A Marvelous R - Foundation

免費統計軟體 – R

育達商業科技大學

資訊管理系

李明昌 (Ming-Chang Lee)

Email: alan9956@ydu.edu.tw

WEPA site: http://web.ydu.edu.tw/~alan9956/

July 4, 2011

Outline

1. Basic R
2. Preparing Data
3. Graphics
4. Applied Statistics
5. Application

Contents

1. Basic R

1.1 Introduction

1.2 Installing R

1.3 Packages

1.4 R Journal

1.5 R Conference

1.6 R Search

1.1 Introduction

- R is used for statistical computing and data visualization.
- R is rooted in S language that was developed at AT&T Bell Laboratories by Rick Becker, John Chambers and Allan Wilks.
- Versions of R are available:
  - Microsoft Windows, Linux, Unix, Macintosh OS X
  - R 1.0.0 (February, 2000) - 8.43 MB
  - R 2.13.0 for Windows (June, 2011) - 37.9 MB

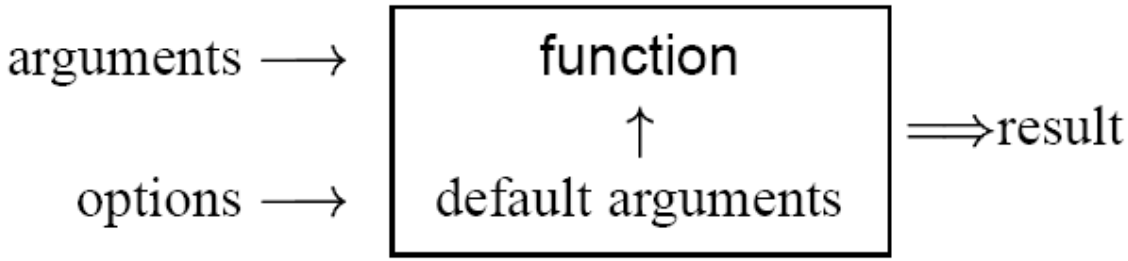
Features

- An effective data handling.
- A suite of operators for calculations on arrays, in particular matrices.
- A large, coherent, integrated collection of intermediate tools for data analysis.
- Graphical facilities for data analysis and display either on-screen or on hardcopy.
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.
- Link to complied language, e.g. C, C++, FORTRAN

How R works

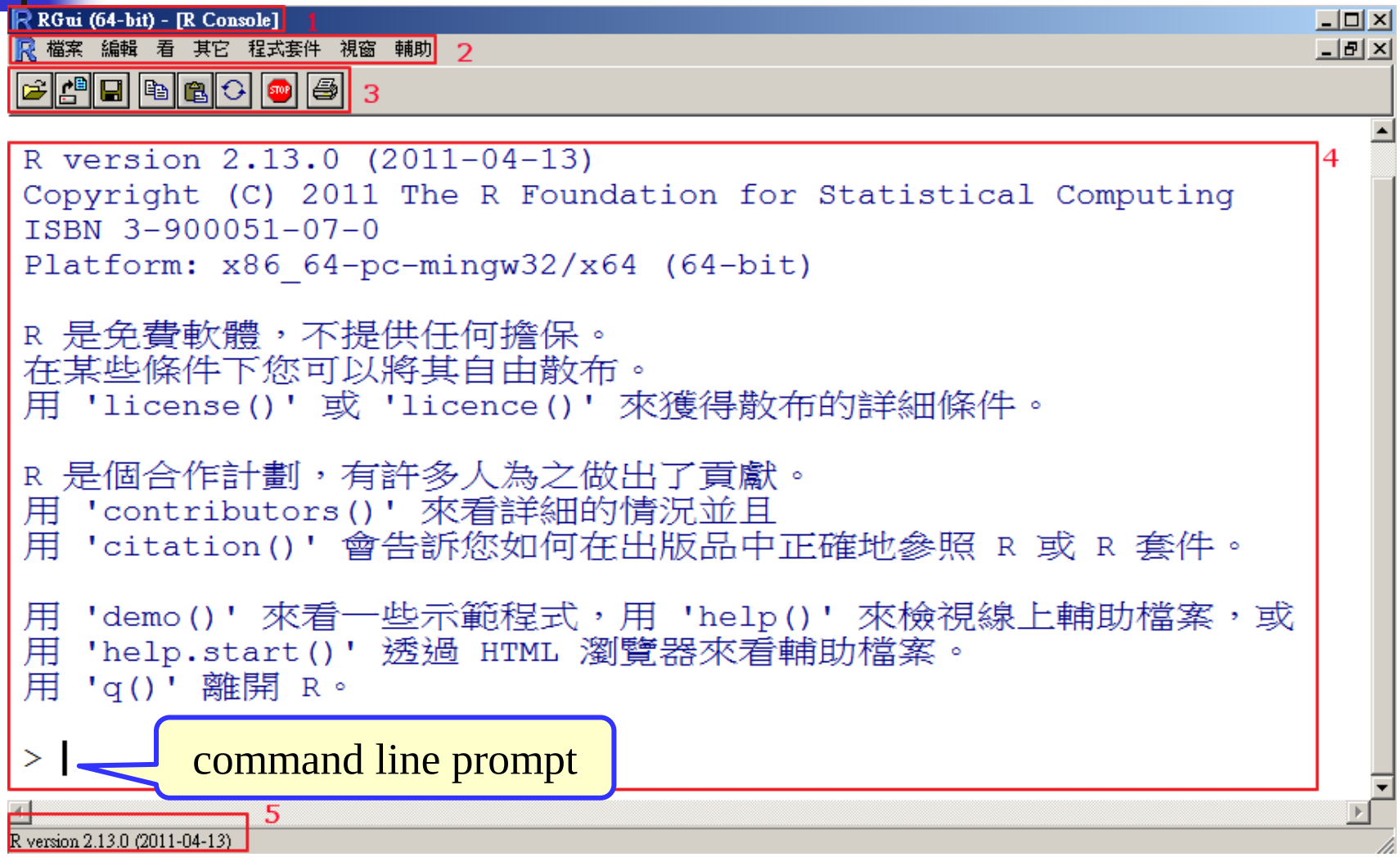
- An interpreted language (Not a complied language).
- An object-oriented language - variables, data, functions, result are stored in the forms of objects.

R function



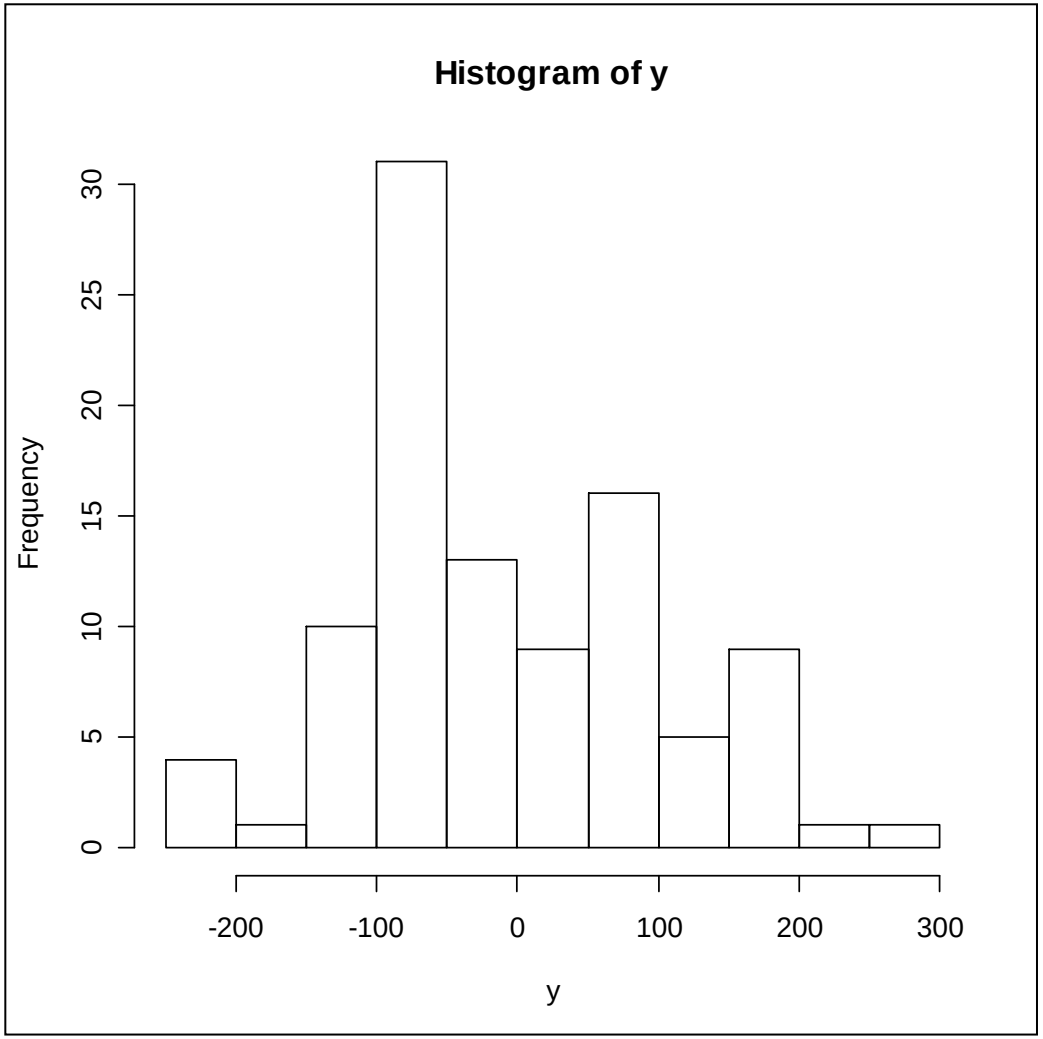
- Arguments:
  data, formulate, expressions.
- Packages of function:
  C:\Program Files\R\R-2.13.0\library

R - Environment



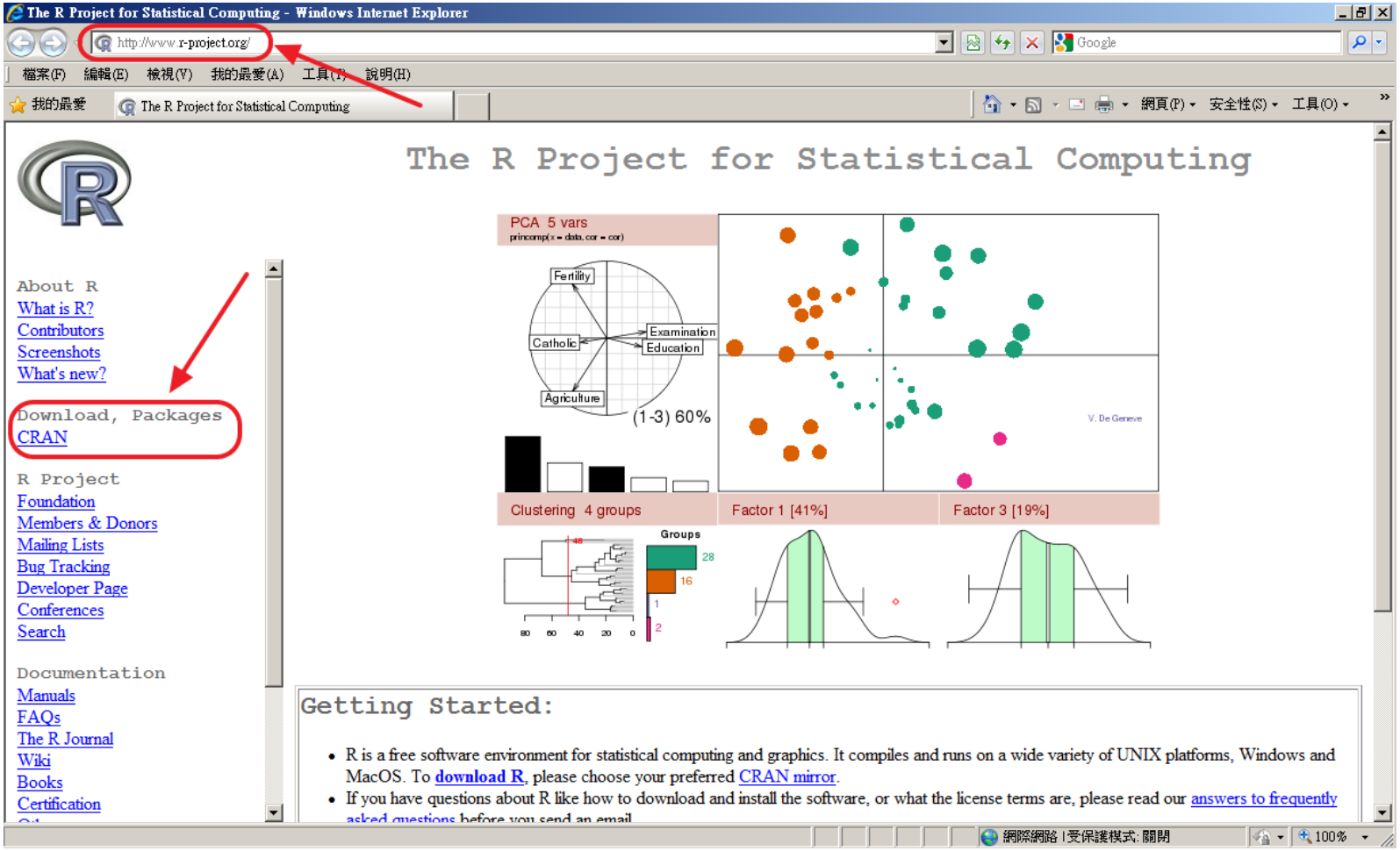
Example

```
> # p.10
> #
> # hist and plot
> x <- c(1:100)
> y <- rnorm(100)*100
> hist(y)
> test.model <- lm(y ~ x)
> test.model

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    20.1824      -0.2101

> plot(x,y)
> library(help="graphics")
> # end
>
> # p.10
> #
> # On-line help:? …
> ? rnorm
starting httpd help server ... done
> ? plot
> #Information on package 'base'
> library(help="base")
> # end
```

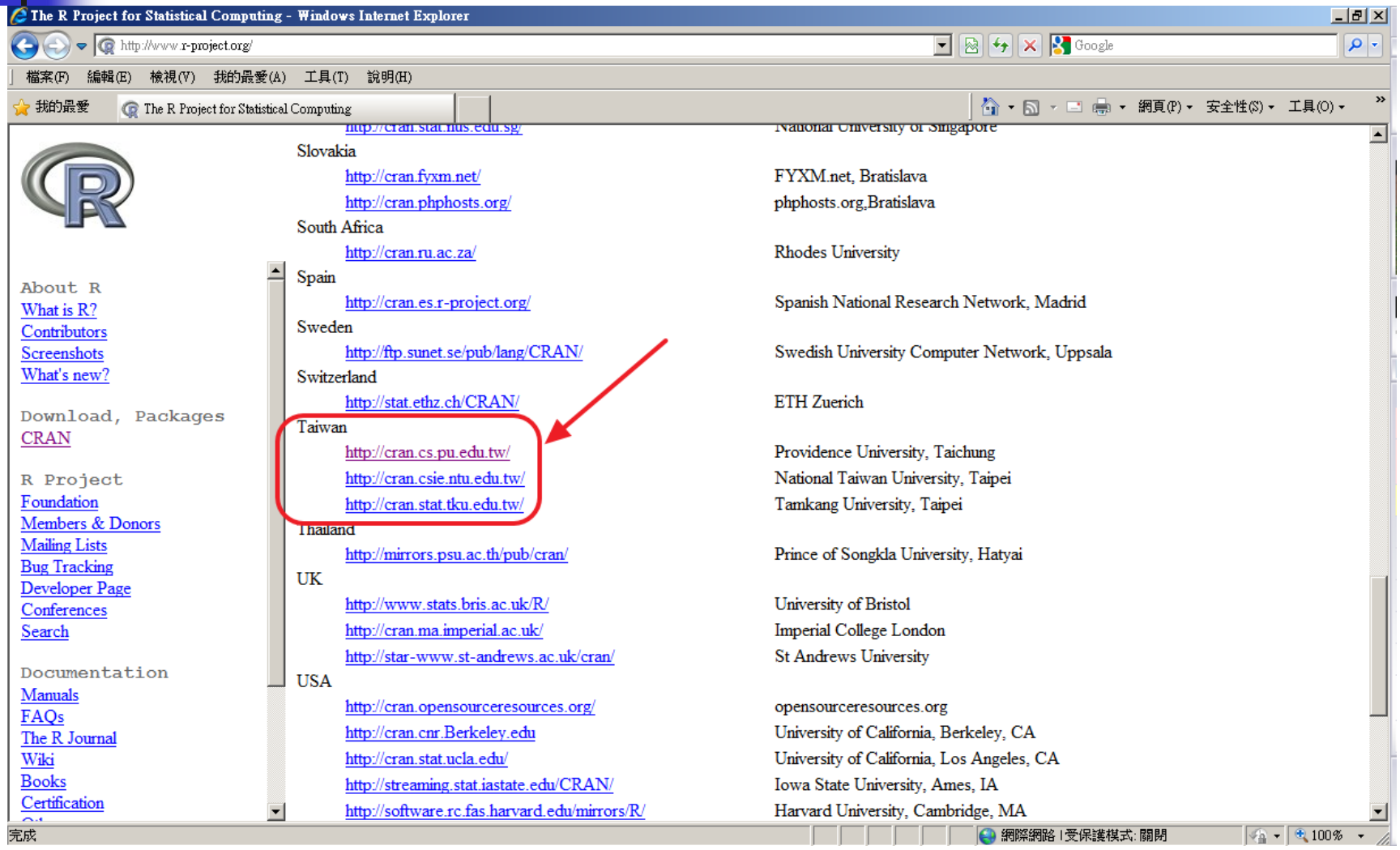


1.2 Installing R

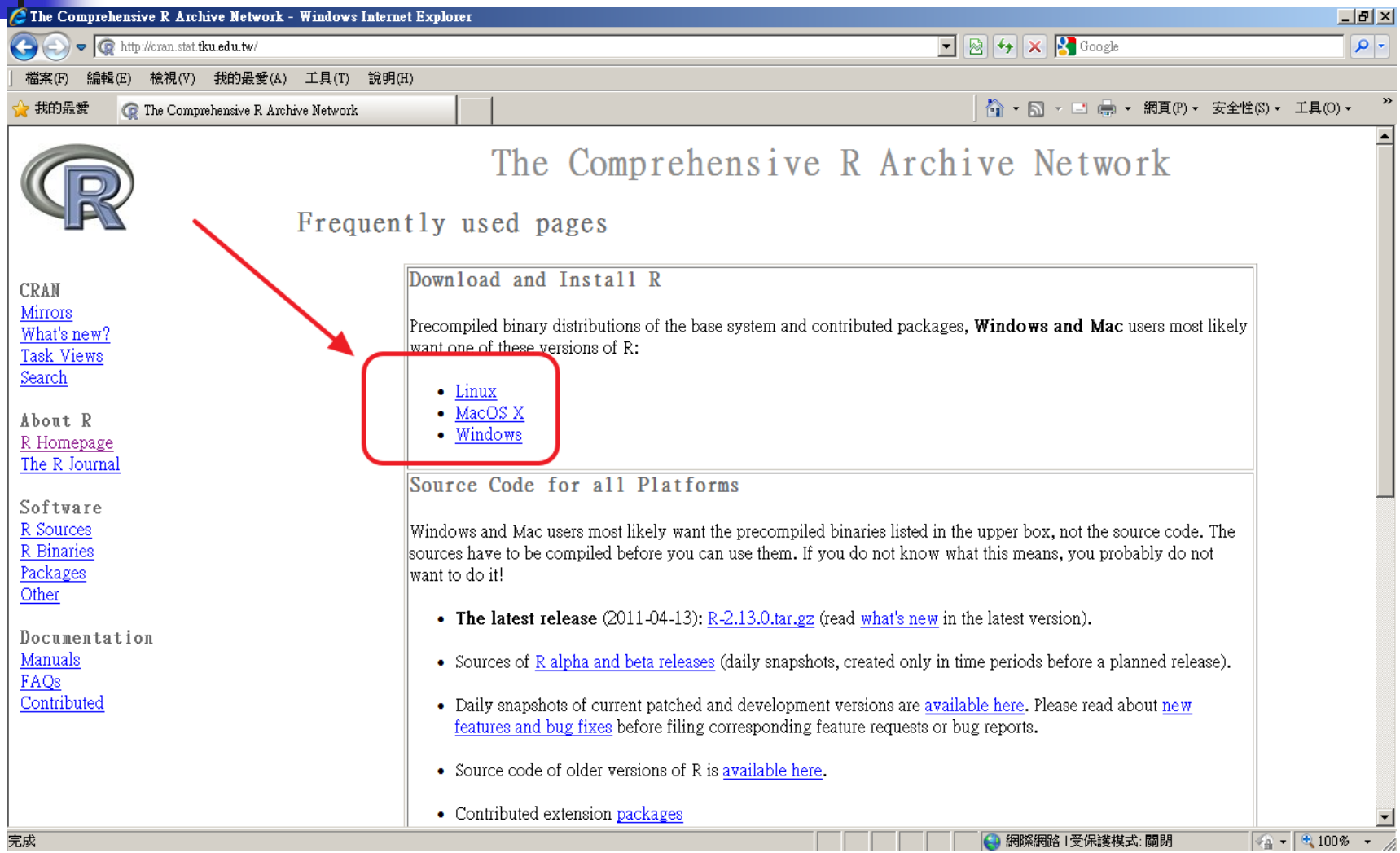
http://www.r-project.org/ → CRAN



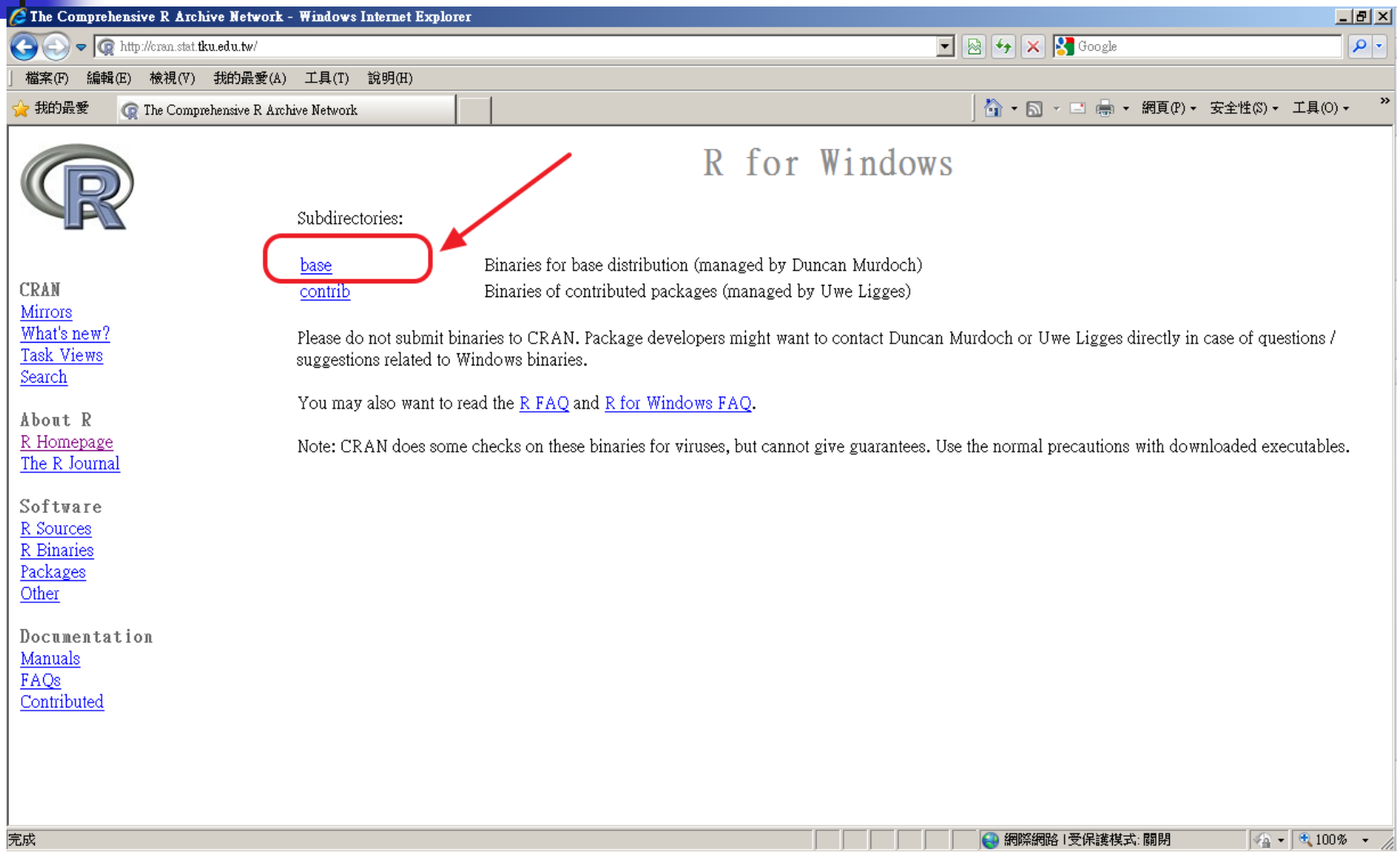
Mirror site


The R Project for Statistical Computing - Windows Internet Explorer
http://www.r-project.org/
Google
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
我的最愛
The R Project for Statistical Computing
網頁(P) 安全性(S) 工具(O)
National University of Singapore
Slovakia
http://cran.fyxm.net/ FYXM.net, Bratislava
http://cran.phphosts.org/ phphosts.org,Bratislava
South Africa
http://cran.ru.ac.za/ Rhodes University
Spain
http://cran.es.r-project.org/ Spanish National Research Network, Madrid
Sweden
http://ftp.sunet.se/pub/lang/CRAN/ Swedish University Computer Network, Uppsala
Switzerland
http://stat.ethz.ch/CRAN/ ETH Zuerich
Taiwan
http://cran.cs.pu.edu.tw/ Providence University, Taichung
http://cran.csie.ntu.edu.tw/ National Taiwan University, Taipei
http://cran.stat.tku.edu.tw/ Tamkang University, Taipei
Thailand
http://mirrors.psu.ac.th/pub/cran/ Prince of Songkla University, Hatyai
UK
http://www.stats.bris.ac.uk/R/ University of Bristol
http://cran.ma.imperial.ac.uk/ Imperial College London
http://star-www.st-andrews.ac.uk/cran/ St Andrews University
USA
http://cran.opensourceresources.org/ opensourceresources.org
http://cran.cnr.Berkeley.edu University of California, Berkeley, CA
http://cran.stat.ucla.edu/ University of California, Los Angeles, CA
http://streaming.stat.iastate.edu/CRAN/ Iowa State University, Ames, IA
http://software.rc.fas.harvard.edu/mirrors/R/ Harvard University, Cambridge, MA
About R
What is R?
Contributors
Screenshots
What's new?
Download, Packages
CRAN
R Project
Foundation
Members & Donors
Mailing Lists
Bug Tracking
Developer Page
Conferences
Search
Documentation
Manuals
FAQs
The R Journal
Wiki
Books
Certification
完成
網際網路 | 受保護模式: 關閉
100%

