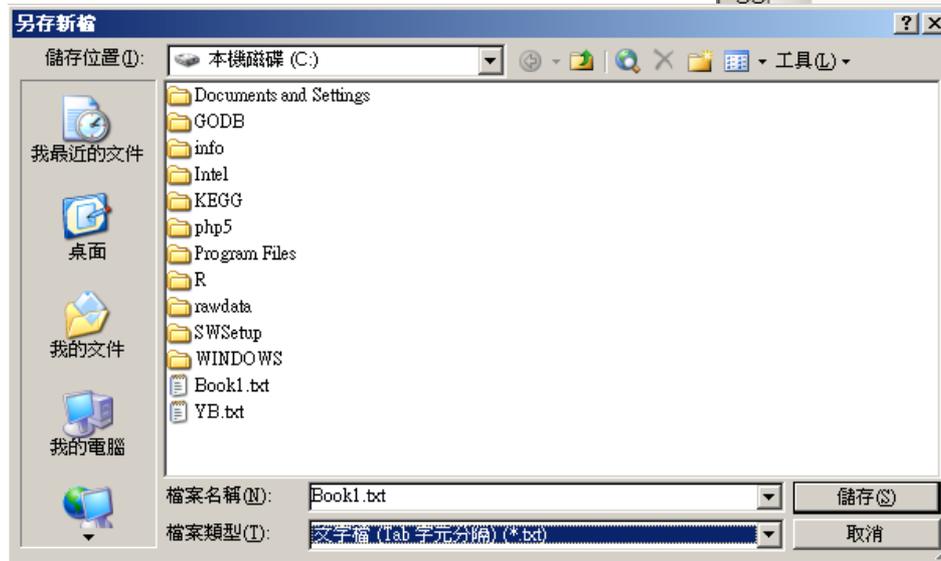
```
> v2 <- scan("c:/test.txt", sep = " ")  
Read 6 items  
> v2  
[1] 12 30 44 25 56 77
```

Data Input(2)··table format

通常我們使用Excel是以本來已經存在的表格載入後做處理或畫圖...在R中也是如此，只是以寫code的方式處理...



	A	B
1	GeneName	intensity
2	aa	100
3	bb	1000
4	cc	10



```
t2 <- read.table("c:/Book1.txt", header=TRUE, sep = "\t")
```

檔名 最上面一行為標題 以Tab分隔

```
> t2 <- read.table("c:/Book1.txt", header=TRUE, sep = "\t")
> t2
  GeneName intensity
1      aa         100
2      bb        1000
3      cc          10
```

Data output(1)···vector format

在資料處理結束後，物件還在R的環境中，當R被關閉時，物件即會消失(跟excel沒存檔一樣)，所以必須下一個命令使物件資料儲存成檔案

```
v3 <- v2*3-5
```

```
write(v3, "c:/v3.txt", sep = "\t")
```

物件名 檔案路徑 以**Tab**分隔



Data output(2)···table format

經常處理的資料實際上是以table為主，所以常使用到的是write.table這個指令

```
t3 <- t2[1:2,]
```

```
t3[,2] <- t3[,2]*2
```

```
write.table(t3, "c:/t3.txt", col.names=TRUE, row.names=FALSE, sep="\t")
```

物件名 檔名

附註

以Tab分隔

	A	B
1	GeneName	intensity
2	aa	200
3	bb	2000

常用統計函式

- 中央趨勢:
 - > `mean(x)` # 樣本均值
 - > `median(x)` # 樣本中位數
 - > `quantile(x,p)` # 樣本100p% 百分位數

- 變異程度:
 - > `range(x)` # 最小值與最大值
 - > `var(x)` # 樣本變方
 - > `sd(x)` # 樣本標準差

- 兩變數的相關程度:
 - > `cov(x,y)` # 樣本共變方
 - > `cor(x,y)` # 樣本相關係數

練習:

以`x=1:10`, `y=seq(12,30,2)`做以上函數的練習