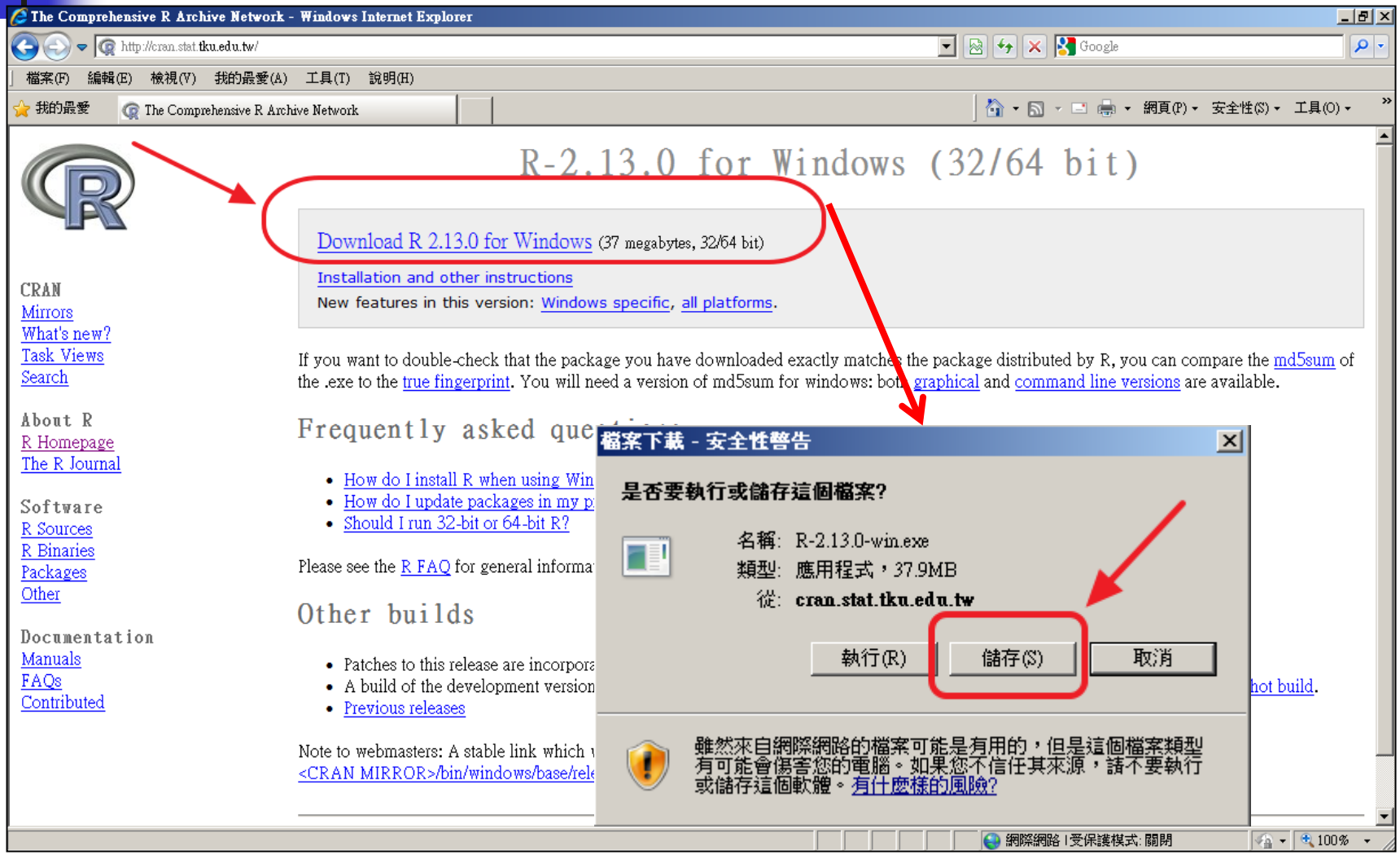
OS: Windows


The Comprehensive R Archive Network - Windows Internet Explorer
http://cran.stat.tku.edu.tw/
The Comprehensive R Archive Network
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 (2011-04-13): R-2.13.0.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

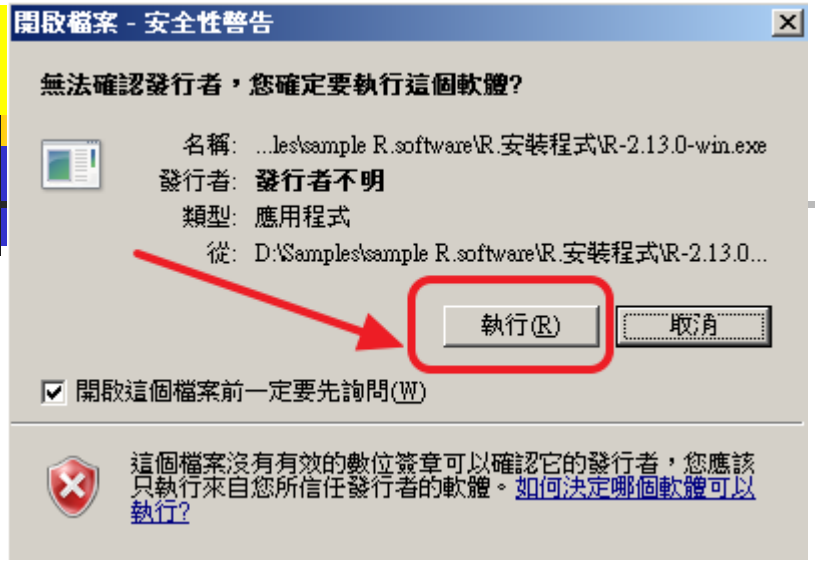
Base distribution



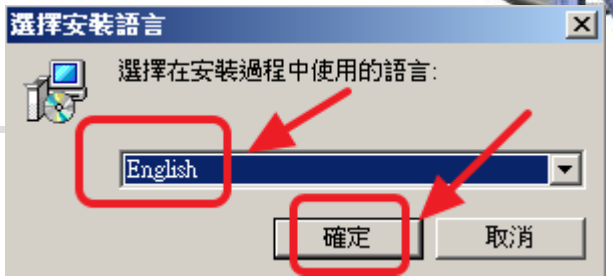
R 2.13.0 for Windows



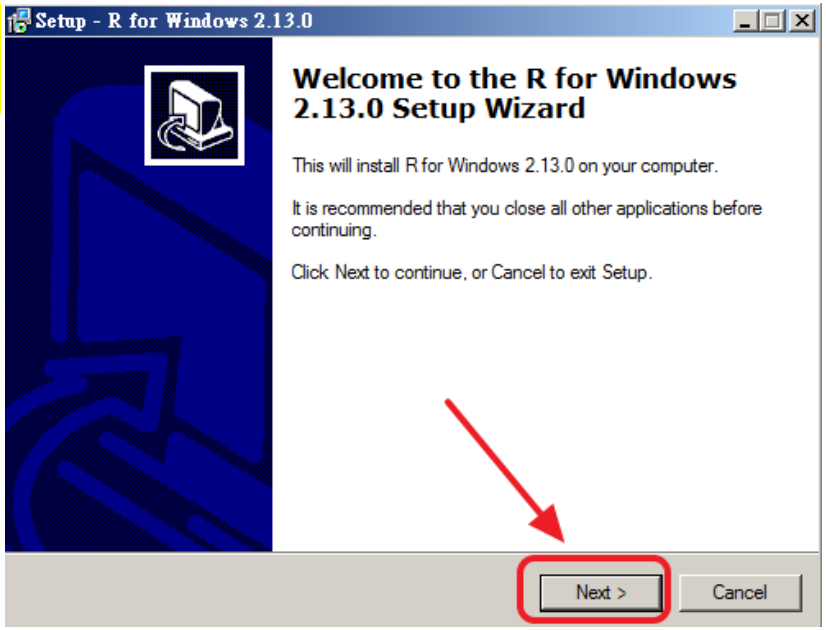
1


開啟檔案 - 安全性警告
無法確認發行者，您確定要執行這個軟體?
名稱: ...les\sample R.software\R.安裝程式\R-2.13.0-win.exe
發行者: 發行者不明
類型: 應用程式
從: D:\Samples\sample R.software\R.安裝程式\R-2.13.0...
執行(R)
取消
開啟這個檔案前一定要先詢問(W)
這個檔案沒有有效的數位簽章可以確認它的發行者，您應該只執行來自您所信任發行者的軟體。如何決定哪個軟體可以執行?

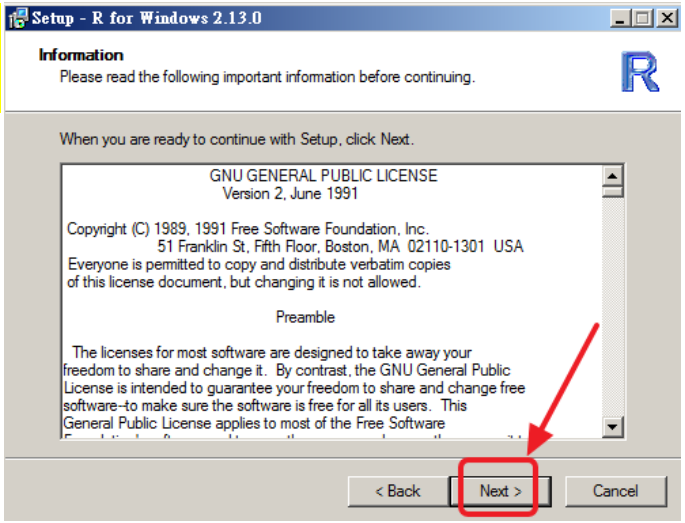
2


選擇安裝語言
選擇在安裝過程中使用的語言:
English
確定
取消

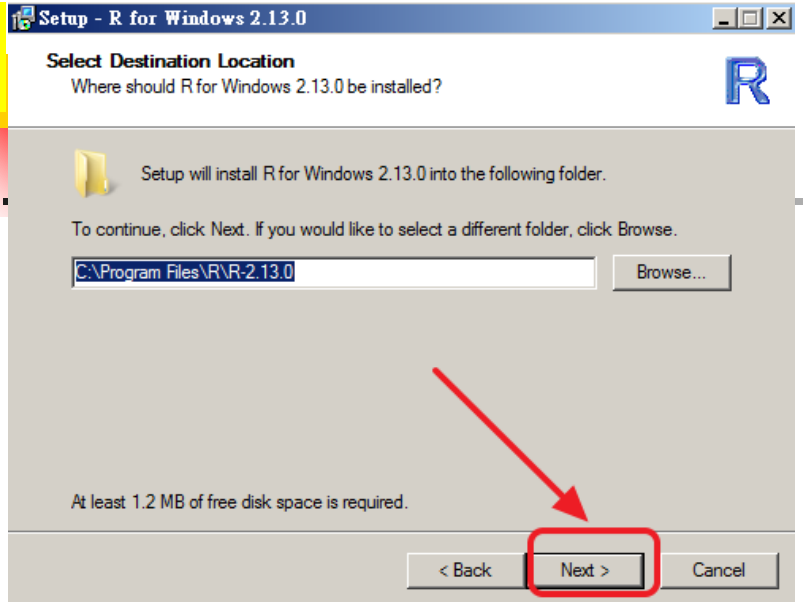
3


Setup - R for Windows 2.13.0
Welcome to the R for Windows 2.13.0 Setup Wizard
This will install R for Windows 2.13.0 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

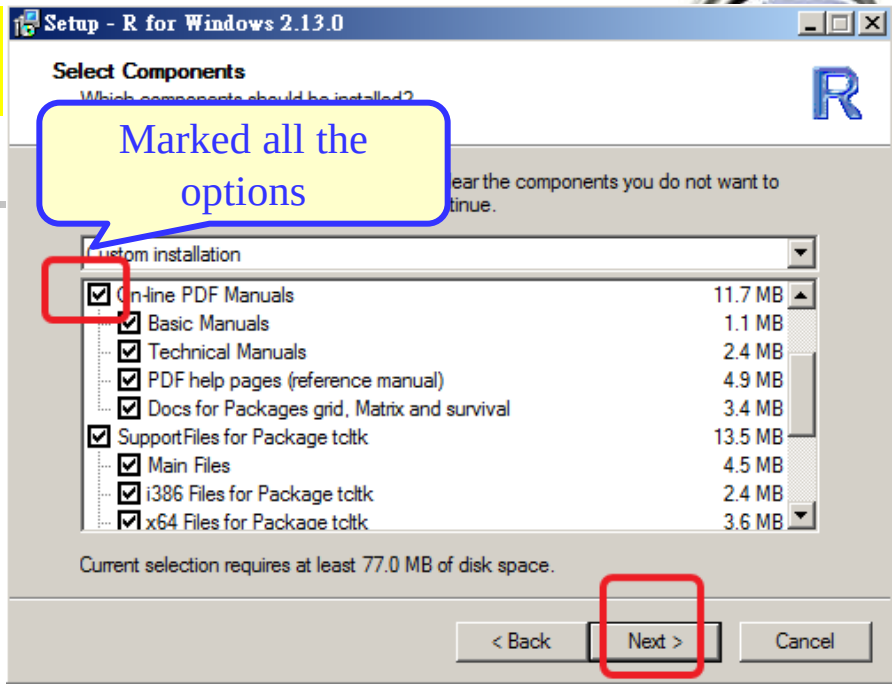
4


Setup - R for Windows 2.13.0
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software
< Back
Next >
Cancel

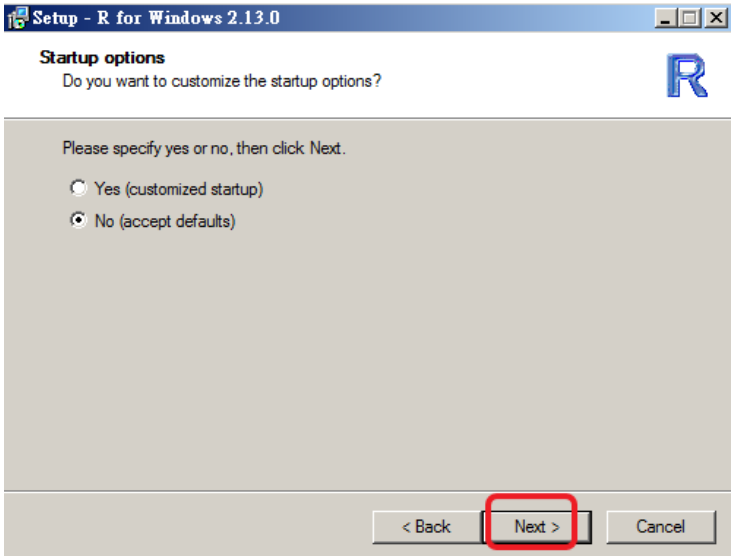
5


Setup - R for Windows 2.13.0
Select Destination Location
Where should R for Windows 2.13.0 be installed?
Setup will install R for Windows 2.13.0 into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\R\R-2.13.0
Browse...
At least 1.2 MB of free disk space is required.
< Back
Next >
Cancel

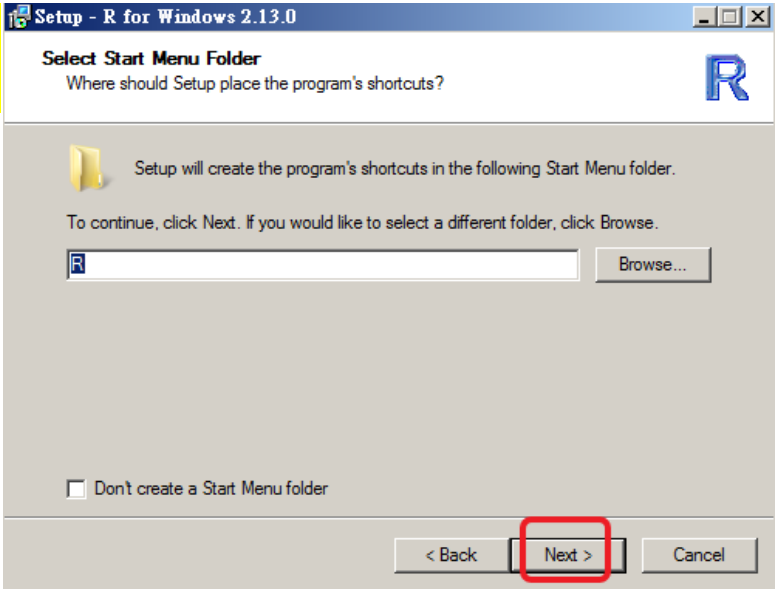
6


Setup - R for Windows 2.13.0
Select Components
Marked all the options
ear the components you do not want to
tinue.
Custom installation
On-line PDF Manuals 11.7 MB
Basic Manuals 1.1 MB
Technical Manuals 2.4 MB
PDF help pages (reference manual) 4.9 MB
Docs for Packages grid, Matrix and survival 3.4 MB
SupportFiles for Package tcltk 13.5 MB
Main Files 4.5 MB
i386 Files for Package tcltk 2.4 MB
x64 Files for Package tcltk 3.6 MB
Current selection requires at least 77.0 MB of disk space.
< Back
Next >
Cancel

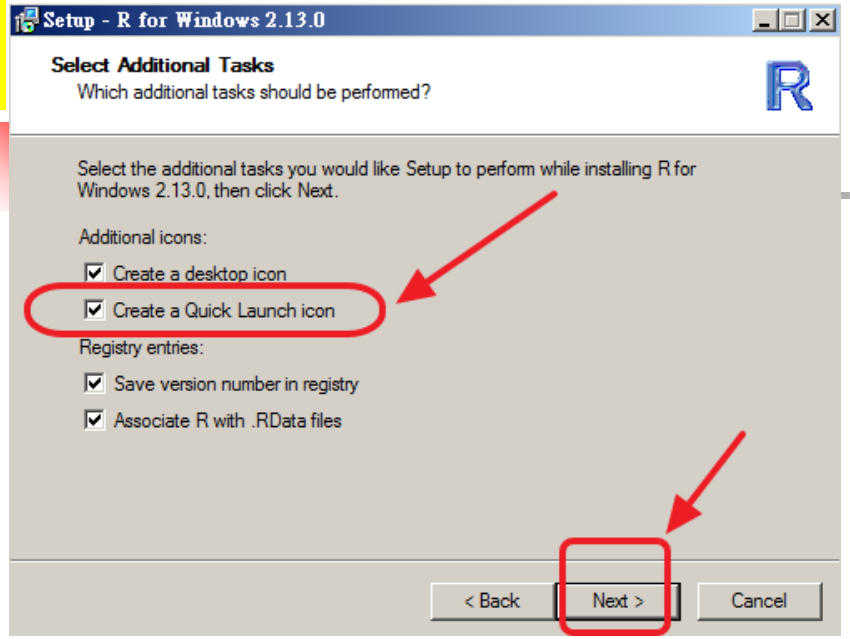
7


Setup - R for Windows 2.13.0
Startup options
Do you want to customize the startup options?
Please specify yes or no, then click Next.
Yes (customized startup)
No (accept defaults)
< Back
Next >
Cancel

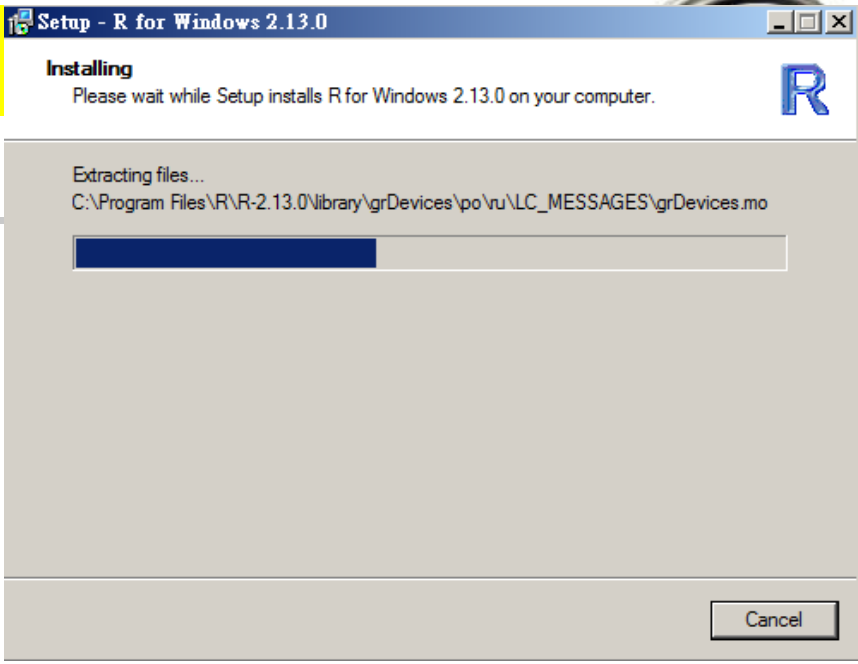
8


Setup - R for Windows 2.13.0
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
R
Browse...
Don't create a Start Menu folder
< Back
Next >
Cancel

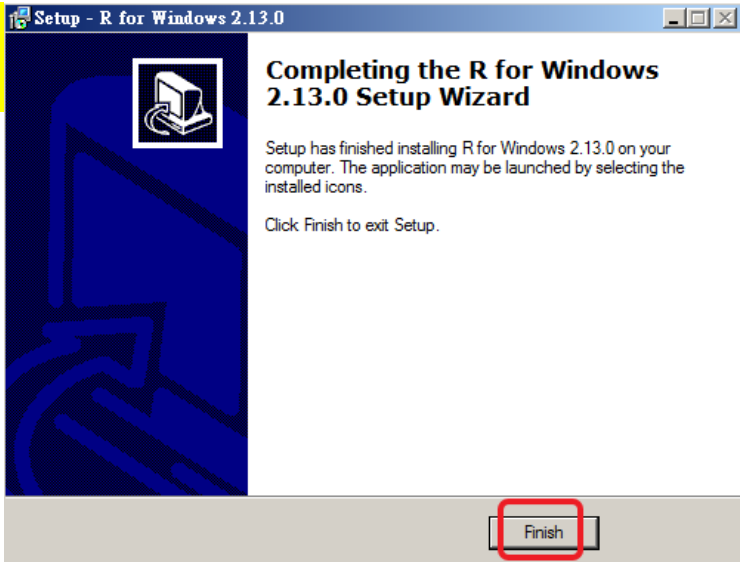
9



10

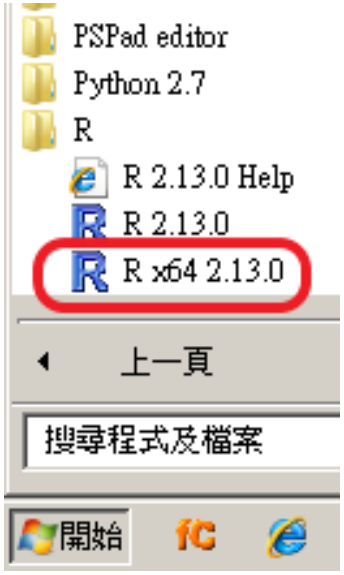


11

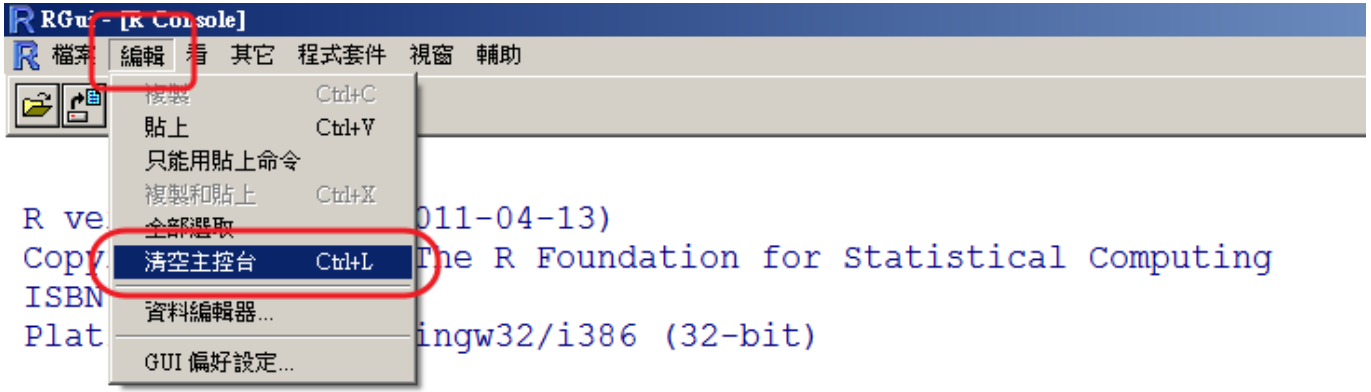


12

Run R

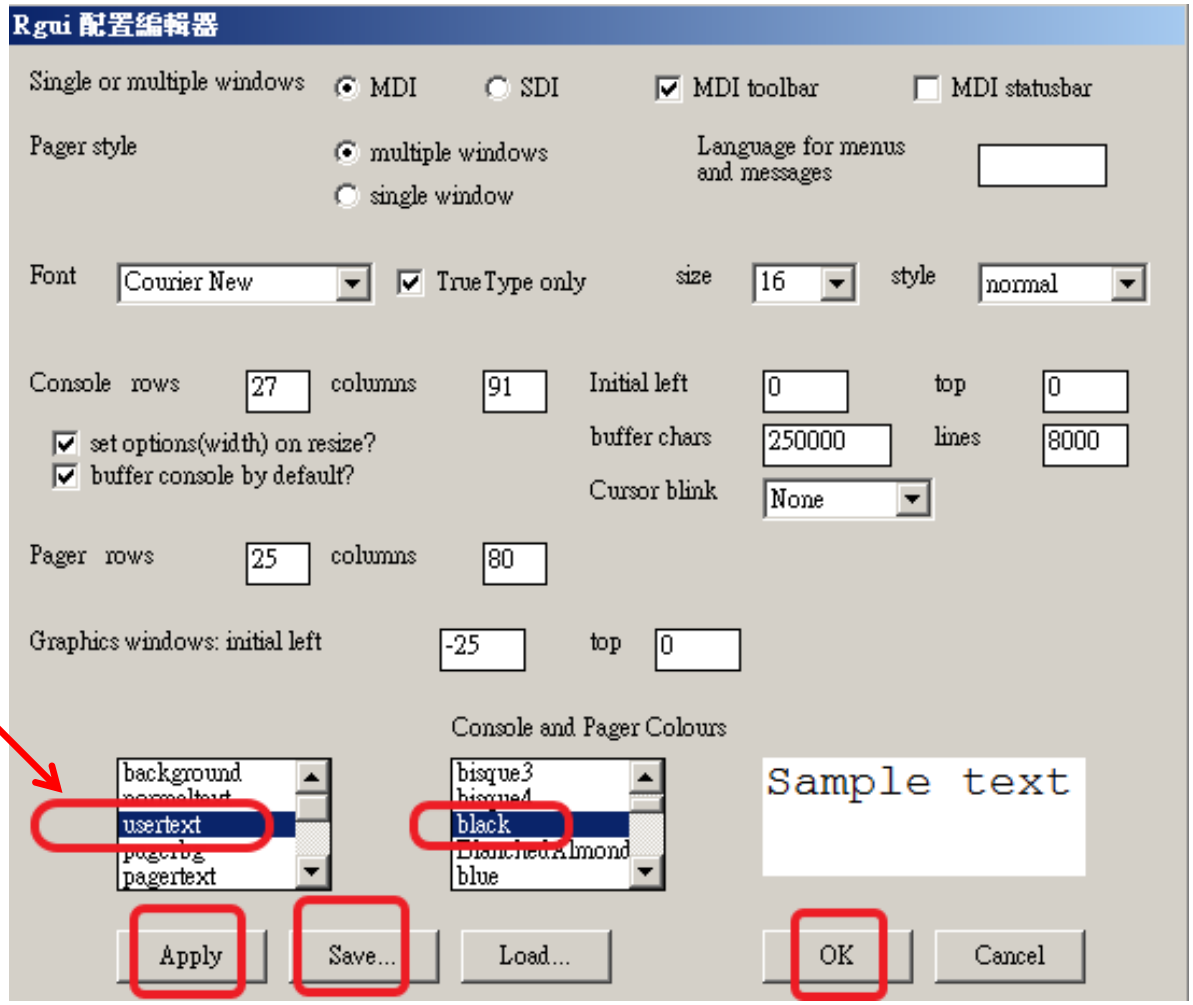


Launch R


RGui - [R Console]
檔案 編輯 看 其它 程式套件 視窗 輔助
複製 Ctrl+C
貼上 Ctrl+V
只能用貼上命令
複製和貼上 Ctrl+X
全部選取
清空主控台 Ctrl+L
資料編輯器...
GUI 偏好設定...
R ve 011-04-13)
Copy The R Foundation for Statistical Computing
ISBN
Plat ingw32/i386 (32-bit)
R 是免費軟體，不提供任何擔保。
在某些條件下您可以將其自由散布。
用 'license()' 或 'licence()' 來獲得散布的詳細條件。
R 是個合作計劃，有許多人為之做出了貢獻。
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。
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。
> # Clear console : Ctrl + L
>
>


RGui (64-bit) - [R Console]
檔案 編輯 看 其它 程式套件 視窗 輔助
1
檔案 編輯 看 其它
輸入 R 程式碼...
建立新的命令稿
開啟命令稿...
顯示檔案...
載入工作空間...
儲存工作空間...
載入命令歷程...
儲存命令歷程...
變更現行目錄...
列印...
儲存到檔案...
離開
2
編輯
Ctrl+C
貼上 Ctrl+V
只能用貼上命令
複製和貼上 Ctrl+X
全部選取
清空主控台 Ctrl+L
資料編輯器...
GUI 偏好設定...
3
看 其它
Toolbar
Statusbar
4
其它 程式套件 視窗 輔助
停止目前的計算 ESC
Stop all computations
緩衝區輸出 Ctrl+W
Word completion
Filename completion
列出目的物件
刪除所有的目的物件
List search path
5
程式套件 視窗 輔助
載入程式套件...
設定 CRAN 鏡像...
選擇存放處...
安裝程式套件...
更新程式套件...
用本機的 zip 檔案來安裝程式套件...
6
視窗 輔助
層疊
Tile Horizontally
Tile Vertically
排列圖示
1 R Console
7
輔助
主控台
關於 R 的 FAQ
關於 Windows 上 R 的 FAQ(W)
手冊 (PDF檔案)
R 函式 (文字檔案)...
Html 輔助
搜尋輔助...
search.r-project.org ...
關於...
R 首頁
CRAN 首頁
關於

Modify Usertext color



Try it

```
> # p.22
> x <- c(1:10)
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.25    5.50    5.50    7.75   10.00
> # end
```

1.3 Packages

- A package is a related set of functions, help files, and data files that have been bundled together.
- Packages in R are similar to modules in Perl, libraries in C/C++, and classes in Java.
- Package:
  - Base packages
  - available packages
- Currently, the CRAN package repository features 3090 available packages (June, 2011).

CRAN Task Views (#28)

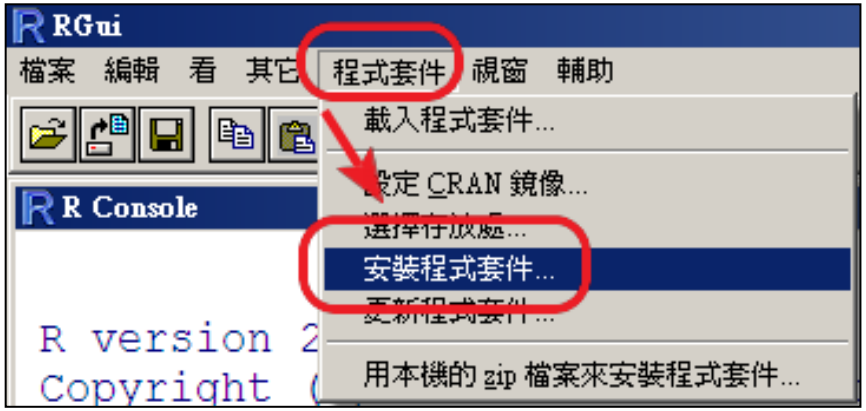
1	Bayesian	Bayesian Inference
2	ChemPhys	Chemometrics and Computational Physics
3	ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
4	Cluster	Cluster Analysis & Finite Mixture Models
5	Distributions	Probability Distributions
6	Econometrics	Computational Econometrics
7	Environmetrics	Analysis of Ecological and Environmental Data
8	ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
9	Finance	Empirical Finance
10	Genetics	Statistical Genetics
11	Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
12	gR	gRaphical Models in R
13	HighPerformanceComputing	High-Performance and Parallel Computing with R
14	MachineLearning	Machine Learning & Statistical Learning
15	MedicalImaging	Medical Image Analysis
16	Multivariate	Multivariate Statistics
17	NaturalLanguageProcessing	Natural Language Processing
18	OfficialStatistics	Official Statistics & Survey Methodology
19	Optimization	Optimization and Mathematical Programming
20	Pharmacokinetics	Analysis of Pharmacokinetic Data
21	Phylogenetics	Phylogenetics, Especially Comparative Methods
22	Psychometrics	Psychometric Models and Methods
23	ReproducibleResearch	Reproducible Research
24	Robust	Robust Statistical Methods
25	SocialSciences	Statistics for the Social Sciences
26	Spatial	Analysis of Spatial Data
27	Survival	Survival Analysis
28	TimeSeries	Time Series Analysis

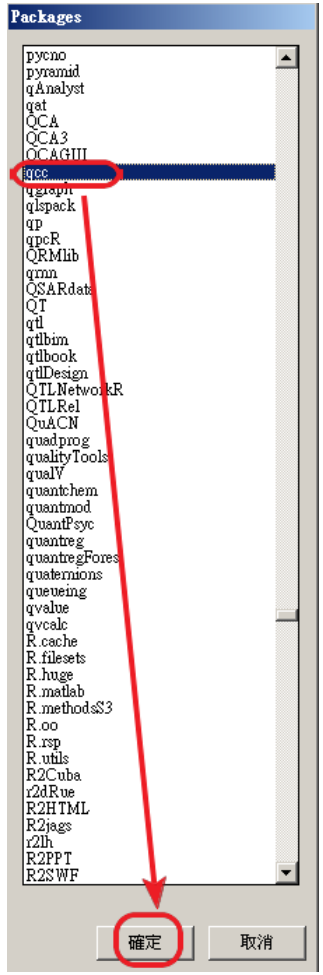
Use packages

- Three steps for use packages:
  1. Download package
  2. Load package
  3. Use package

Step 1. Download package (qcc)

- qcc package
  (Quality Control Charts)






```
> utils:::menuInstallPkgs()
--- Please select a CRAN mirror for use in this session ---
試開URL 'http://cran.csie.ntu.edu.tw/bin/windows/contrib/2.13/qcc_2.0.1.zip'
Content type 'application/zip' length 290101 bytes (283 Kb)
開啓了 URL
downloaded 283 Kb

package 'qcc' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\Documents and Settings\Admin\Local Settings\Temp\RtmpORRVuf\downloaded_packages
> |
```

Step 2. Load package

```
> library(qcc)
Package 'qcc', version 2.0.1
Type 'citation("qcc")' for citing this R package in publications.
> ?qcc
starting httpd help server ... done
>
```

qcc {qcc} R Documentation

Quality Control Charts

Description

Create an object of class 'qcc' to perform statistical quality control. This object may then be used to plot Shewhart charts, drawing OC curves, computes capability indices, and more.

Usage

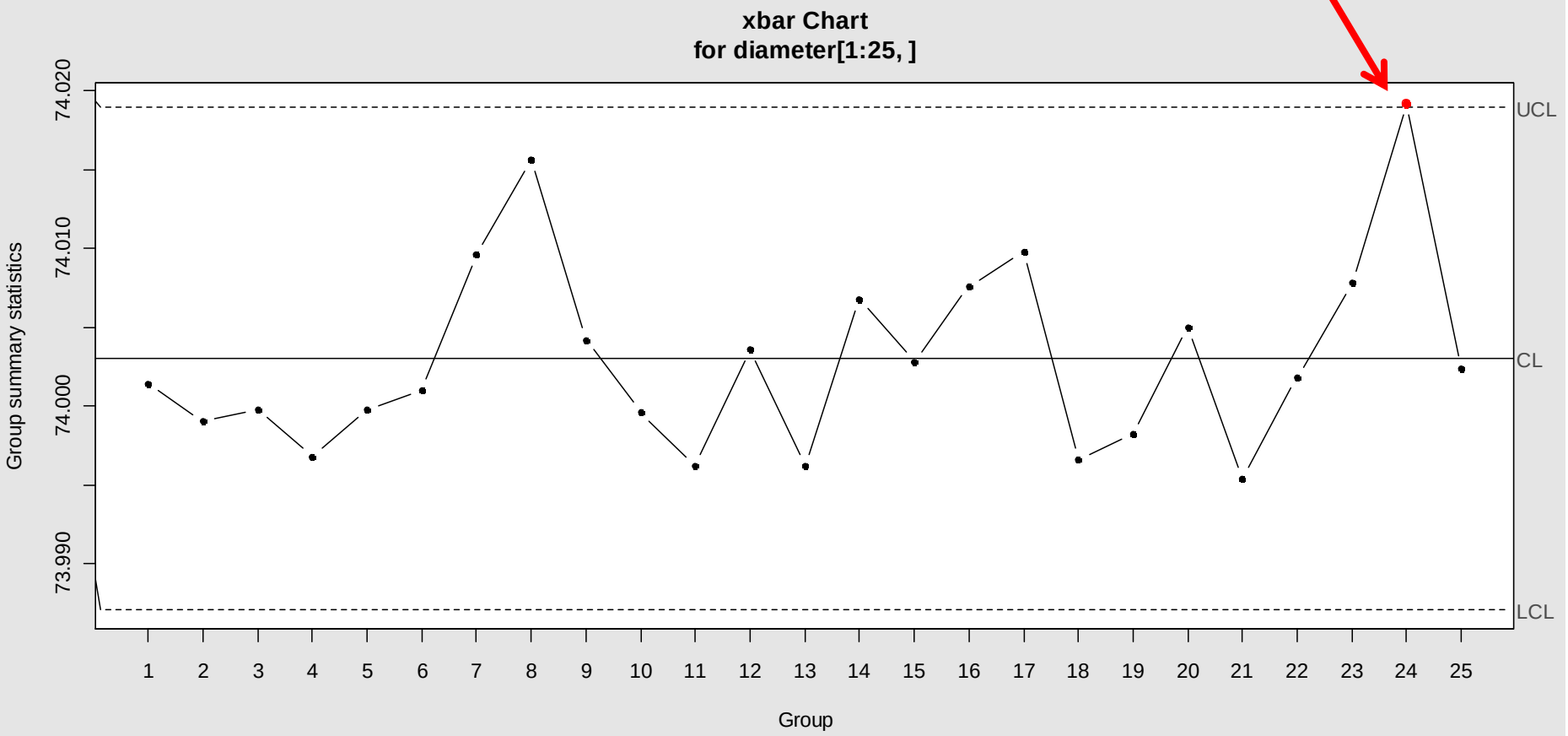
```
qcc(data, type, sizes, center, std.dev, limits,
    data.name, labels, newdata, newsizes, newlabels,
    nsigmas = 3, confidence.level, rules = shewhart.rules,
    plot = TRUE, ...)

## S3 method for class 'qcc'
print(x, ...)
```

Step 3. Use package

```
39 # X-bar quality control chart
40 # load qcc library
41 library(qcc)
42 data(pistonrings)
43 attach(pistonrings)
44 pistonrings
45 diameter <- qcc.groups(diameter, sample)
46 qcc(diameter[1:25,], type="xbar")
47 # end
```

```
# TRY!
head(pistonrings)
head(pistonrings, n=3)
tail(pistonrings)
# end
```

Quality control chart

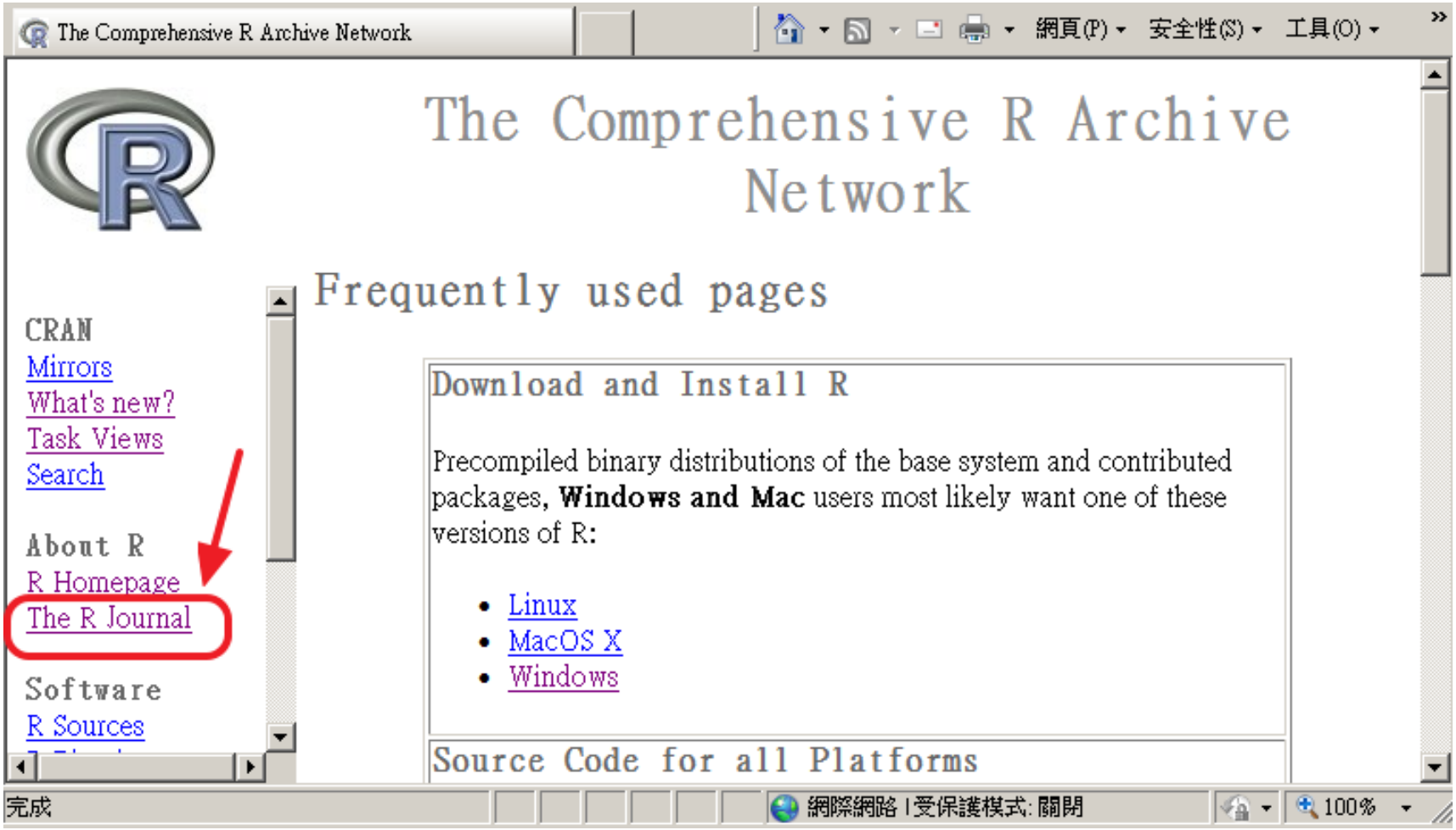


Number of groups = 25
Center = 74.00305
StdDev = 0.01186586

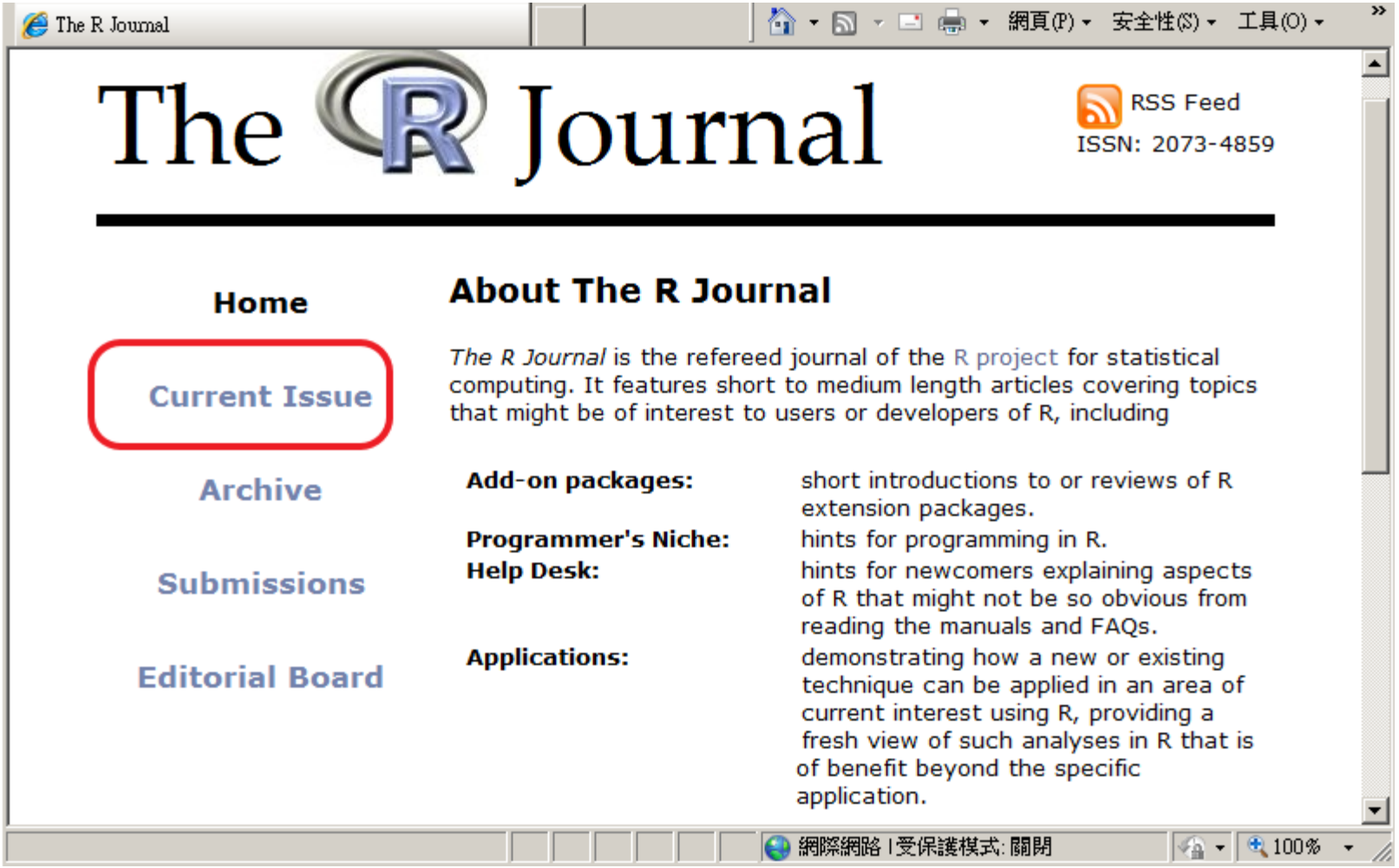
LCL = 73.98713
UCL = 74.01897

Number beyond limits = 1
Number violating runs = 0

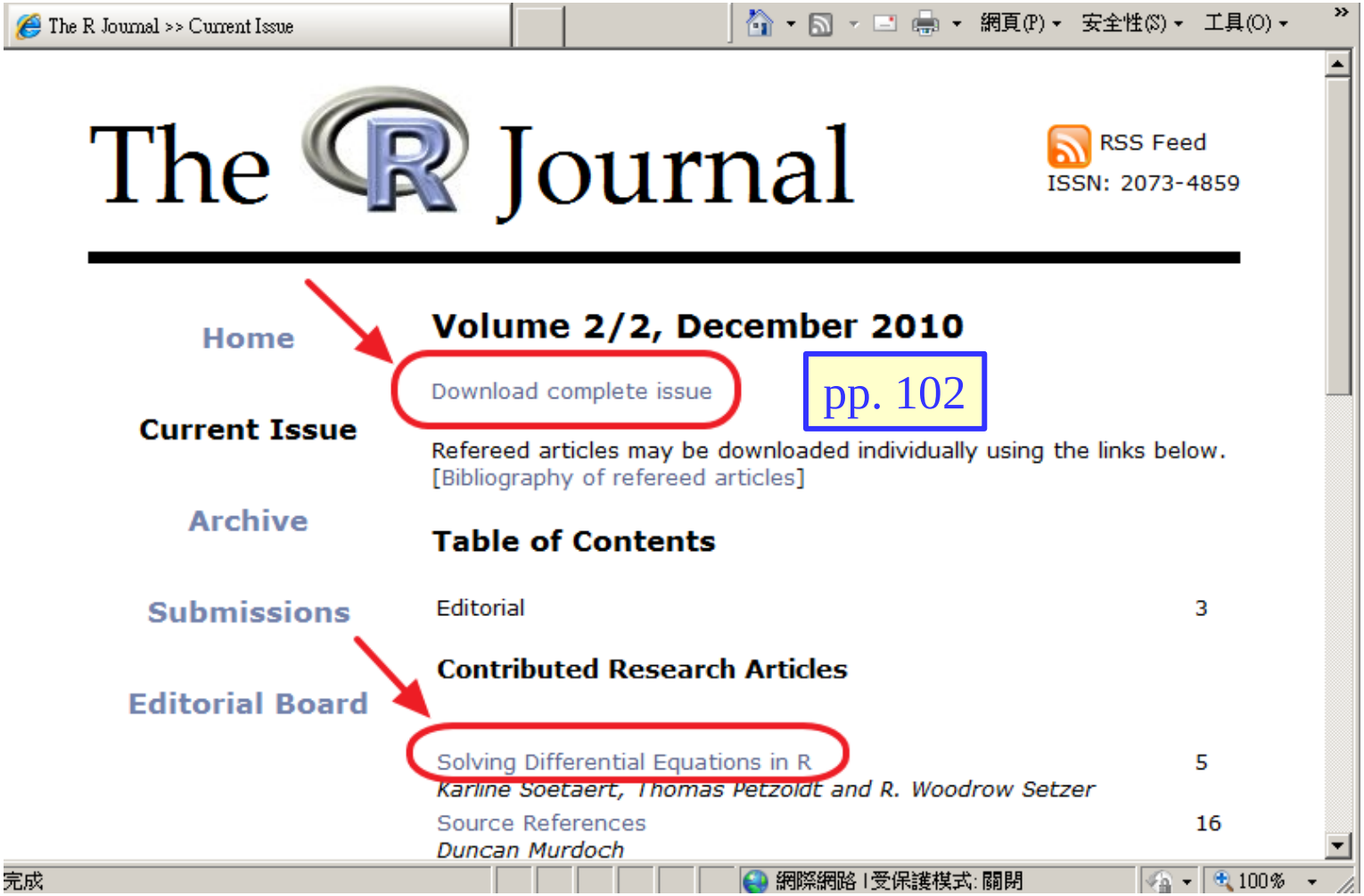
1.4 R Journal


The Comprehensive R Archive Network
網頁(P)
安全性(S)
工具(O)
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
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
完成
網際網路 | 受保護模式: 關閉
100%

1.4 R Journal (cont.)



1.4 R Journal (cont.)



1.4 R Journal (cont.)

Solving Differential Equations in R

by Karline Soetaert, Thomas Petzoldt and R. Woodrow Setzer[1]

Abstract Although R is still predominantly applied for statistical analysis and graphical representation, it is rapidly becoming more suitable for mathematical computing. One of the fields where considerable progress has been made recently is the solution of differential equations. Here we give a brief overview of differential equations that can now be solved by R.

Introduction

Since `odesolve`, much effort has been made to improve R's capabilities to handle differential equations, mostly by incorporating published and well tested numerical codes, such that now a much more complete repertoire of differential equations can be numerically solved.

More specifically, the following types of differential equations can now be handled with add-on packages in R:

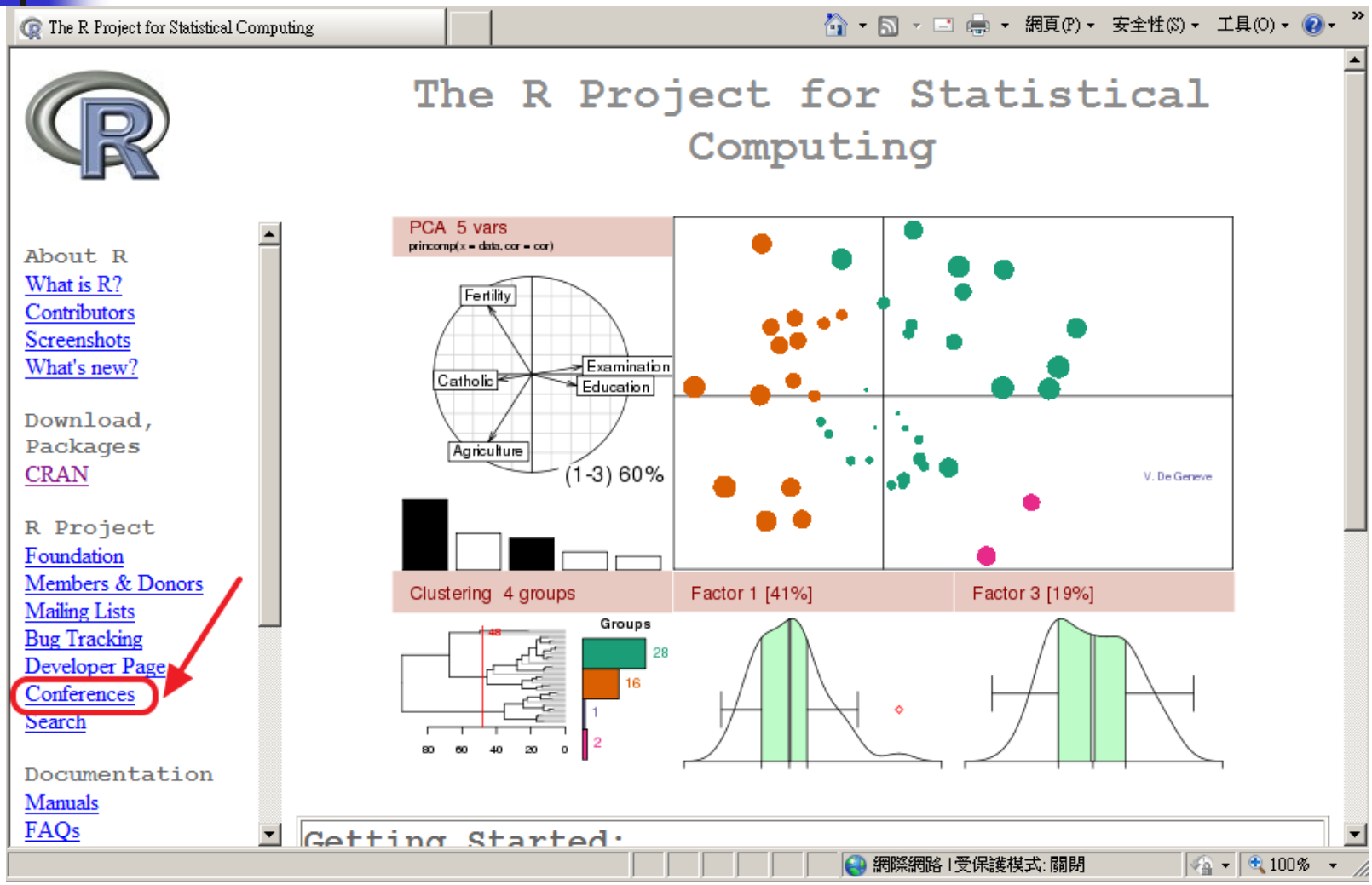
- Initial value problems (IVP) of ordinary differential equations (ODE), using package **deSolve** (Soetaert et al., 2010b).
- Initial value differential algebraic equations (DAE), package **deSolve** .
- Initial value partial differential equations (PDE), packages **deSolve** and **ReacTran**

any, a set of other variables p, often called *parameters*:

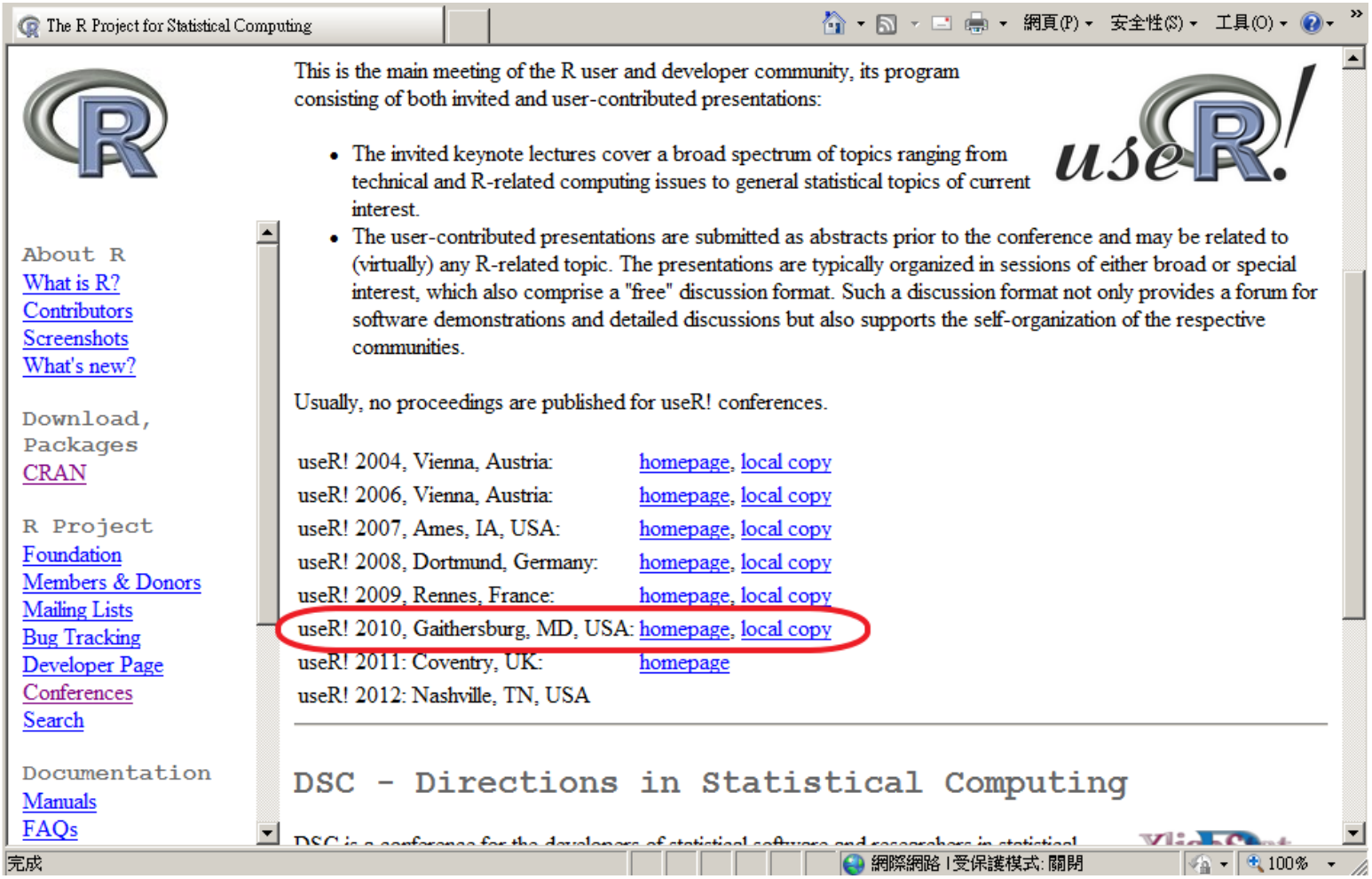
$$y' = \frac{dy}{dt} = f(t, y, p)$$

[1]The views expressed in this paper are those of the authors and do not necessarily reflect the views or policies of the U.S. Environmental Protection Agency

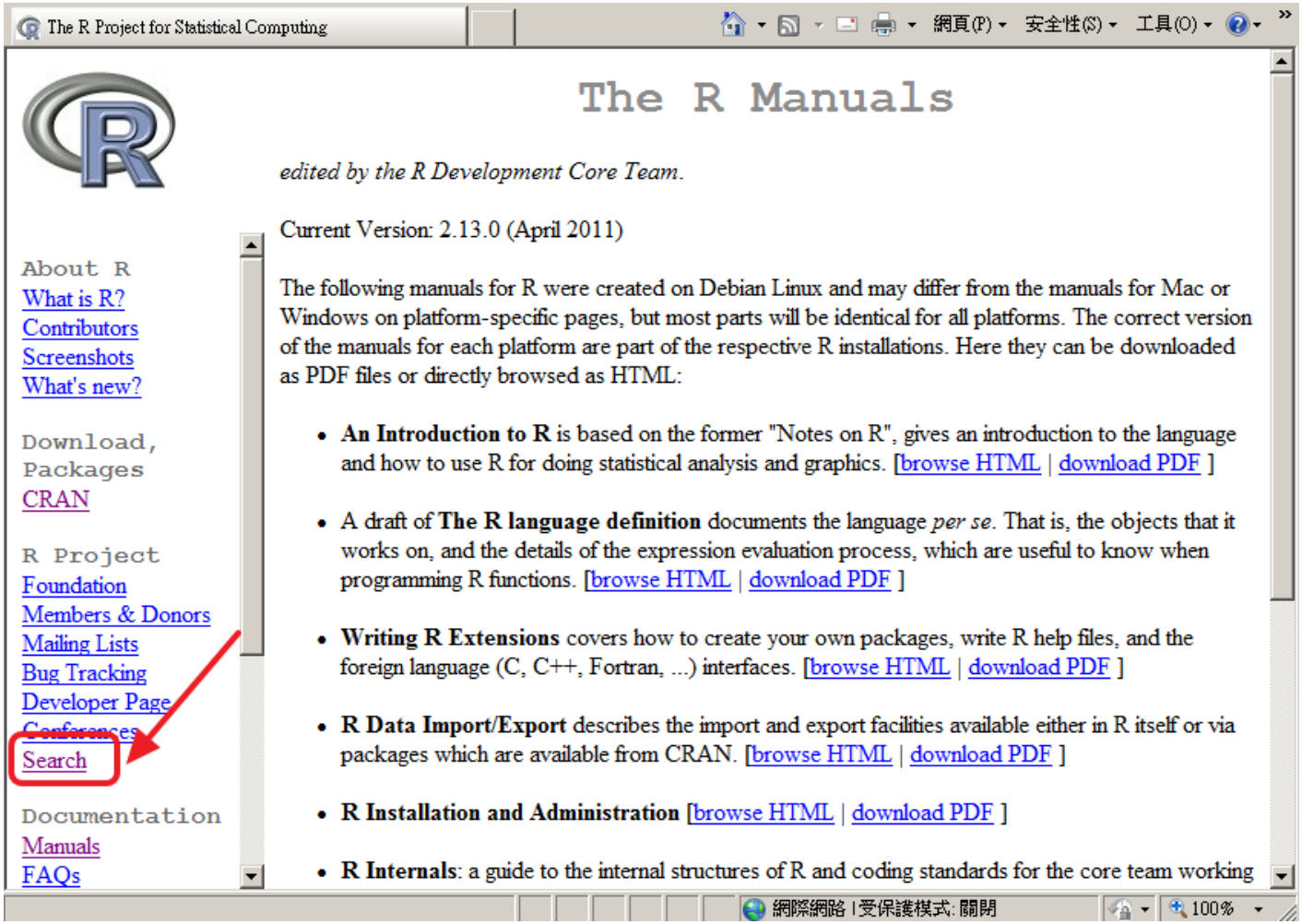
1.5 R conference



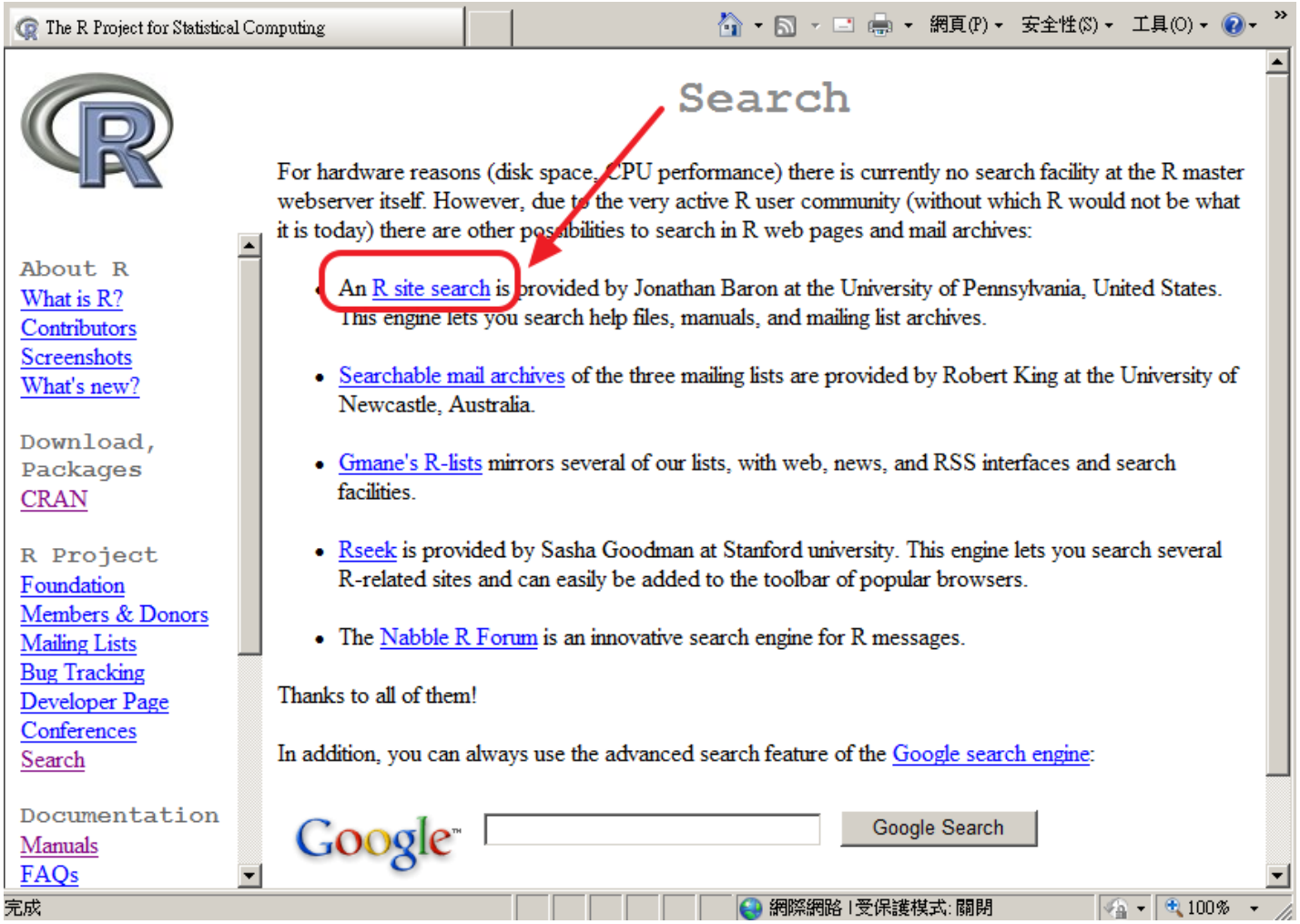
1.5 R conference (cont.)



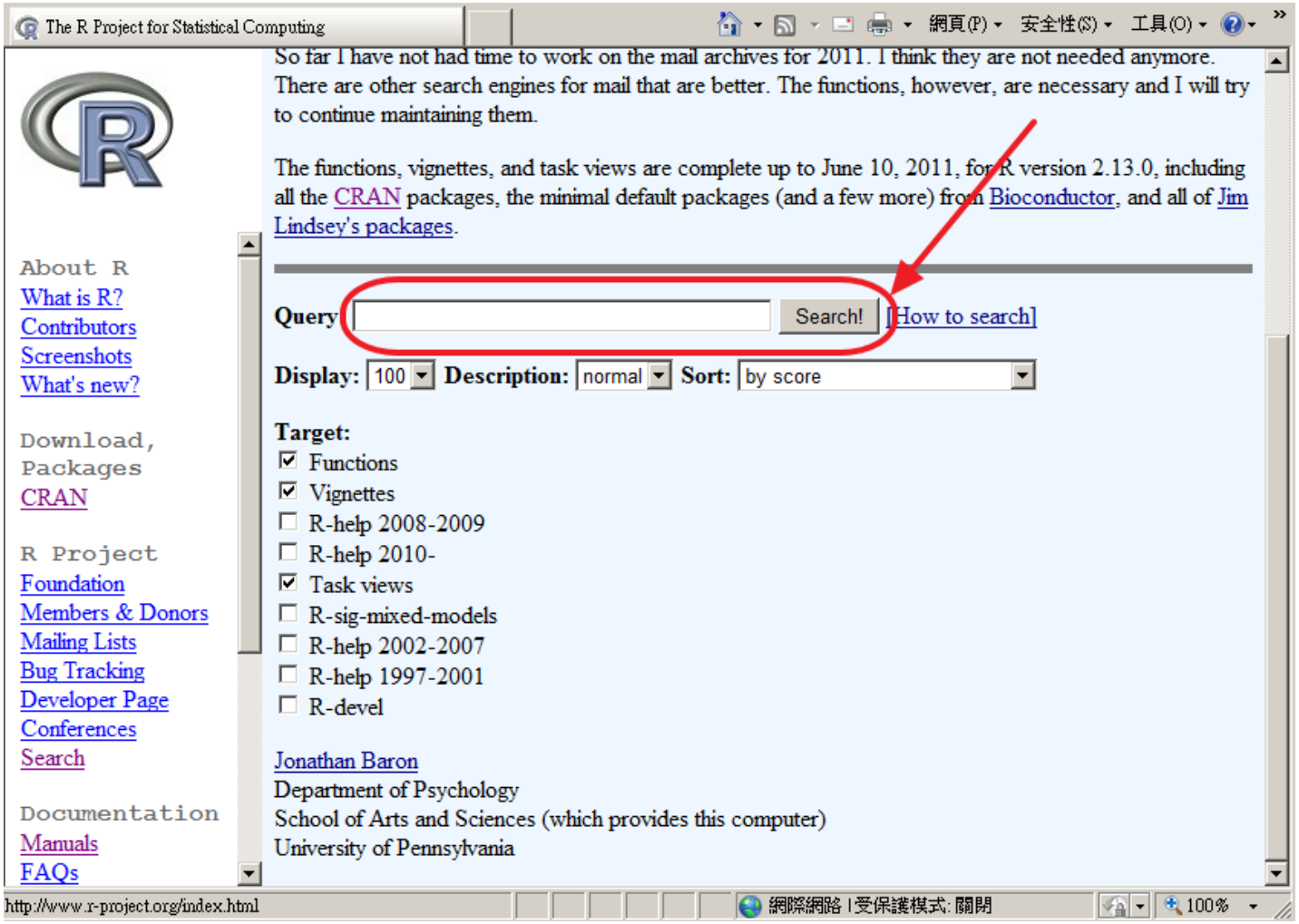
1.6 R Search


The R Project for Statistical Computing
網頁(P)
安全性(S)
工具(O)
The R Manuals
edited by the R Development Core Team.
Current Version: 2.13.0 (April 2011)
The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms. The correct version of the manuals for each platform are part of the respective R installations. Here they can be downloaded as PDF files or directly browsed as HTML:
• An Introduction to R is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics. [browse HTML | download PDF]
• A draft of The R language definition documents the language per se. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions. [browse HTML | download PDF]
• Writing R Extensions covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces. [browse HTML | download PDF]
• R Data Import/Export describes the import and export facilities available either in R itself or via packages which are available from CRAN. [browse HTML | download PDF]
• R Installation and Administration [browse HTML | download PDF]
• R Internals: a guide to the internal structures of R and coding standards for the core team working
About R
What is R?
Contributors
Screenshots
What's new?
Download, Packages
CRAN
R Project
Foundation
Members & Donors
Mailing Lists
Bug Tracking
Developer Page
Conferences
Search
Documentation
Manuals
FAQs
網際網路 | 受保護模式: 關閉
100%

1.6 R Search (cont.)



1.6 R Search (cont.)



2. Preparing Data

2.1 Data Manipulation

2.2 Generating Data

2.3 Creating Objects

2.4 Operators

2.5 Mathematical Functions

2.6 Accessing Data

2.7 Import/Export Data

2.1 Data Manipulation

- Objects have two intrinsic attributes:
  - mode - the basic type of the elements of the object.
    1. Numeric
    2. Character
    3. Logical (FALSE or TRUE).
    4. Complex
  - Length - the number of elements of the object.

```
> # mode and length
> x <- c(1:10)
> mode(x)
[1] "numeric"
> length(x)
[1] 10
> # end
```

Name of object

- Assign operator:
  <-
- The name of an object must start with a letter (A-Z and a-z) and can include letters, digits (0-9), and dots (.).
- R discriminates for the names of the objects the uppercase letters from the lowercase ones.
  (i.e. x and X can name two distinct objects)
- TRY !

```
> # name of object
> A <- "WEPA"; compar <- TRUE; z <- 3+4i
> mode(A); mode(compar); mode(z)
[1] "character"
[1] "logical"
[1] "complex"
> # end
```

Special numbers

- R correctly represents non-finite numeric values:
  - + ∞ (positive infinity): Inf
  - - ∞ (negative infinity): -Inf
- NaN: Not a number
- NA: Used to represent missing values. (NA stands for “not available.”

Example: Inf, -Inf

```
> # Special numbers: Inf, -Inf, NaN, NA
> x <- 5/0
> x
[1] Inf
> exp(x)
[1] Inf
> exp(-x)
[1] 0
> x - x
[1] NaN
> 0/0
[1] NaN
> # NA
> x.NA <- c(1,2,3)
> x.NA
[1] 1 2 3
> length(x.NA) <- 4
> x.NA
[1]  1  2  3 NA
> # end
```

Type of Objects

object	modes	several modes possible in the same object?
vector	numeric, character, complex *or* logical	No
factor	numeric *or* character	No
array	numeric, character, complex *or* logical	No
matrix	numeric, character, complex *or* logical	No
data.frame	numeric, character, complex *or* logical	Yes
ts	numeric, character, complex *or* logical	Yes
list	numeric, character, complex, logical, function, expression, . . .	Yes

PS: *ts* – time series

2.2 Generating Data

- Regular sequences
  - c( )
  - seq( )
  - scan( )
  - rep( )
  - sequence( )
  - gl( )
  - Constants
  - Missing values

Joining (concatenating) vectors: c

- c(…) : Join these numbers together in to a vector.

```
> # Joining (concatenating) vectors: c
> x <- c(2,3,5,2,7,1)
> x
[1] 2 3 5 2 7 1
> y <- c(10,15,12)
> y
[1] 10 15 12
> z <- c(x, y)
> z
[1]  2  3  5  2  7  1 10 15 12
> # end
```

Subsets of Vectors

```
> # Subsets of vectors: Specify the numbers of the elements
> # that are to be extracted
> # Assign to x the values 3, 11, 8, 15, 12
> x <- c(3,11,8,15,12)
>
> # Extract elements no. 2 and 4
> x[c(2,4)]
[1] 11 15
>
> # Use negative numbers to omit elements:
> x[-c(2,4)]
[1]  3  8 12
>
> # Generate a vector of logical (T or F)
> x>10
[1] FALSE  TRUE FALSE  TRUE  TRUE
>
> # Subset for user's defined conditions
> x[x>10]
[1] 11 15 12
```

Subsets of Vectors (cont.)

```
> # Vectors have named elements- method 1
> c(ALAN=100, SERENA=2000, ANDY=300, ALPHA=400)[c("ALAN","ANDY")]
ALAN ANDY
 100  300
>
> # Vectors have named elements- method 2
> score <- c(ALAN=100, SERENA=2000, ANDY=300, ALPHA=400)
> score[c("ALAN","ANDY")]
ALAN ANDY
 100  300
> # end
```

Regular sequences: seq, scan

- Regular sequence of integers : seq(from, to, steps)
- Combine Values into a Vector or List function : c( )
- Using keyboard to input data: scan()

```
> # Regular sequences: seq
> x1 <- 1:100
> x2 <- 100:1
> x3 <- seq(1,10, 0.5)
> x4 <- seq(length=9, from=1, to=5)
> x5 <- c(1,2,2.5,6,10)
> # Regular sequences: scan
> x6 <- scan()
1: 1
2: 2
3: 3
4: 4
5: 5
6:
Read 5 items
>
```

rep, sequence (cont.)

- creates a vector with all its elements identical: rep( )
- creates a series of sequences of integers each ending by the numbers given as arguments: sequence( )

```
> # Regular sequences: rep
> rep(1,5)
[1] 1 1 1 1 1
> # Regular sequences: sequence
> sequence(5)
[1] 1 2 3 4 5
> # Error !!!
> sequence(5,2)
錯誤在sequence(5, 2) : unused argument(s) (2)
> # the concatenated sequences 1:5 and 1:2
> sequence(c(5,2))
[1] 1 2 3 4 5 1 2
> # the concatenated sequences 1:5, 1:2 and 1:3
> sequence(c(5,2,3))
 [1] 1 2 3 4 5 1 2 1 2 3
> # end
```

Generate levels (factors): gl

- gl( ): Generating regular series of factors.
- gl(n, k, length = n*k, labels = 1:n, ordered = FALSE)

  n: An integer giving the number of levels.
  k: An integer giving the number of replications.
  length: An integer giving the length of the result.
  labels: An optional vector of labels for factor levels.
  ordered: The result is ordered or not.

Example: gl

```
> # gl
> gl(3, 5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
> gl(3, 5, length=30)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
> gl(2, 5, label=c("Male", "Female"))
 [1] Male   Male   Male   Male   Male   Female Female Female Female Female
Levels: Male Female
> # end
```

```
# TRY
x <- gl(3, 4, label=c("優良", "普通", "加油"), length=27)
x
# end
```

Constants

- LETTERS
- letters
- month.abb
- month.name
- pi

Example: Constants

```
> # Built-in Constants
> # Upper-case letters
> x <- LETTERS
> x
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S
> length(x)
[1] 26
> # Lower-case letters
> y <- x[-c(2:10)]
> y
 [1] "A" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
> length(y)
[1] 17
> # Three-letter abbreviations for the month names
> monthname.abb <- month.abb
> monthname.abb
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
> monthname.abb[c(1:10)]
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
> # English names for the months
> monthname.full <- month.name
> monthname.full
 [1] "January"   "February"  "March"     "April"     "May"       "June"      "J
 [9] "September" "October"   "November"  "December"
> # Ratio of the circumference of a circle to its diameter
> circle.area <- pi*10^2
> circle.area
[1] 314.1593
> # end
```

Missing values - NA

- 'NA' is a logical constant of length 1 which contains a missing value indicator.

```
> # NA
> x <- c(pi, 1, 2,3)
> x
[1] 3.141593 1.000000 2.000000 3.000000
> x[c(2,4)] <- NA
> x
[1] 3.141593       NA 2.000000       NA
> is.na(x[2])
[1] TRUE
> is.na(x[1])
[1] FALSE
>
> # To replace all NAs by 0
> x[is.na(x)] <- 0
> x
[1] 3.141593 0.000000 2.000000 0.000000
> # end
```

2.3 Creating Objects

- vector
- list
- factor
- array
- matrix
- data.frame
- Time series (ts)

Creating objects: vector

- Vector produces a vector of the given length and mode.
- Coerces all of its arguments into a single type

Example: vector

```
> # A vector of five numbers
> v1 <- c(.29, .30, .15, .89, .12)
> v1
[1] 0.29 0.30 0.15 0.89 0.12
> class(v1)
[1] "numeric"
> typeof(v1)
[1] "double"
> # Coerces into a single type
> v2 <- c(.29, .30, .15, .89, .12, "wepa")
> v2
[1] "0.29" "0.3"  "0.15" "0.89" "0.12" "wepa"
> class(v2)
[1] "character"
> typeof(v2)
[1] "character"
```

Example: vector (cont.)

```
> # vector(mode, length)
> x1 <- vector(mode="numeric", length=1000000)
> # View x1
> head(x1)
[1] 0 0 0 0 0 0
> # Verify a vector
> is.vector(x1)
[1] TRUE
>
> # vector
> x2 <- c("Taiwan", "China", "USA")
> x2
[1] "Taiwan" "China"  "USA"
> is.vector(x2)
[1] TRUE
>
> # Expand the length of a vector
> length(x2) <- 5
> x2
[1] "Taiwan" "China"  "USA"    NA       NA
> # end
```

Creating objects: list

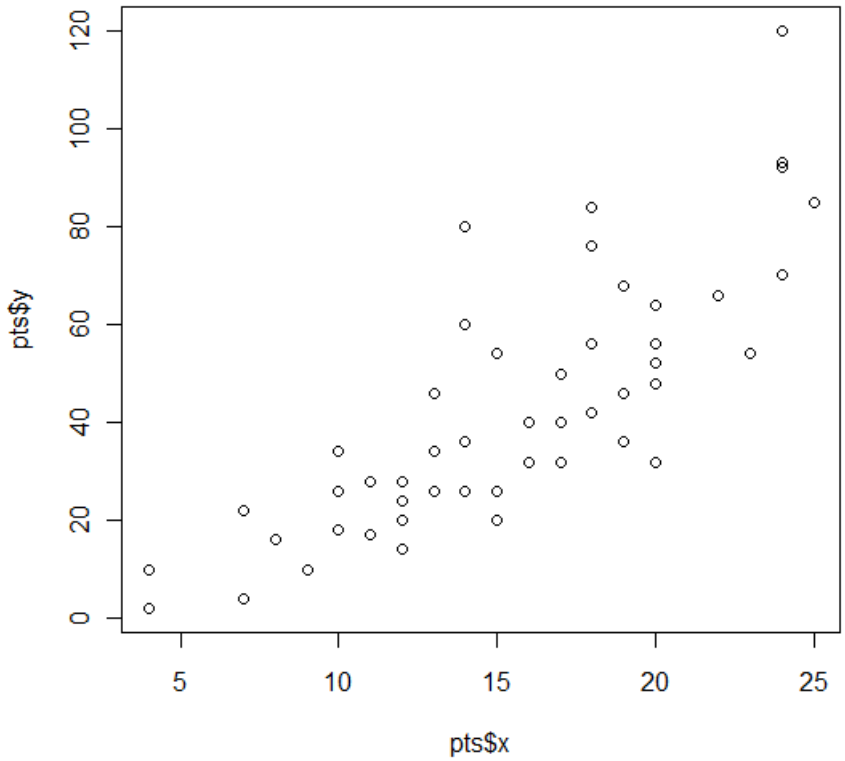
- List is an ordered collection of objects.
- Each element in a list may be given a *name*

Example: list

```
> # list
> list.test1 <- list(1,2,3,4,5)
> list.test1[1]
[[1]]
[1] 1

> list.test1[[1]]
[1] 1
> # Each element in a list may be given a name
> product <- list(destination="Taipei",
+     dimensions=c(3,6,10),price=712.19)
> product[2]
$dimensions
[1]  3  6 10

> product[[2]]
[1]  3  6 10
> product$price
[1] 712.19
> # end
```

Example: list (cont.)

```
> # list all available data sets
> # data()
> head(cars,n=3)
  speed dist
1     4    2
2     4   10
3     7    4
> pts <- list(x=cars[,1], y=cars[,2])
> plot(pts)
> # end
```



Creating objects: factor

- A factor includes not only the values of the corresponding categorical variable, but also the different possible levels of that variable (even if they are present in the data).
- factor (x,
  levels = sort(unique(x), na.last = TRUE),
  labels = levels,
  exclude = NA,
  ordered = is.ordered(x))
  - x: a vector of data, usually taking a small number of distinct values
  - levels specifies the possible levels of the factor (by default the unique values of the vector x),
  - labels defines the names of the levels,
  - exclude the values of x to exclude from the levels,
  - ordered is a logical argument specifying whether the levels of the factor are ordered.

Example: factor

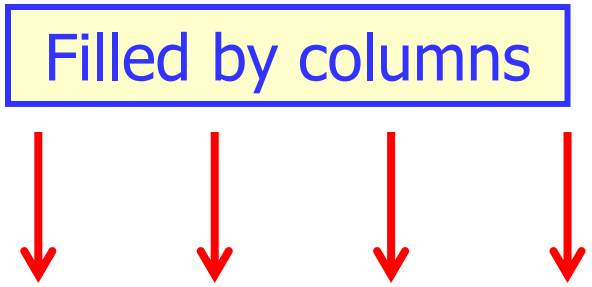
```
> # factor
> f1 <- factor(1:3)
> f2 <- factor(1:3, levels=1:5)
> f3 <- factor(1:3, labels=c("A", "B", "C"))
> f4 <- factor(letters[1:6], label="YDU")
> f4
[1] YDU1 YDU2 YDU3 YDU4 YDU5 YDU6
Levels: YDU1 YDU2 YDU3 YDU4 YDU5 YDU6
> class(f4)
[1] "factor"
> eye.colors <- factor(c("brown", "blue", "blue", "green",
+     "brown", "brown", "brown"))
> eye.colors
[1] brown blue  blue  green brown brown brown
Levels: blue brown green
> levels(eye.colors)
[1] "blue"  "brown" "green"
> # end
```

Creating objects: arrary

- Creates or tests for arrays.
  - array(data = NA, dim = length(data), dimnames = NULL)
  - as.array(x)
  - is.array(x)
- An array is an extension of a vector to more than two dimensions.
- Arrays are used to represent multidimensional data of a single type.

Example: array

```
> # array
> a1 <- array(letters)
> class(a1)
[1] "array"
> dim(a1)
[1] 26
> # array
> a2 <- array(1:3, c(2,4))
> a2
     [,1] [,2] [,3] [,4]
[1,]    1    3    2    1
[2,]    2    1    3    2
> dim(a2)
[1] 2 4
> length(a2)
[1] 8
> a2[1, ] # select row 1
[1] 1 3 2 1
> a2[, 4] # select column 4
[1] 1 2
> # end
```



Example: array (cont.)

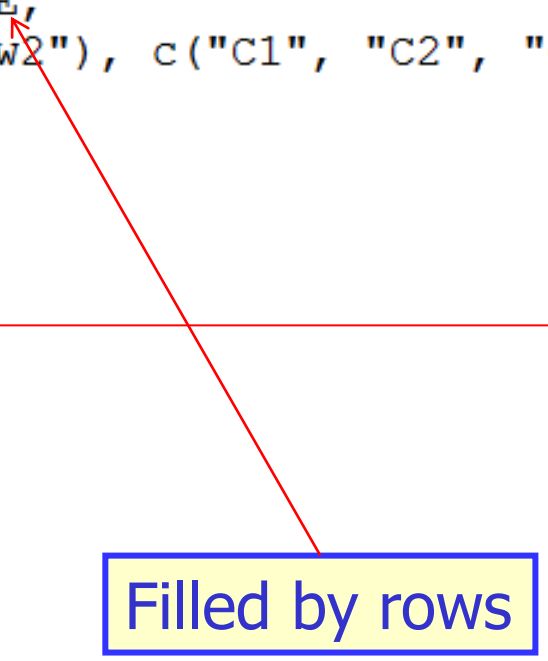
```
> # array
> a3 <- array(data=1:24,dim=c(3,4,2))
> a3
, , 1

     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

     [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24

> # end
```

Creating objects: matrix

- A matrix is an extension of a vector to two dimensions.
- A matrix is used to represent two-dimensional data of a single type.
- matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
- as.matrix(x) #transform data structure into a matrix
- is.matrix(x)

Example: matrix

```
> # matrix
> matrix.data <- matrix(c(1,2,3,4,5,6),
+     nrow = 2, ncol = 3, byrow=TRUE,
+     dimnames = list(c("row1", "row2"), c("C1", "C2", "C3")))
> matrix.data
     C1 C2 C3
row1  1  2  3
row2  4  5  6
> # end
```



Creating objects: data.frame

- A data frame is the type of object normally used in R to store a data matrix.
- A list of variables of the same length, but possibly of different types (numeric, factor, character, logical, . . .).

- data.frame(..., row.names = NULL, check.rows = FALSE, check.names = TRUE)

Example 1: data.frame

```
> # data.frame
> x <- c(1:4); n <- 10; m <- c(10, 35); y <- c(2:4)
> df1 <- data.frame(x, n)
> df1
  x  n
1 1 10
2 2 10
3 3 10
4 4 10
> df2 <-data.frame(x, m)
> df2
  x  m
1 1 10
2 2 35
3 3 10
4 4 35
> # end
```

```
# TRY !
df3 <-data.frame(x, y)
df4 <- data.frame(var1= rnorm(5), var2=LETTERS[1:5])
df4
# end
```

Example 2: data.frame

```
> # The data give the speed of cars and
> # the distances taken to stop.
> data(cars)
> # help(cars)
> class(cars)
[1] "data.frame"
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
> # end
```

TRY ! How to add row names (e.g., Row1, Row2,…)

	speed	dist
Row1	4	2
Row2	4	10
…		

Creating objects: ts

- Time series objects:
  - Many problems look at how a variable changes over time.
- Regression functions for time series (like ar or arima) use time series objects.
- Many plotting functions in R have special methods for time series.

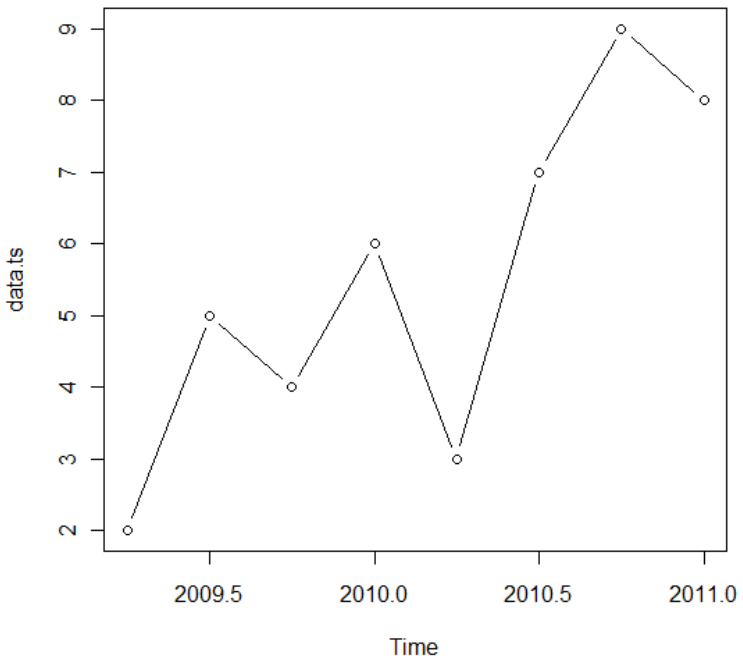
ts function

- ts(data = NA, start = 1, end = numeric(0),
  frequency = 1, deltat = 1,
  ts.eps = getOption("ts.eps"), class = , names =)
- as.ts(x, ...)
- is.ts(x)

- frequency: the number of observations per unit of time.
- deltat: the fraction of the sampling period between successive observations; e.g., 1/12 for monthly data. Only one of frequency or deltat should be provided.
- ts.eps: time series comparison tolerance. Frequencies are considered equal if their absolute difference is less than ts.eps.
- class: class to be given to the result, or none if NULL or "none". The default is "ts" for a single series, c("mts", "ts") for multiple series.

Example: ts

```
> # Time-series
> data.ts <- ts(c(2,5,4,6,3,7,9:8),start=c(2009,2),frequency=4)
> data.ts
     Qtr1 Qtr2 Qtr3 Qtr4
2009         2    5    4
2010    6    3    7    9
2011    8
> is.ts(data.ts)
[1] TRUE
> start(data.ts)
[1] 2009    2
> end(data.ts)
[1] 2011    1
> frequency(data.ts)
[1] 4
> deltat(data.ts) # 0.25(=1/4)
[1] 0.25
> plot(data.ts, type="b")
> # end
```



2.4 Operators

Operators					
Arithmetic		Comparison		Logical	
+	addition	<	lesser than	! x	logical NOT
-	subtraction	>	greater than	x & y	logical AND
*	multiplication	<=	lesser than or equal to	x && y	id.
/	division	>=	greater than or equal to	x | y	logical OR
^	power	==	equal	x || y	id.
%%	modulo	!=	different	xor(x, y)	exclusive OR
%/%	integer division				

PS:The following characters are also operators for R:

- $
- [
- [[
- :
- ?
- <-

2.5 Mathematical Functions

sum(x)	sum of the elements of x
prod(x)	product of the elements of x
max(x)	maximum of the elements of x
min(x)	minimum of the elements of x
which.max(x)	returns the index of the greatest element of x
which.min(x)	returns the index of the smallest element of x
range(x)	id. than c(min(x), max(x))
length(x)	number of elements in x
mean(x)	mean of the elements of x
median(x)	median of the elements of x
var(x) or cov(x)	variance of the elements of x (calculated on *n* -1)
cor(x)	correlation matrix of x if it is a matrix or a data frame (1 if x is a vector)
var(x, y) or cov(x, y)	covariance between x and y, or between the columns of x and those of y if they are matrices or data frames
cor(x, y)	linear correlation between x and y, or correlation matrix if they are matrices or data frames

2.5 Mathematical Functions (cont.)

round(x, n)	rounds the elements of x to n decimals.
ceiling(x)	returns a numeric vector containing the smallest integers not less than x.
floor(x)	returns a numeric vector containing the largest integers not greater than x.
rev(x)	reverses the elements of x.
sort(x)	sorts the elements of x in increasing order. To sort in decreasing order: rev(sort(x)).
rank(x)	ranks of the elements of x
log(x, base)	computes the logarithm of x with base "base"
choose(n, k)	computes the combinations of *k* events among *n* repetitions = *n*! / [(*n*-*k*)!**k*!]
sample(x, size)	resample randomly and without replacement. The option replace = TRUE allows to resample with replacement.

2.6 Accessing Data

```
> # Accessing Data
> x <- c(1:8)
> # how many elements?
> length(x)
[1] 8
> # ith element, i=2
> x[2]
[1] 2
> # all but ith element, i=2
> x[-2]
[1] 1 3 4 5 6 7 8
> # first k elements, k=5
> x[1:5]
[1] 1 2 3 4 5
> # last k elements, k=5
> x[(length(x)-5+1):length(x)]
[1] 4 5 6 7 8
```

2.6 Accessing Data (cont.)

```
> # specific elements
> x[c(1,3,5)]
[1] 1 3 5
> # all greater than some value
> x[x>3]
[1] 4 5 6 7 8
> # biger than or less than some values
> x[x<3 | x>7]
[1] 1 2 8
> # which indices are largest
> which(x==max(x))
[1] 8
> # end
```

2.7 Import/Export Data

Step 1. Set working directory

Step 2. Create text file

Step 3. Read data

Step 4. Merge data

Step 5. Export data

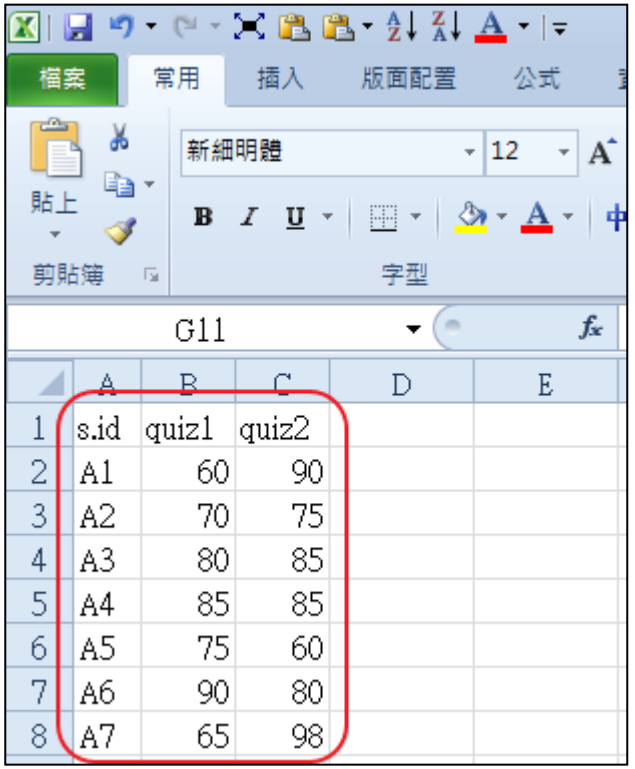
Step 1. Set working directory

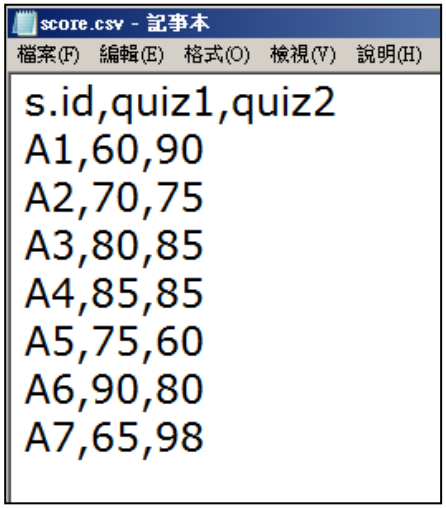
```
> # Import/Export data
> # Create a data directory C:\R.data
> # Get working directory
> getwd()
[1] "C:/Users/Administrator/Documents"
> # Set working directory
> workpath <- "C:/R.data"
> setwd(workpath)
> getwd()
[1] "C:/R.data"
```

Step 2. Create text file

s.id	quiz1	quiz2
A1	60	90
A2	70	75
A3	80	85
A4	85	85
A5	75	60
A6	90	80
A7	65	98

Copy to Excel file





Save as "score.csv" in c:\R.data

Step 3. Read data: read.table

- The read.table function reads a text file into R and returns a data.frame object.
- Each row in the input file is interpreted as an observation.
- Each column in the input file represents a variable.
- The read.table function expects each field to be separated by a delimiter.

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", row.names, col.names,
           as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown")
```

Argument	Description	Default
file	The name of the file to open or, alternatively, the name of a connection containing the data. You can use a URL.	None
header	A logical value indicating whether the first row of the file contains variable names.	FALSE
sep	The character (or characters) separating fields. When "" is specified, any whitespace is used as a separator.	""
quote	the set of quoting characters.	""
dec	The character used for decimal points.	.
row.names	A character vector containing row names for the returned data frame.	None
col.names	A character vector containing column names for the returned data frame.	None
comment.char	read.table can ignore comment lines in input files if the comment lines begin with a single special character	#

Example: read.table

```
> # Create a CSV file (C:\R.data\score.csv)
> # in which each field is separated by commas.
> # Import dataset
> score1 <- read.table(file="score.csv", header= TRUE, sep=",")
> # TRY !
> # score1 <- read.table(file="score.csv", header= TRUE)
> score1
  s.id quiz1 quiz2
1   A1    60    90
2   A2    70    75
3   A3    80    85
4   A4    85    85
5   A5    75    60
6   A6    90    80
7   A7    65    98
> dim(score1)
[1] 7 3
> names(score1)
[1] "s.id"  "quiz1" "quiz2"
> row.names(score1)
[1] "1" "2" "3" "4" "5" "6" "7"
```

Import Data

Function	header	sep	quote	dec	fill	comment.char
read.table	FALSE		\" or \'	.	!blank.lines.skip	#
read.csv	TRUE	,	\"	.	TRUE	
read.csv2	TRUE	;	\"	,	TRUE	
read.delim	TRUE	\t	\"	.	TRUE	
read.delim2	TRUE	\t	\"	,	TRUE	

Read Fixed-width files

```
read.fwf(file, widths, header = FALSE, sep = "\t",
         skip = 0, row.names, col.names, n = -1,
         buffersize = 2000, ...)
```

Functions to read and write data

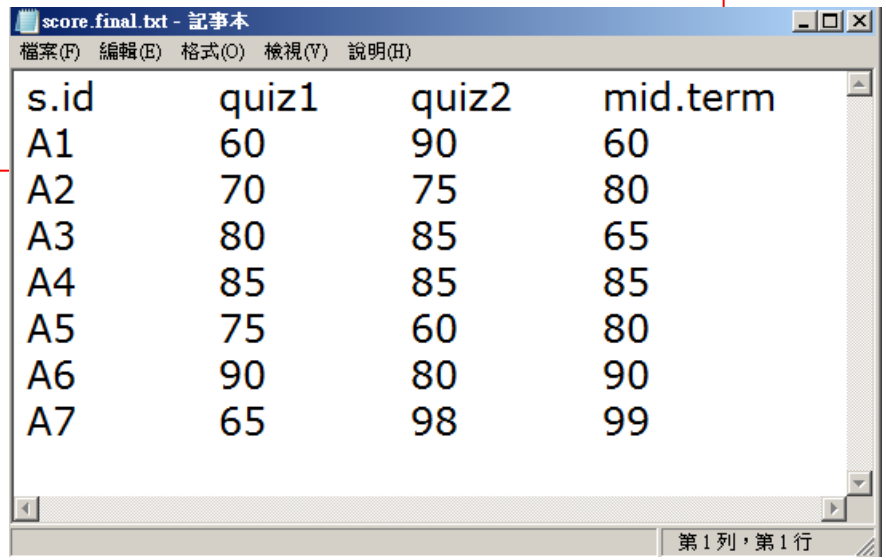
File format	Reading	Writing
ARFF	read.arff	write.arff
DBF	read.dbf	write.dbf
Stata	read.dta	write.dta
Epi Info	read.epiinfo	
Minitab	read.mtp	
Octave	read.octave	
S3 binary files, data.dump files	read.S	
SPSS	read.spss	
SAS Permanent Dataset	read.ssd	
Systat	read.sysstat	
SAS XPORT File	read.xport	

Step 4. Merge data

```
> # Add new column data for mid_term
> mid.term <- matrix(c(60,80,65,85,80,90,99), nrow=7, ncol=1,
+   byrow=FALSE,   dimnames = list(c(),c("mid.term")))
> mid.term
     mid.term
[1,]       60
[2,]       80
[3,]       65
[4,]       85
[5,]       80
[6,]       90
[7,]       99
>
> # Merge two data.frame( score1 and mid_term)
> score2 <- data.frame(score1, mid.term)
> score2
  s.id quiz1 quiz2 mid.term
1   A1    60    90       60
2   A2    70    75       80
3   A3    80    85       65
4   A4    85    85       85
5   A5    75    60       80
6   A6    90    80       90
7   A7    65    98       99
```

Step 5. Export data

```
> # Export dataset
> write.table(score2 , file= "score.final.txt",
+    sep = "\t",
+    append=FALSE,
+    row.names=FALSE,
+    col.names = TRUE,
+    quote= FALSE)
> # end
```



3. Graphics

3.1 Graphical device

3.2 Plot

3.3 Bar chart

3.4 Pie chart

3.5 Box-and-whisker plot

3.6 Stem-and-Leaf plot

3.7 Customized plot

3.8 3D plot

```
> demo(graphics)


        demo(graphics)
        ---- ~~~~~~~~

Type  <Return>   to start :
```

3.1 Graphical device

- The result of a graphical function is sent to a graphical device.
  - Graphical window
  - File
- There are two kinds of graphical functions:
  1. High-level plotting functions which create a new graph.
  2. Low-level plotting functions which add elements to an already existing graph.

Graphical devices

- Open a graphical window:
  x11() or windows()
- List of available graphical devices:
  dev.list()
- Show/Change the active device:
  dev.cur(), dev.set(3)
- Close the active device:
  dev.off(), dev.off(2)

- "null device"' is always device 1.
- cur: current

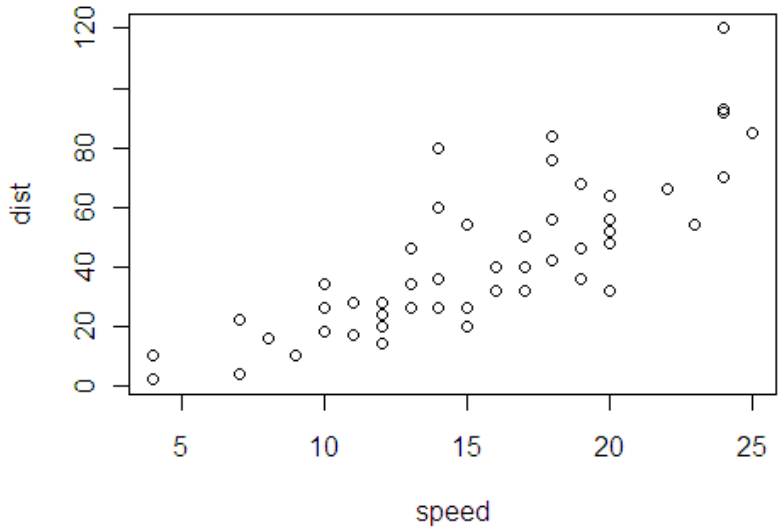
3.2 Plot( )

- plot(x, y) # Same as “plot(y ~ x)”
- type:
  - p :point,
  - l: line,
  - b: both
- pch: controls the type of symbol, either an integer between 1 and 25, or any single character within “ “
- col: controls the colour of symbols. e.g., “red”
- xlab = "string"
- ylab = "string“
- main = "string"
- sub = "string”
- cex: a value controlling the size of texts and symbols with respect to the default
- lwd: a numeric which controls the width of lines

Plot - pch

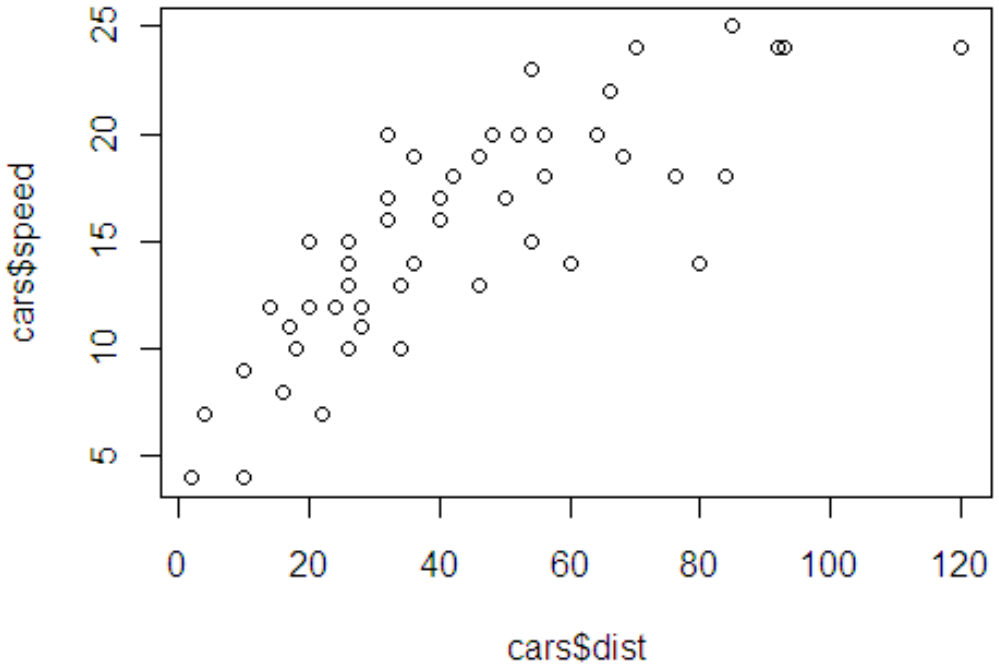
Example: plot

```
> # plot
> data(cars)
> head(cars, n=3)
  speed dist
1     4    2
2     4   10
3     7    4
> plot(cars) # x-axis:speed; y-axis: dist
```



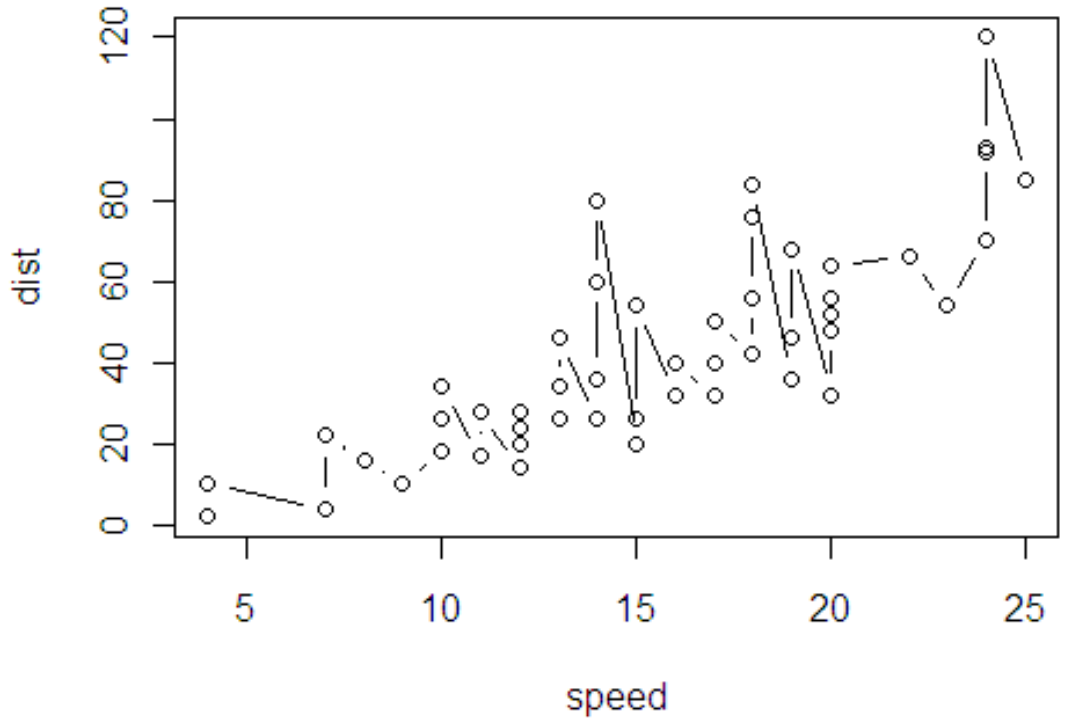
Example: plot (cont.)

```
> # use the variable symbol "$"
> plot(cars$dist, cars$speed)
```



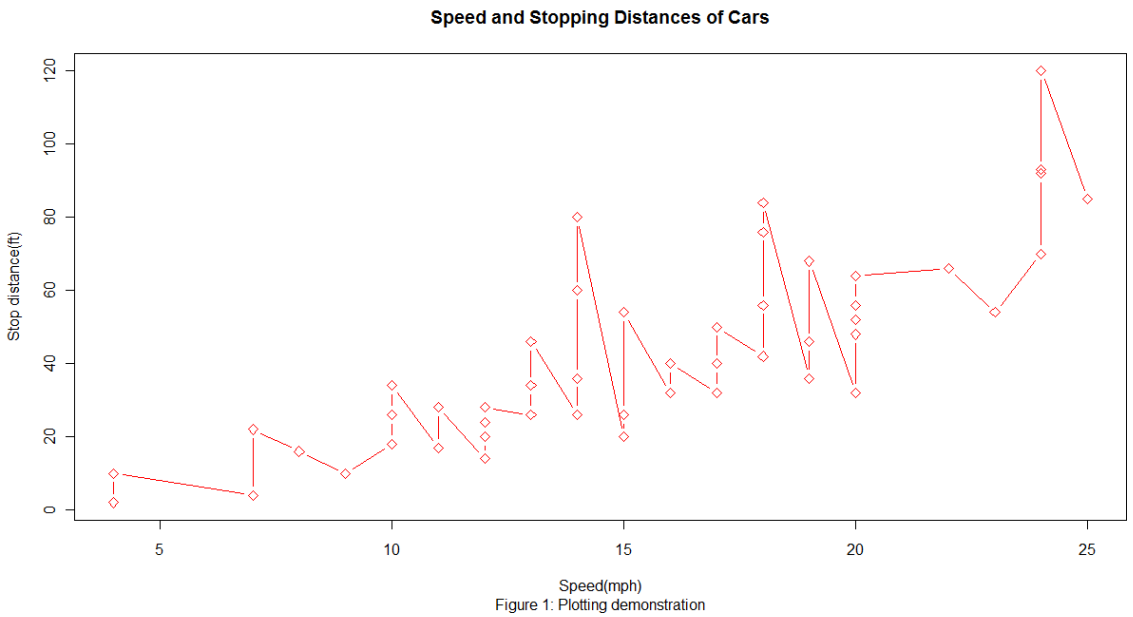
Example: plot (cont.)

```
> plot(cars, type="b")
> # end
```



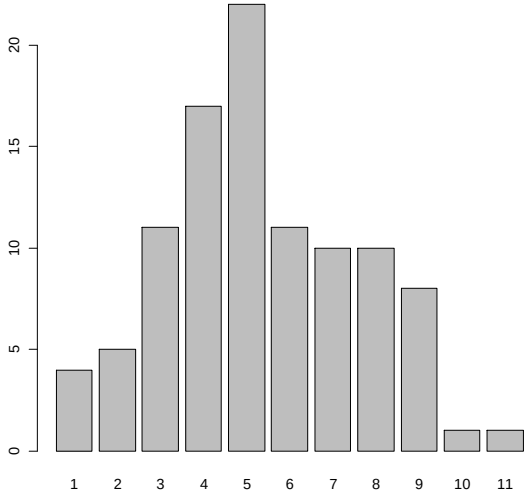
Example: plot (cont.)

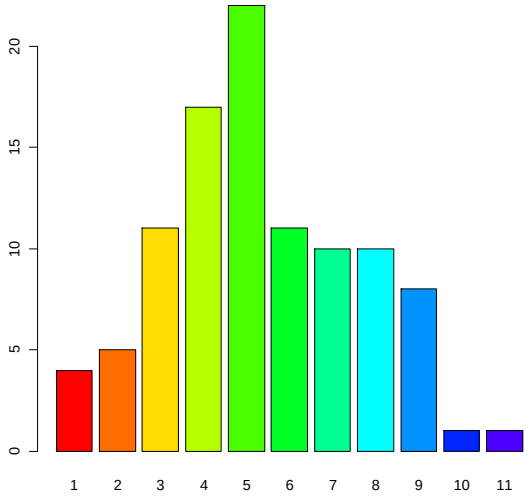
```
> # Available colors (#657)
> # colors()
> plot(cars,type="b",pch=5,col="red",
+   xlab="Speed(mph)", ylab="Stop distance(ft)",
+   main="Speed and Stopping Distances of Cars",
+   sub= "Figure 1: Plotting demonstration")
> # end
```



3.3 Bar chart – barplot

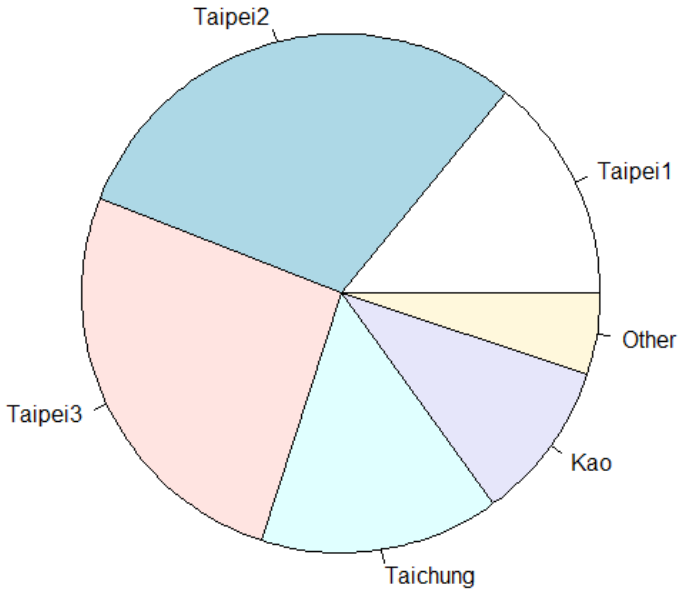
```
> # Barplot
> CarArrived <- table(NumberOfCar <- rpois(100, lambda=5))
> CarArrived

 0  1  2  3  4  5  6  7  8  9 11 12
 1  2  8 15 14 15 17 16  5  3  3  1
> barplot(CarArrived)
> barplot(CarArrived, col=rainbow(14))
> #end
```





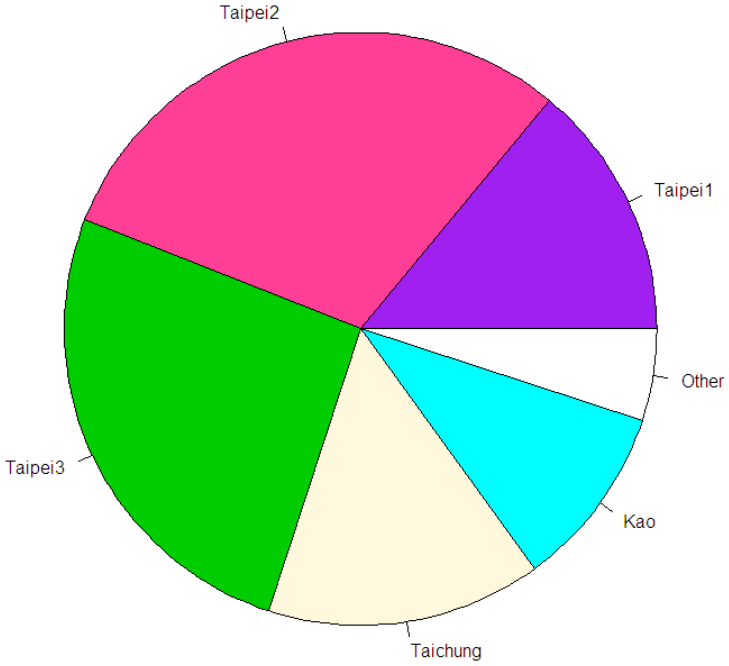
3.4 Pie chart – pie( )

```
> # pie
> # Sales ratio
> pie.sales <- c(0.14, 0.30, 0.26, 0.15, 0.10, 0.05)
> # Sales area
> names(pie.sales) <- c("Taipei1", "Taipei2", "Taipei3", "Taichung",
+   "Kao", "Other")
> # default colours
> pie(pie.sales)
> # end
```



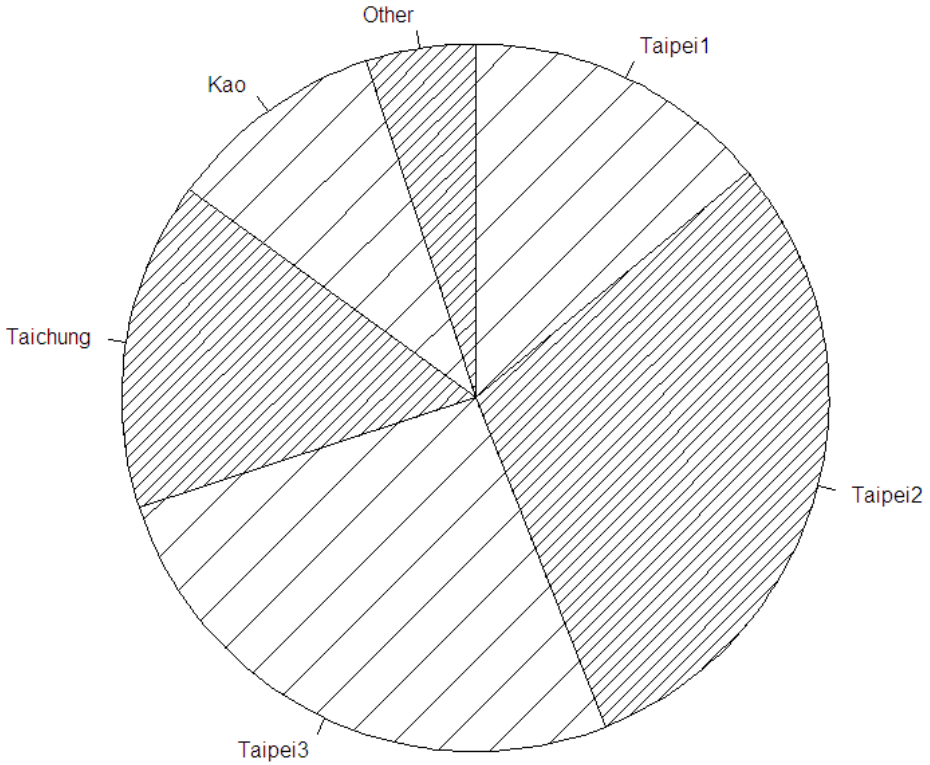
Pie (cont.)

```
> # pie with customized colour
> pie(pie.sales, col = c("purple", "violetred1", "green3",
+   "cornsilk", "cyan", "white"))
> # end
```



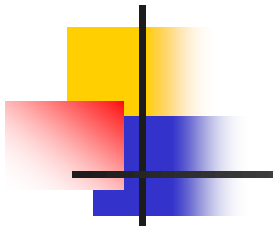
Pie (cont.)

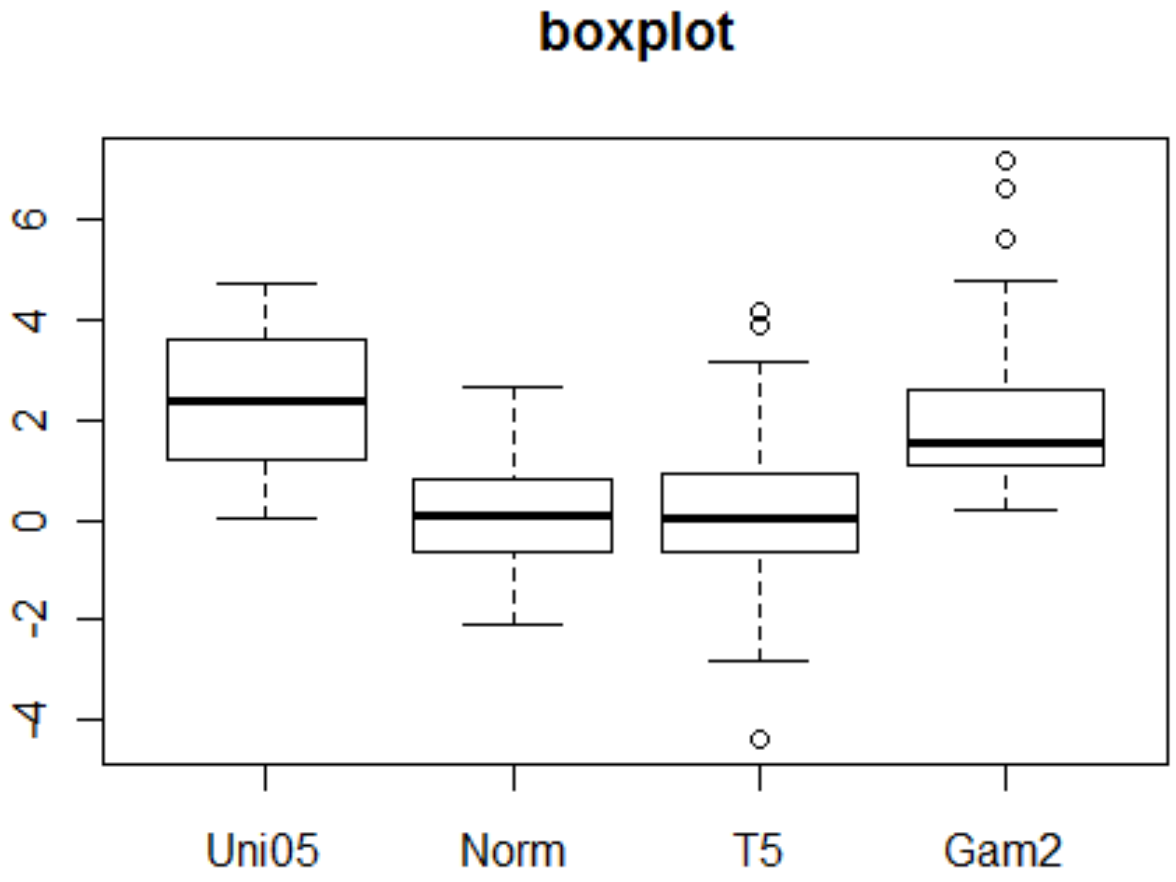
```
> # pie with density of shading lines
> pie(pie.sales, density = c(5,20), clockwise=TRUE)
> # end
```



3.5 Box-and-whisker Plot – boxplot

```
> # p.104
> # Box-and-whisker plot(s) of the given (grouped) values.
> mat <- cbind(Uni05 = (1:100)/21, Norm=rnorm(100), T5 = rt(100,df= 5),
+   Gam2 = rgamma(100, shape = 2))
> head(mat)
          Uni05       Norm          T5      Gam2
[1,] 0.04761905  1.4197446  0.8145416 1.6255930
[2,] 0.09523810  1.3873491  1.5772466 1.3065899
[3,] 0.14285714  0.6634665 -0.3631364 1.3032103
[4,] 0.19047619 -0.3717907 -7.6237581 1.2642297
[5,] 0.23809524  1.2132767 -0.5007267 0.7824783
[6,] 0.28571429  1.1171439 -0.1099461 1.4585379
> boxplot(data.frame(mat), main = "boxplot")
> # end
```

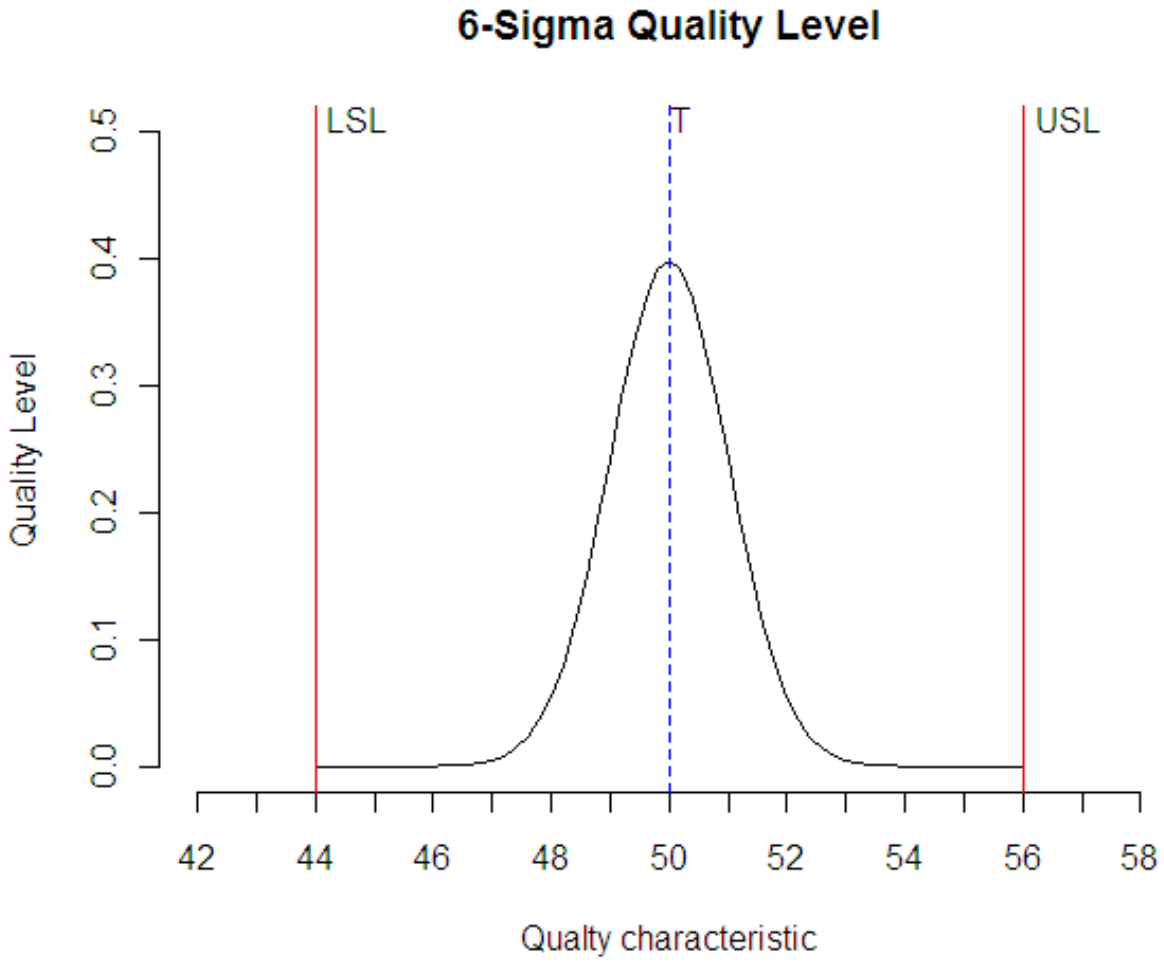



boxplot
6
4
2
0
-2
-4
Uni05
Norm
T5
Gam2

3.6 Stem-and-Leaf Plot – stem

```
> # stem plot
> mat <- round(rnorm(10,30,10),0)
> mat
 [1] 31 32 31 30 37 25  9 34 31  1
> stem(mat)

  The decimal point is 1 digit(s) to the right of the |

  0 | 19
  1 |
  2 | 5
  3 | 0111247

> # end
```

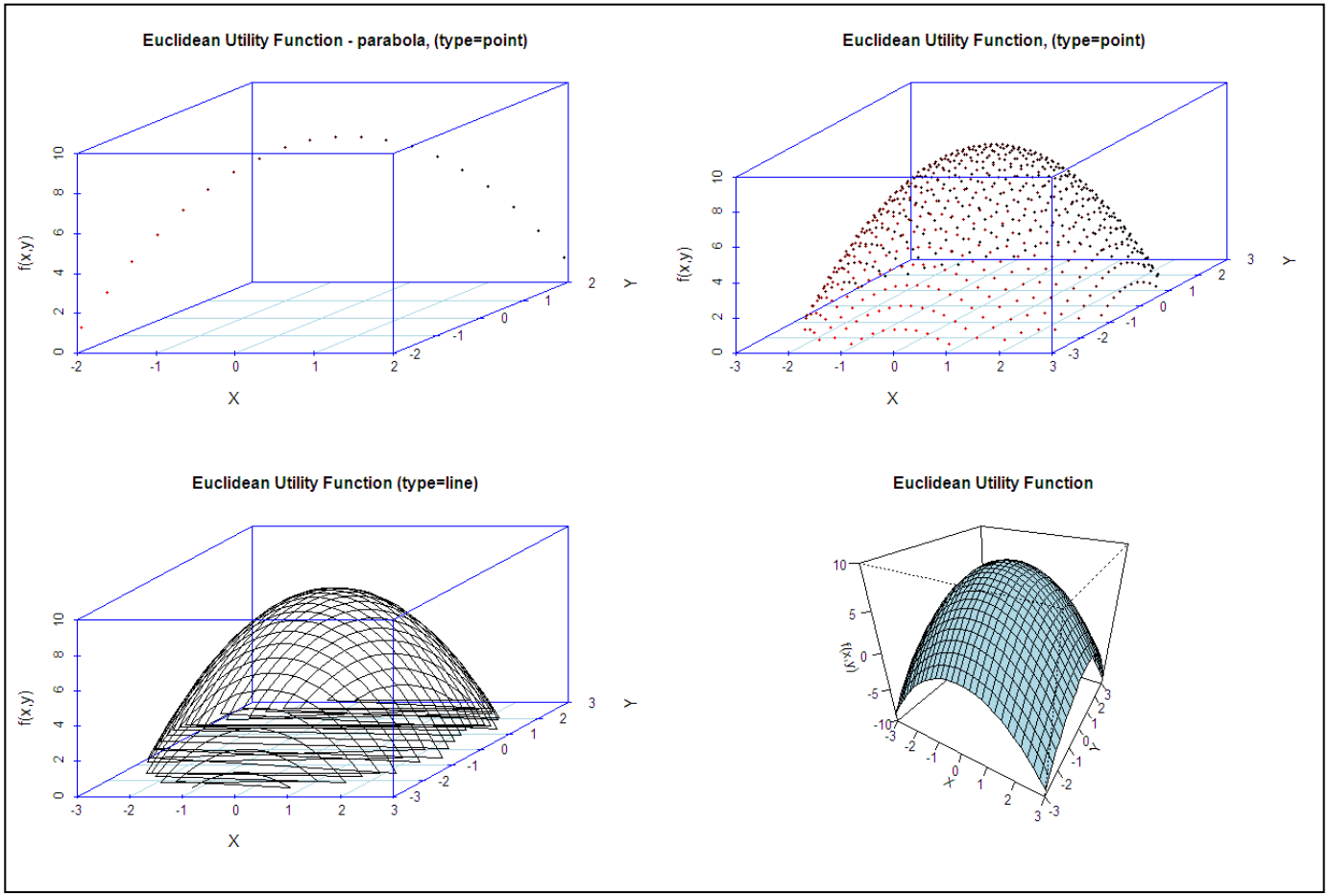
3.7 Customized plot

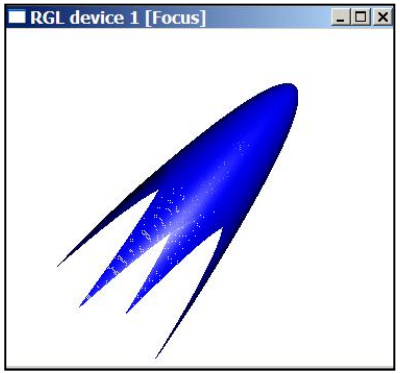


3.7 Customized plot (cont.)

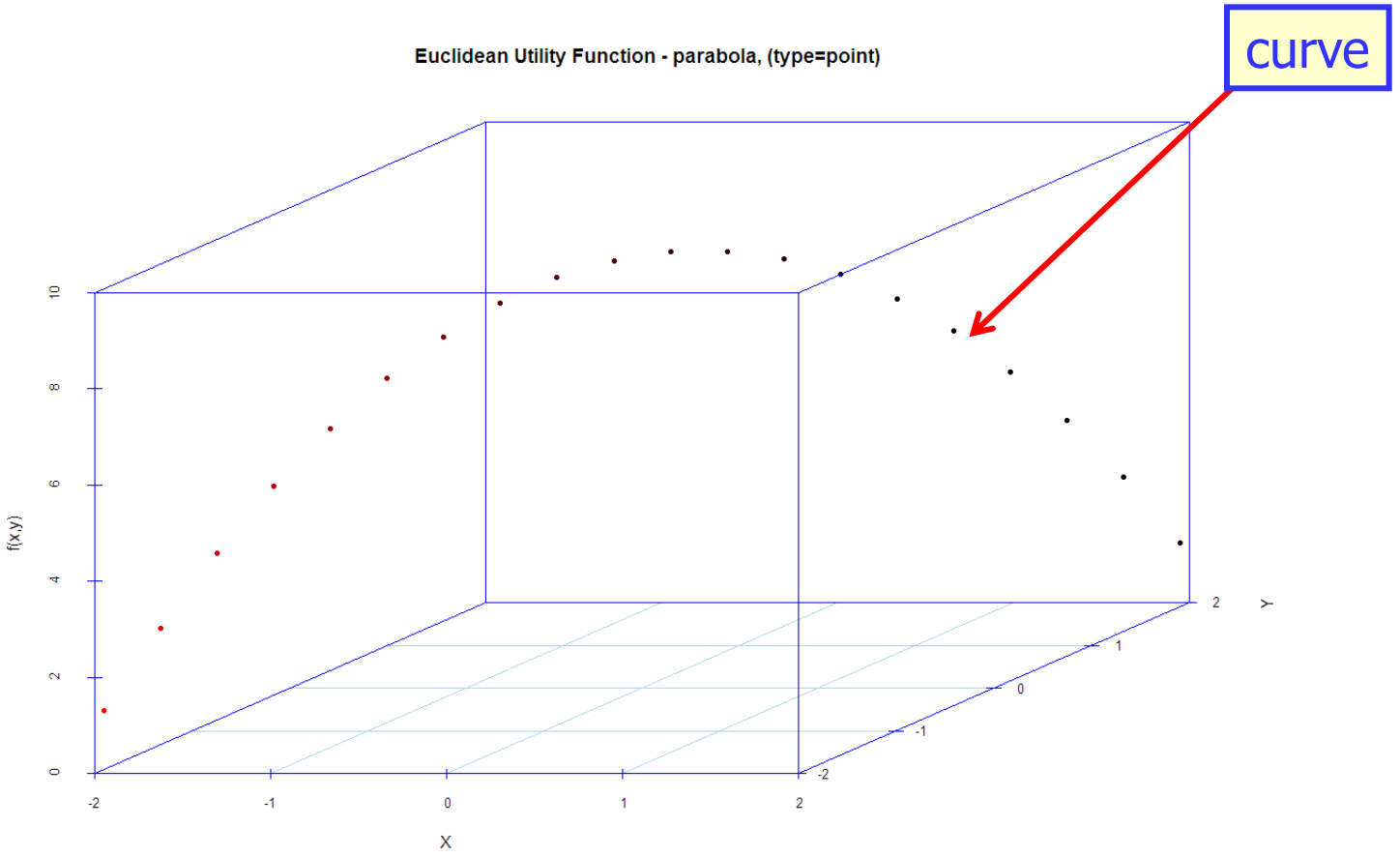
```
> # Customized plot for 6-Sigma Quality Level
> z <- pretty(c(44,56), 50) # Find 50 equally spaced points
> ht <- dnorm(z, mean=50,sd=1) # By default: mean=0, standard deviation=1
> plot(z, ht, type="l", main="6-Sigma Quality Level",
+    xlab="Qualty characteristic", ylab="Quality Level" ,
+    axes=FALSE, xlim=c(42,58), ylim=c(0,0.5))
>
> # Add axis
> # 1=below, 2=left, 3=above and 4=right
> axis(side=1, c(42:58),tick = TRUE)
> axis(side=2, tick = TRUE)
>
> # Add vertical line
> # h=0: horizontal line(s);v=0: vertical line(s)
> abline(v=c(44,50,56), lty=c(1,2,1), col=c("red","blue","red"))
>
> # Add text
> text(44,0.5,"LSL", adj = c(-0.2,0))
> text(50,0.5,"T",adj = c(-0.2,0))
> text(56,0.5,"USL",adj = c(-0.2,0))
> # end
```

3.8 3D plot


Euclidean Utility Function - parabola, (type=point)
Euclidean Utility Function, (type=point)
Euclidean Utility Function (type=line)
Euclidean Utility Function
f(x,y)
X
Y


RGL device 1 [Focus]

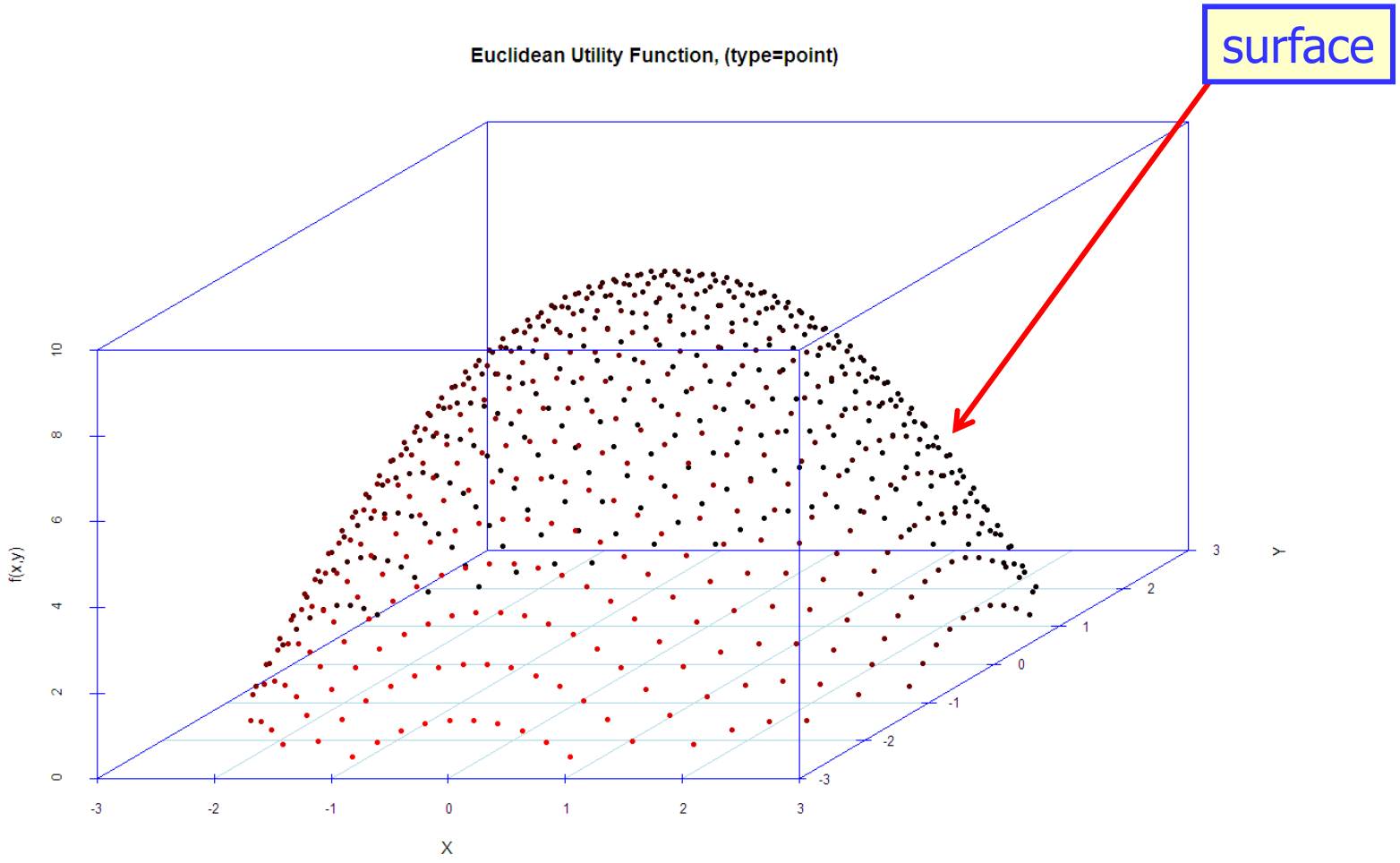
3D plot - method 1



3D plot - method 1 (cont.)

```
> # 3D plot
> # scatterplot3d package
> # setup plotting environment for 2 rows and 2 columns
> # par(bg = "white")
> op <- par(mfrow = c(2,2))
>
> # method 1: scatterplot3d (type=point) - Parabola
> library(scatterplot3d)
> x <- seq(-3, 3, length = 30)
> y <- x
> f <- function (x,y) {a <- 9; a - x^2 - y^2}
> scatterplot3d(x, y, f(x,y),
+  highlight.3d=TRUE, col.axis="blue",
+  pch=20,
+  main="Euclidean Utility Function - parabola, (type=point)",
+  xlab="X", ylab="Y", zlab="f(x,y)",
+  zlim=c(0,9),
+  col.grid="lightblue",
+  type="p"
+ )
```

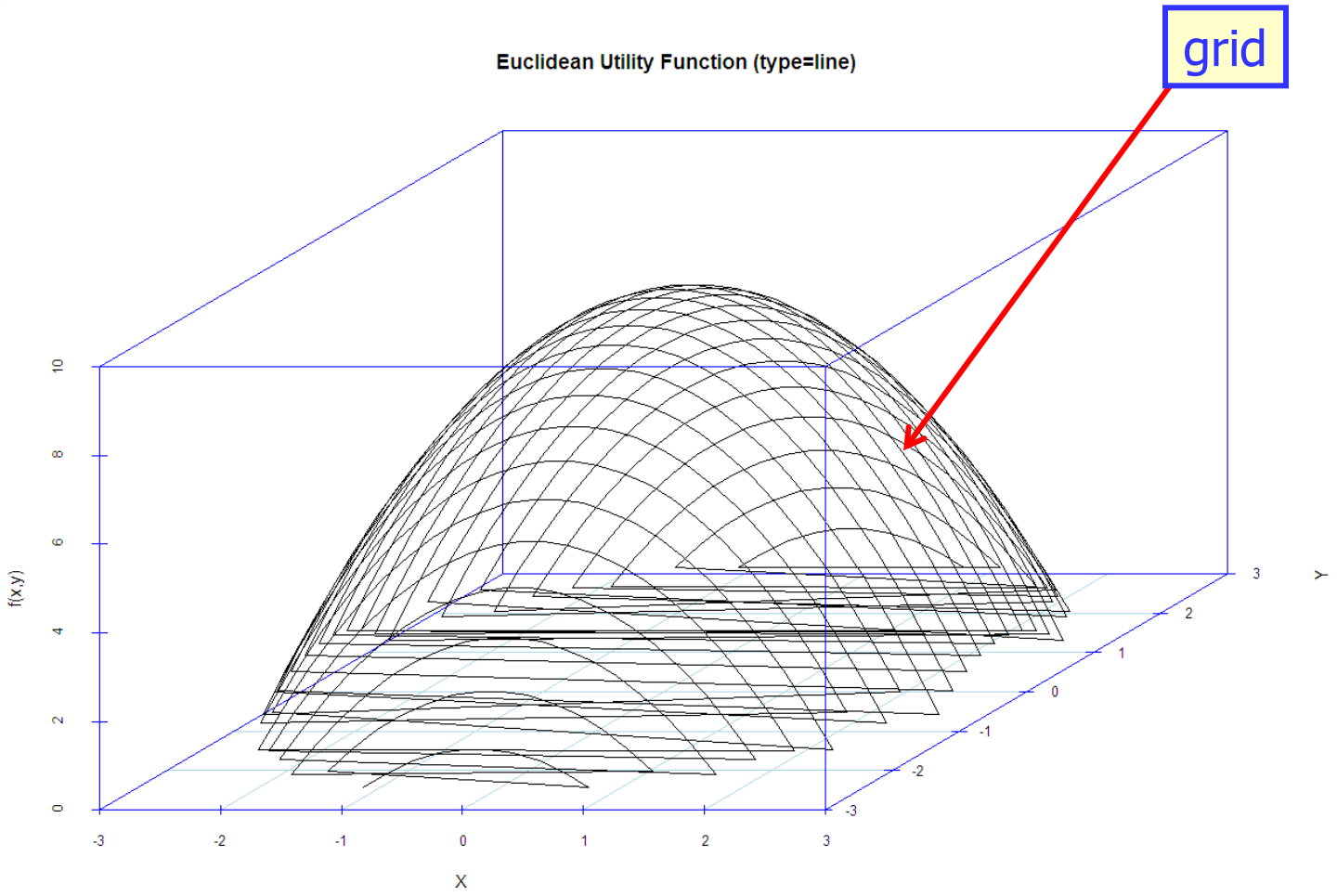
3D plot – method 2



3D plot – method 2 (cont.)

```
> # method 2: scatterplot3d (type=point)
> library(scatterplot3d)
> x <- seq(-3, 3, length = 30)
> f <- function (x,y) {a <- 9; a - x^2 - y^2}
> x1 <- rep(x, 30)
> x2 <- rep(x, each=30)
> znew <- f(x1, x2)
> scatterplot3d(x1, x2, znew,
+  highlight.3d=TRUE, col.axis="blue",
+  pch=20,
+  main="Euclidean Utility Function, (type=point)",
+  xlab="X", ylab="Y", zlab="f(x,y)",
+  zlim=c(0,9),
+  col.grid="lightblue",
+  type="p"
+ )
>
```

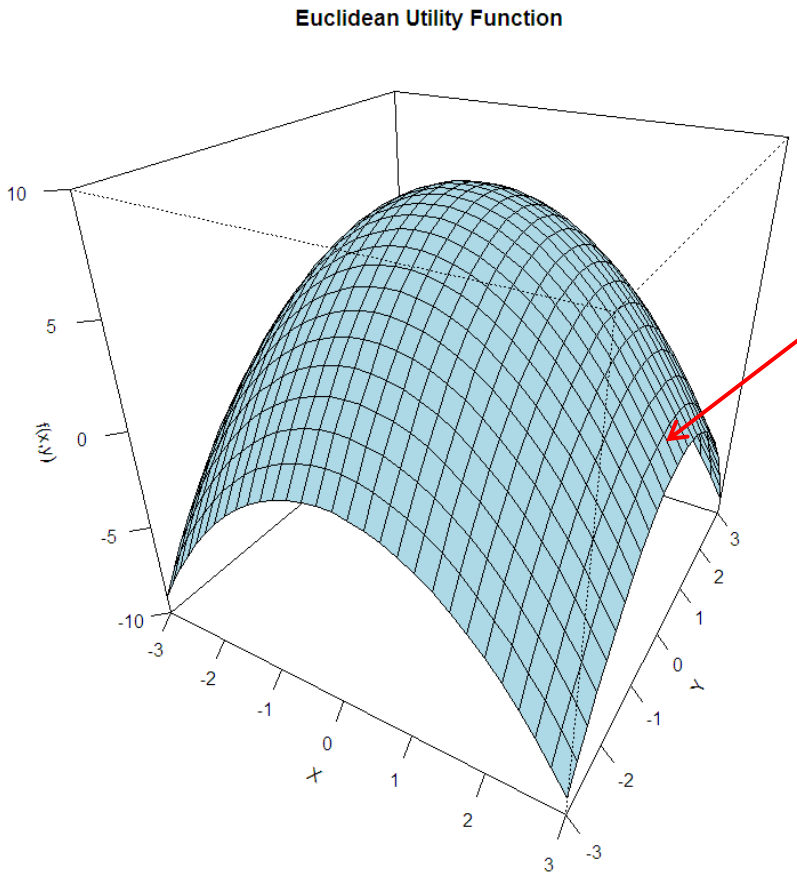
3D plot – method 3



3D plot – method 3 (cont.)

```
> # method 3: scatterplot3d (type=line)
> library(scatterplot3d)
> x <- seq(-3, 3, length = 30)
> f <- function (x,y) {a <- 9; a - x^2 - y^2}
> x1 <- rep(x, 30)
> x2 <- rep(x, each=30)
> znew <- f(x1, x2)
> scatterplot3d(x1, x2, znew,
+  highlight.3d=TRUE, col.axis="blue",
+  pch=20,
+  main="Euclidean Utility Function (type=line)",
+  xlab="X", ylab="Y", zlab="f(x,y)",
+  zlim=c(0,9),
+  col.grid="lightblue",
+  type="l"
+ )
>
```

3D plot – method 4

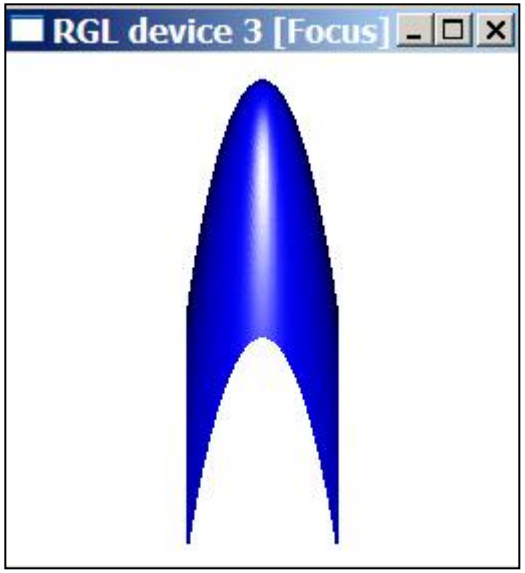


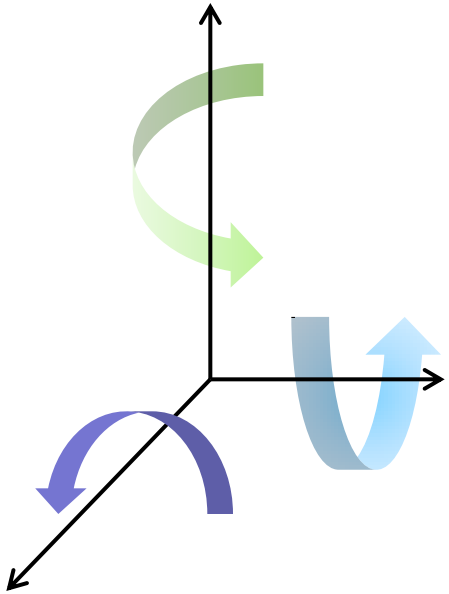
perspective plots of a surface

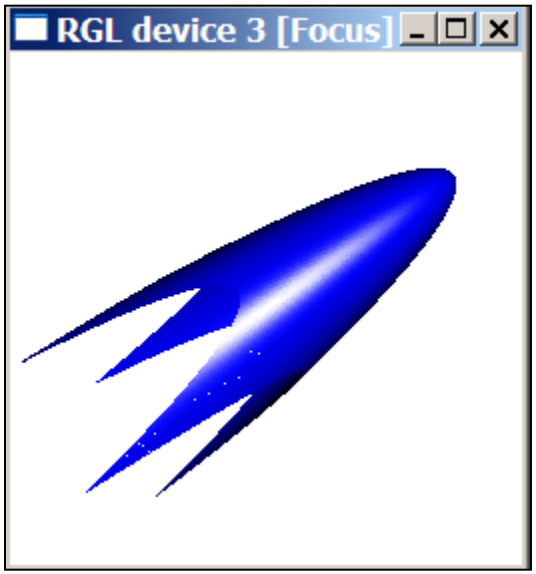
3D plot – method 4 (cont.)

```
> # method 4: persp
> x <- seq(-3,3,length = 30)
> y <- x
> f <- function (x,y) { a <- 9; a-x^2-y^2}
> z <- outer(x,y,f)
> persp(x,y,z,zlim = range(c(-10:10), na.rm = TRUE),
+  expand=1,theta = 30, phi = 30,
+  col = "lightblue",ticktype="detailed",
+  xlab="X", ylab="Y", zlab="f(x,y)",
+  main="Euclidean Utility Function")
```

3D plot – misc3d package



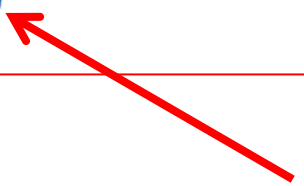




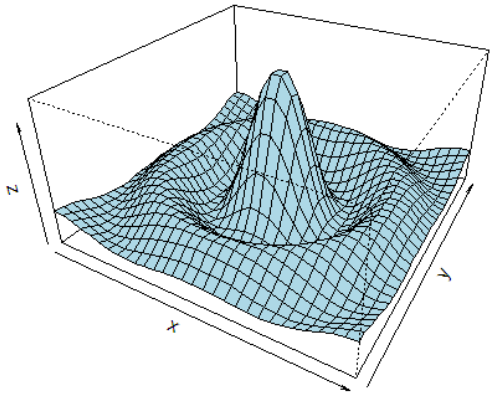
3D plot – misc3d package (cont.)

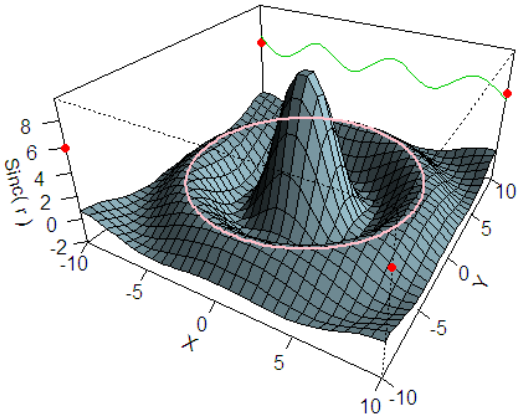
```
> # method 5: misc3d (need "rgl" package)
> # misc3d package
> library(misc3d)
> parametric3d(
+   fx = function(u, v) u,
+   fy = function(u, v) v,
+   fz = function(u, v) -9 - u^2 - v^2 ,
+   fill = FALSE,
+   color = "blue",
+   scale = FALSE,
+   umin = -3, umax = 3, vmin = -3, vmax = 3, n = 100)
>
```

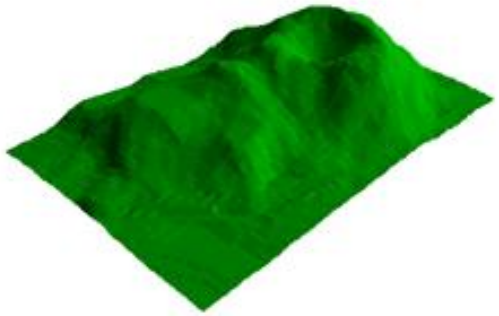
```
> # setup plotting environment to the default
> # par(mfrow=c(1,1))
> par(op)
> # end
```

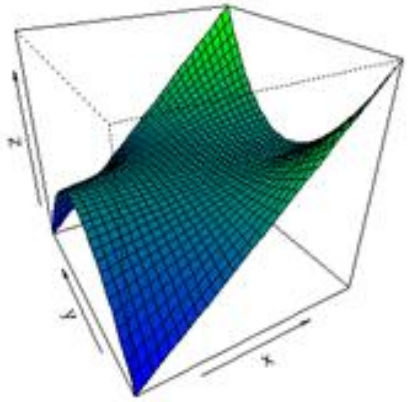


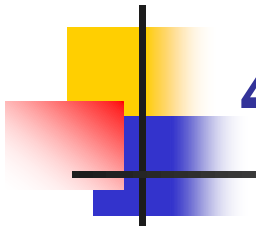
persp package sample











4. Applied Statistics

4.1 Descriptive Statistics

4.2 Hypothesis Test

4.3 Analysis of Variance

4.4 Linear Regression

4.1 Descriptive Statistics

```
> # descriptive statistics
> # Set working directory
> workpath <- "C:/R.data"
> setwd(workpath)
> # import data
> score <- read.table(file="score.csv", header= TRUE, sep=",")
> # view data
> score
  s.id quiz1 quiz2
1   A1    60    90
2   A2    70    75
3   A3    80    85
4   A4    85    85
5   A5    75    60
6   A6    90    80
7   A7    65    98
> # end
```

Descriptive Statistics (cont.)

```
> # summary data
> # TRY summary(score)
> score[2]
  quiz1
1    60
2    70
3    80
4    85
5    75
6    90
7    65
> score[, 2]
[1] 60 70 80 85 75 90 65
> score[3, ]
  s.id quiz1 quiz2
3   A3    80    85
> score$quiz1
[1] 60 70 80 85 75 90 65
> quiz1 <- score$quiz1
> mean(quiz1)
[1] 75
> max(quiz1)
[1] 90
> min(quiz1)
[1] 60
> # end
```

```
> score
  s.id quiz1 quiz2
1   A1    60    90
2   A2    70    75
3   A3    80    85
4   A4    85    85
5   A5    75    60
6   A6    90    80
7   A7    65    98
>
```

Descriptive Statistics (cont.)

```
> # standard deviation
> std(quiz1) # error function
[1] 10.80123
>
> # solution 1
> sqrt( sum( (quiz1 - mean(quiz1))^2 /(length(quiz1)-1)))
[1] 10.80123
>
> # solution 2
> # user's function
> std = function(x) sqrt(var(x))
> std(quiz1)
[1] 10.80123
>
> # solution 3
> sd(quiz1)
[1] 10.80123
> # end
```

4.2 Hypothesis Test

- Testing the Mean of a Sample
- Problem
  - You have a sample from a population.
  - Given this sample, you want to know if the mean of the population could reasonably be a articular value *u*.

t.test

- The output includes a *p*-value. If $p < 0.05$ then the population mean is unlikely to be *u* (reject H_0) whereas $p > 0.05$ provides no such evidence.
- If your sample size *n* is small, then the underlying population must be normally distributed in order to derive meaningful results from the *t* test. A good rule of thumb is that "small" means $n < 30$.

t.test (cont.)

- Analysis

Usage

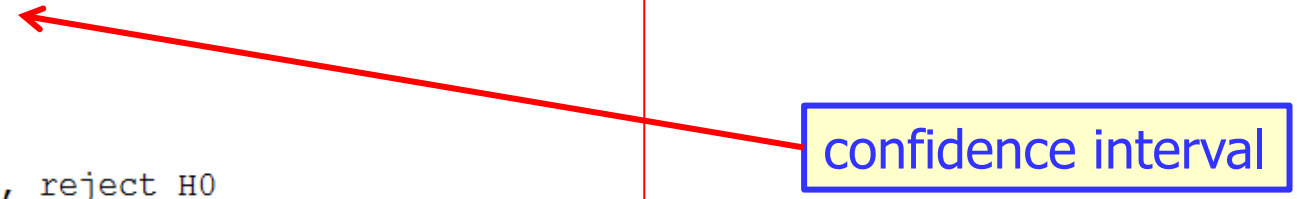
```
t.test(x, ...)

## Default S3 method:
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)

## S3 method for class 'formula'
t.test(formula, data, subset, na.action, ...)
```

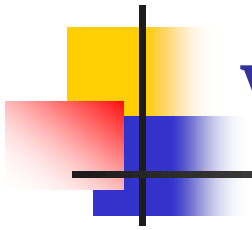
Default is "two.sided"

Sample: t.test

- The following example simulates sampling from a normal population with mean μ = 100. It uses the *t* test to ask if the population mean could be 95, and t.test reports a *p*-value of 0.003087.

```
> # sample- t.test
> # H0: u=95, H1: u<>95
> t.test.data <- rnorm(50, mean=100, sd=15)
> t.test(t.test.data, mu=95)

        One Sample t-test

data:  t.test.data
t = 3.1132, df = 49, p-value = 0.003087
alternative hypothesis: true mean is not equal to 95
95 percent confidence interval:
  97.72046 107.62815
sample estimates:
mean of x
 102.6743

> # p value is small, reject H0
> # end
```





wilcox.test

- Wilcoxon Rank Sum and Signed Rank Tests
- Problem
  - You have a data sample, and you want to know the confidence interval for the median.

Sample: wilcox.test

- Analysis

```
> # wilcox.test(x, conf.int=TRUE)
> wilcox.test.data <- rnorm(50, mean=100, sd=15)
> wilcox.test(wilcox.test.data, conf.int=TRUE, conf.level=0.99)

        Wilcoxon signed rank test with continuity correction

data:  wilcox.test.data
V = 1275, p-value = 7.79e-10
alternative hypothesis: true location is not equal to 0
99 percent confidence interval:
  91.79209 103.47331
sample estimates:
(pseudo)median
      97.29906

> # end
```

prop.test

- Testing a Sample Proportion
- Problem
  - You have a sample of values from a population consisting of successes and failures.
  - You believe the true proportion of successes is *p*, and you want to test that hypothesis using the sample data.

prop.test (cont.)

- Analysis
  - Suppose the sample size is *n* and the sample contains *x* successes:
  - The output includes a *p*-value. Conventionally, a *p*-value of less than 0.05 indicates that the true proportion is unlikely to be *p* whereas a *p*-value exceeding 0.05 fails to provide such evidence.

Usage

x: a vector of counts of successes
n: vector of counts of trials

```
prop.test(x, n, p = NULL,
          alternative = c("two.sided", "less", "greater"),
          conf.level = 0.95, correct = TRUE)
```

Sample: prop.test

- We consider an example where 39 of 215 randomly chosen patients are observed to have asthma and one wants to test the hypothesis that the probability of a "random patient" having asthma is 0.14.

```
> # prop.test
> prop.test(39,215,0.15)

        1-sample proportions test with continuity correction

data:  39 out of 215, null probability 0.15
X-squared = 1.425, df = 1, p-value = 0.2326
alternative hypothesis: true p is not equal to 0.15
95 percent confidence interval:
 0.1335937 0.2408799
sample estimates:
        p
0.1813953

> # end
```

shapiro.test

- Shapiro-Wilk Normality Test
- Problem
  - You want a statistical test to determine whether your data sample is from a normally distributed population.

shapiro.test (cont.)

- Analysis
  - Conventionally, $p < 0.05$ indicates that the population is likely not normally distributed whereas p > 0.05 provides no such evidence.

Usage

```
shapiro.test(x)
```

Sample: shapiro.test

```
> # shapiro.test
> shapiro.test(rnorm(100, mean = 5, sd = 3))

        Shapiro-Wilk normality test

data:  rnorm(100, mean = 5, sd = 3)
W = 0.9882, p-value = 0.5202

> shapiro.test(runif(100, min = 2, max = 4))

        Shapiro-Wilk normality test

data:  runif(100, min = 2, max = 4)
W = 0.9474, p-value = 0.0005579

> # end
```

Testing for Normality

- nortest package
  1. Anderson–Darling test (**ad.test**)
  2. Cramer–von Mises test (**cvm.test**)
  3. Lilliefors test (**lillie.test**)
  4. Pearson chi-squared test for the composite hypothesis of normality (**pearson.test**)
  5. Shapiro–Francia test (**sf.test**)
- Histogram plot
- Quantile-quantile plot

Quantile-Quantile Plot

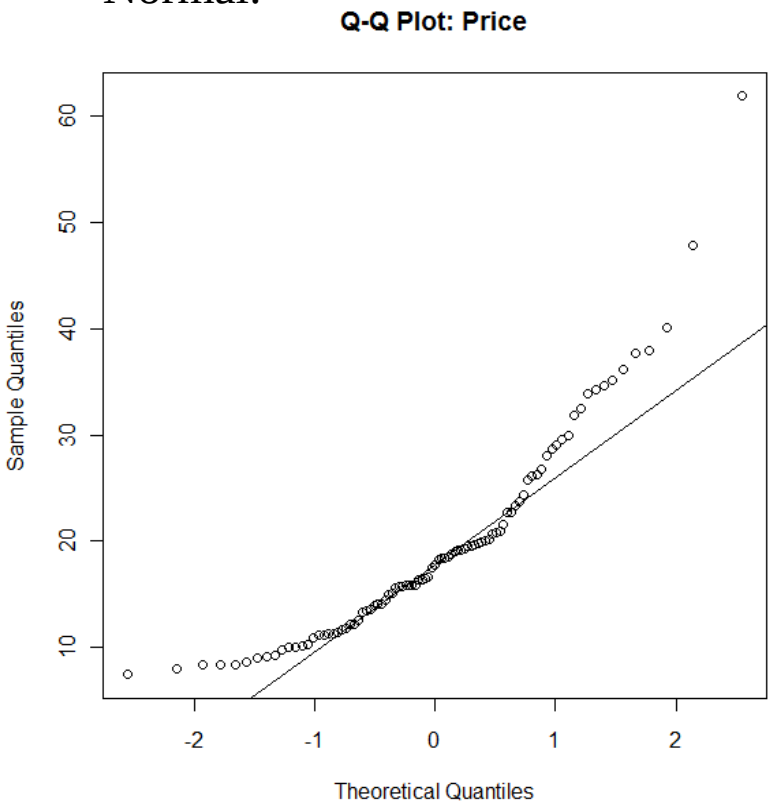
- You want to know whether the data is normally distributed.
- If the data had a perfect normal distribution, then the points would fall exactly on the diagonal line.
- Many points are close, especially in the middle section, but the points in the tails are pretty far off.
- Too many points are above the line, indicating a general skew to the left.
- The leftward skew might be cured by a logarithmic transformation.
- Also see qqplot function.

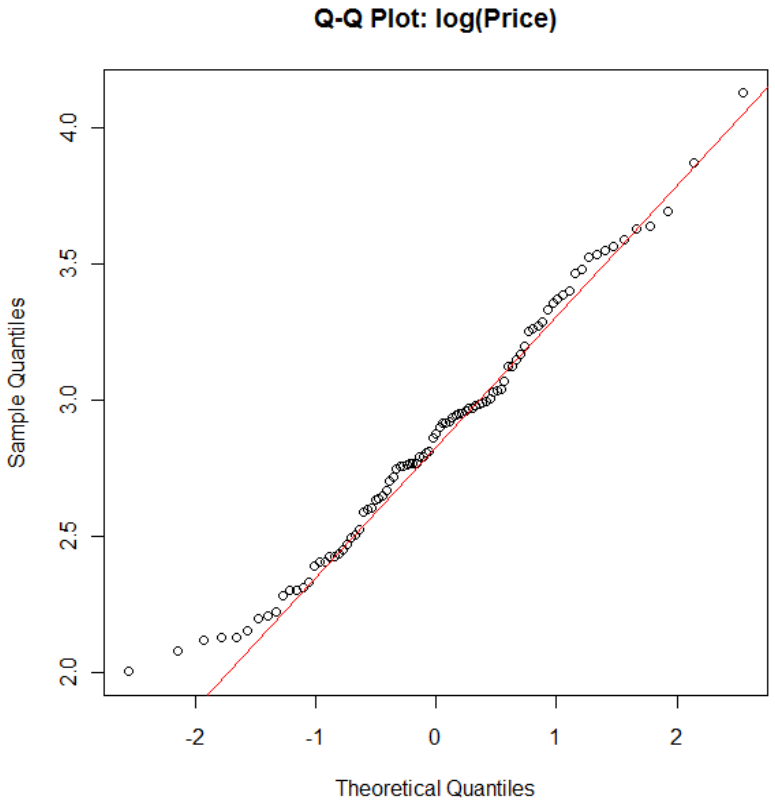
Sample: QQ plot

```
> # QQ plot without logarithmic transformation
> # first: qqnorm, second: qqline
> data(Cars93, package="MASS")
> qqnorm(Cars93$Price, main="Q-Q Plot: Price")
> qqline(Cars93$Price)
>
> # QQ plot without logarithmic transformation
> qqnorm(log(Cars93$Price), main="Q-Q Plot: log(Price)")
> qqline(log(Cars93$Price), col=2)
> # end
```

Sample: QQ plot (cont.)

- Notice that the points in the right plot are much better behaved, staying close to the line except in the extreme left tail. It appears that log(Price) is approximately Normal.





Sample: QQ plot (cont.)

- qqnorm is a generic function the default method of which produces a normal QQ plot of the values in y. qqline adds a line to a normal quantile-quantile plot which passes through the first and third quartiles.
- qqplot produces a QQ plot of two datasets.

Usage

```
qqnorm(y, ...)
## Default S3 method:
qqnorm(y, ylim, main = "Normal Q-Q Plot",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
       plot.it = TRUE, datax = FALSE, ...)

qqline(y, datax = FALSE, ...)

qqplot(x, y, plot.it = TRUE, xlab = deparse(substitute(x)),
       ylab = deparse(substitute(y)), ...)
```

4.3 Analysis of Variance

- Performing One-Way ANOVA
- Problem
  - Data is divided into groups, and the groups are normally distributed.
  - You want to know if the groups have significantly different means.

oneway.test

- Analysis
  - Test whether two or more samples from normal distributions have the same means. The variances are not necessarily assumed to be equal.

```
> oneway.test(x ~ f)
```

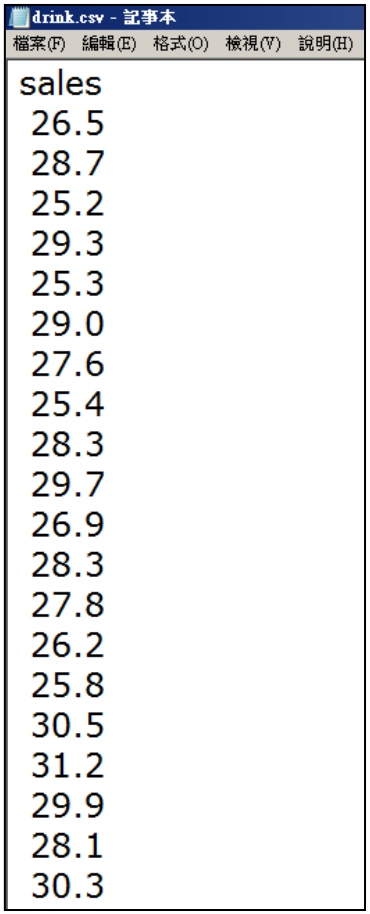
  - *x* is a vector of numeric values and *f* is a factor that identifies the groups.
  - Conventionally, a p-value of less than 0.05 indicates that two or more groups have significantly different means whereas a value exceeding 0.05 provides no such evidence.

Sample: oneway.test

- Comparing the average sales for four drinks.

品牌			
A	B	C	D
26.5	29.0	26.9	30.5
28.7	27.6	28.3	31.2
25.2	25.4	27.8	29.9
29.3	28.3	26.2	28.1
25.3	29.7	25.8	30.3

Save as "drink.csv"



```
> # one-way amova
> # Copy the data to C:\R.data
> # Import data
> drink.sales <- read.table("drink.csv", header=TRUE, sep=",")
> head(drink.sales)
  sales
1  26.5
2  28.7
3  25.2
4  29.3
5  25.3
6  29.0
>
> # drink.type <- factor(gl(4,5,label=c(letters[1:4])))
> drink.type <- gl(4,5,label=c(letters[1:4]))
> drink.type
 [1] a a a a a b b b b b c c c c c d d d d d
Levels: a b c d
>
> drink <- data.frame(drink.type=drink.type, drink.sales)
> head(drink)
  drink.type sales
1          a  26.5
2          a  28.7
3          a  25.2
4          a  29.3
5          a  25.3
6          b  29.0
> class(drink)
[1] "data.frame"
```

Sample: oneway.test (cont.)

```
> # method 1. oneway.test
> drink.oneway <- oneway.test(drink$sales ~ drink$drink.type,
+    var.equal=TRUE)
> drink.oneway

        One-way analysis of means

data:  drink$sales and drink$drink.type
F = 4.5351, num df = 3, denom df = 16, p-value = 0.01749

>
```

Sample – oneway.test (cont.)

aov always assumes equal variances

```
> # method 2. aov
> drink.anova <- aov(drink$sales ~ drink$drink.type)
> summary(drink.anova)
                  Df Sum Sq Mean Sq F value  Pr(>F)
drink$drink.type   3  30.00  10.000  4.5351 0.01749 *
Residuals         16  35.28   2.205
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```
> # method 3. Linear model
> drink.lm <- lm(drink$sales ~ drink$drink.type)
> anova(drink.lm)
Analysis of Variance Table

Response: drink$sales
                  Df Sum Sq Mean Sq F value  Pr(>F)
drink$drink.type   3  30.00  10.000  4.5351 0.01749 *
Residuals         16  35.28   2.205
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> # end
```

4.4 Linear Regression

- A simple linear regression is the most basic model. It's just two variables and is modeled as a linear relationship with an error term:
- x: predictor variable; independent variables
- y: response variable; dependent variables
- We are given the data for x and y. Our mission is to *fit the model*, which will give us the best estimates for β_0 and β_1.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Sample: lm

```
> # linear regression
> # Build-in data: cars
> # x: speed, y: dist
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
> dim(cars)
[1] 50  2
```

Sample: lm (cont.)

```
> # linear model
> cars.lm <- lm(dist~speed, data=cars)
> summary(cars.lm)

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511,     Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12

> # end
```

Verify Information

- ***Is the model statistically significant?***
  - Check the *F* statistic at the bottom of the summary.
- ***Are the coefficients significant?***
  - Check the coefficient's *t* statistics and *p*-values in the summary, or check their confidence intervals.
- ***Is the model useful?***
  - Check the R^2 near the bottom of the summary.
- ***Does the model fit the data well?***
  - Plot the residuals and check the regression diagnostics.
- ***Does the data satisfy the assumptions behind linear regression?***
  - Check whether the diagnostics confirm that a linear model is reasonable for your data.

Regression Information

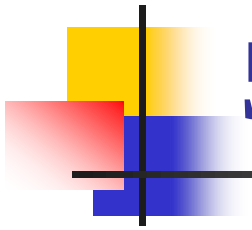
```
> data.lm <- lm(y ~ x1+ x2 + x3)
```

anova(data.lm)	ANOVA table
coefficients(data.lm)	Model coefficients
coef(data.lm)	Same as coefficients(data.lm)
confint(data.lm)	Confidence intervals for the regression coefficients
deviance(data.lm)	Residual sum of squares
effects(data.lm)	Vector of orthogonal effects
fitted(data.lm)	Vector of fitted y values
residuals(data.lm)	Model residuals
resid(data.lm)	Same as residuals(data.lm)
summary(data.lm)	Key statistics, such as R2, the F statistic, and the residual standard error (σ)
vcov(data.lm)	Variance–covariance matrix of the main parameters

Sample: Regression information

```
> anova(cars.lm)
Analysis of Variance Table

Response: dist
          Df Sum Sq Mean Sq F value    Pr(>F)
speed      1  21186 21185.5  89.567 1.49e-12 ***
Residuals 48  11354   236.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> coefficients(cars.lm)
(Intercept)       speed
 -17.579095    3.932409
> coef(cars.lm)
(Intercept)       speed
 -17.579095    3.932409
```

```
> residuals(cars.lm)
         1          2          3          4          5          6          7          8          9
 3.849460  11.849460  -5.947766  12.052234   2.119825  -7.812584  -3.744993   4.255007  12.255007
        10         11         12         13         14         15         16         17         18
-8.677401   2.322599 -15.609810  -9.609810  -5.609810  -1.609810  -7.542219   0.457781   0.457781
        19         20         21         22         23         24         25         26         27
12.457781 -11.474628  -1.474628  22.525372  42.525372 -21.407036 -15.407036  12.592964 -13.339445
        28         29         30         31         32         33         34         35         36
-5.339445 -17.271854  -9.271854   0.728146 -11.204263   2.795737  22.795737  30.795737 -21.136672
        37         38         39         40         41         42         43         44         45
-11.136672  10.863328 -29.069080 -13.069080  -9.069080  -5.069080   2.930920  -2.933898 -18.866307
        46         47         48         49         50
 -6.798715  15.201285  16.201285  43.201285   4.268876
```



5. Application

5.1 R Commander

5.2 RStudio

5.3 Quality Control Chart

5.4 SVM

5.1 R Commander

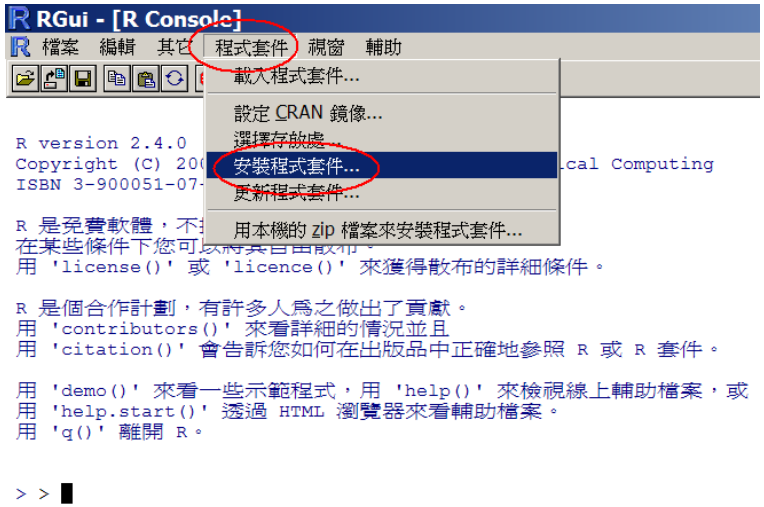
- Introduction - R Commander package
- Download and Starting R Commander
- Input data
- Summary data
- Graphics
- Probability distribution
- Regression analysis

Introduction - Rcmdr package

- The Rcmdr package provides a basic-statistics graphical user interface to R called the “R Commander.”。
- The design objectives of the R Commander were as follows: to support, through an easy-to-use, extensible, cross-platform GUI, the statistical functionality required for a basic-statistics course.

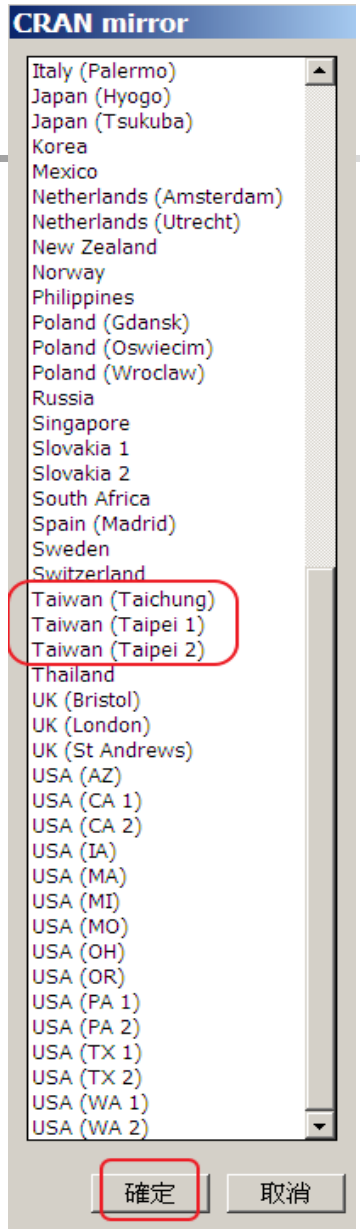
Download and Start Rcmdr

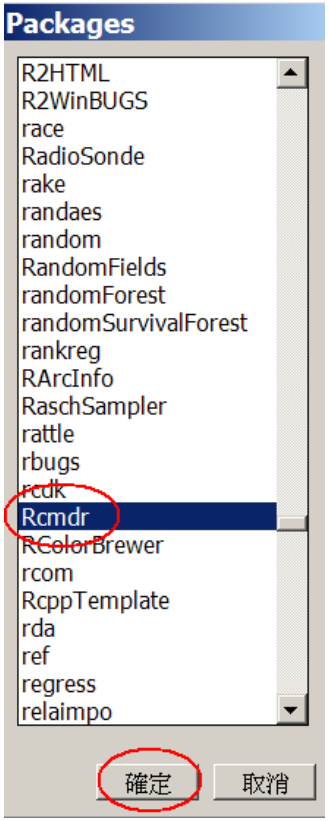
- Down load R Commander package
  - an active Internet connection
  - R → 程式套件 \ 安裝程式套件... \



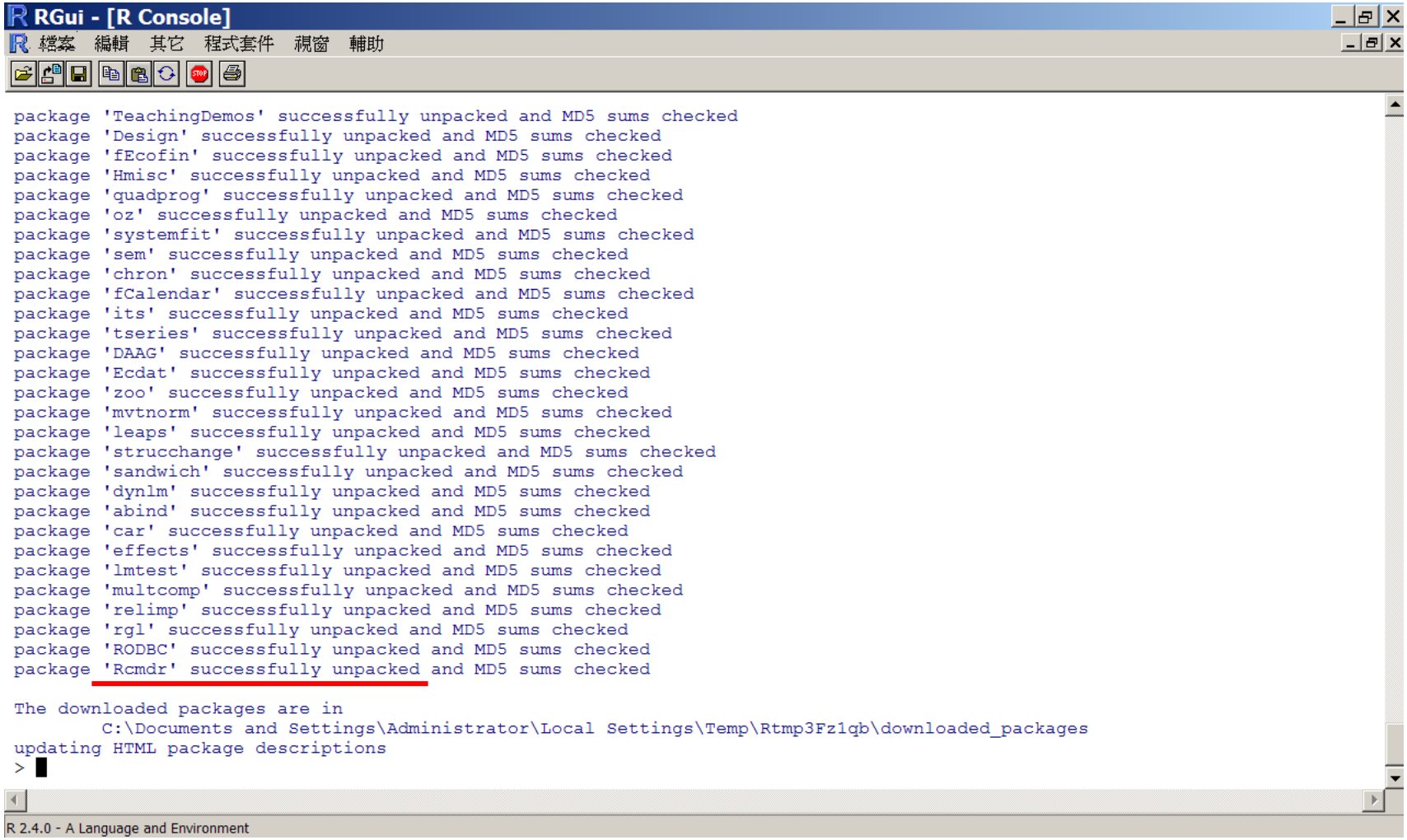

```
> install.packages("Rcmdr", dependencies=TRUE)
Installing package(s) into 'C:/Users/Administrator/Documents/R/win-library/2.13'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
```

- CRAN mirror 視窗
  → 選取 Taiwan (Taipei 1)
  → 確定
- Packages視窗
  → 選取 Rcmdr
  → 確定





Installation



Start R Commander

功能表

工具列

按此鈕可切換不同資料集

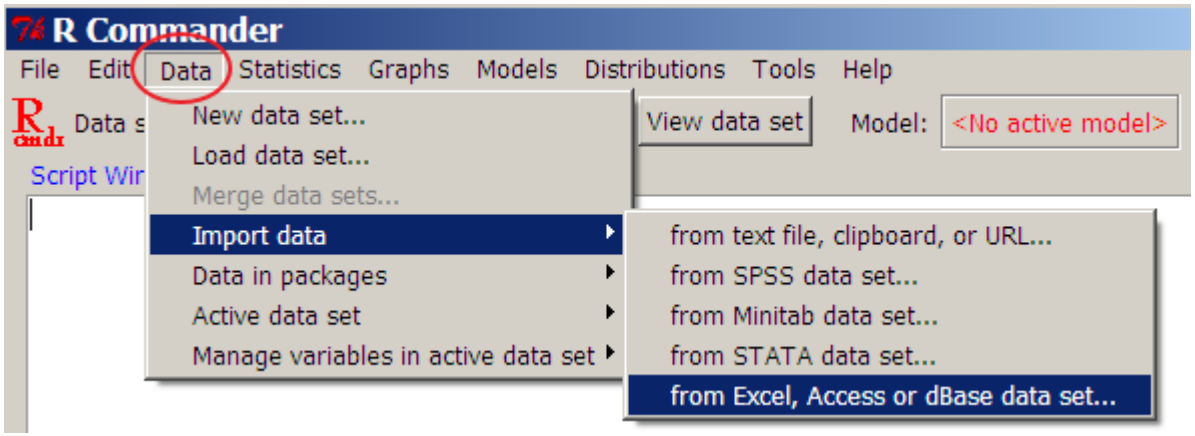
編輯區(輸入R程式碼)

輸出區

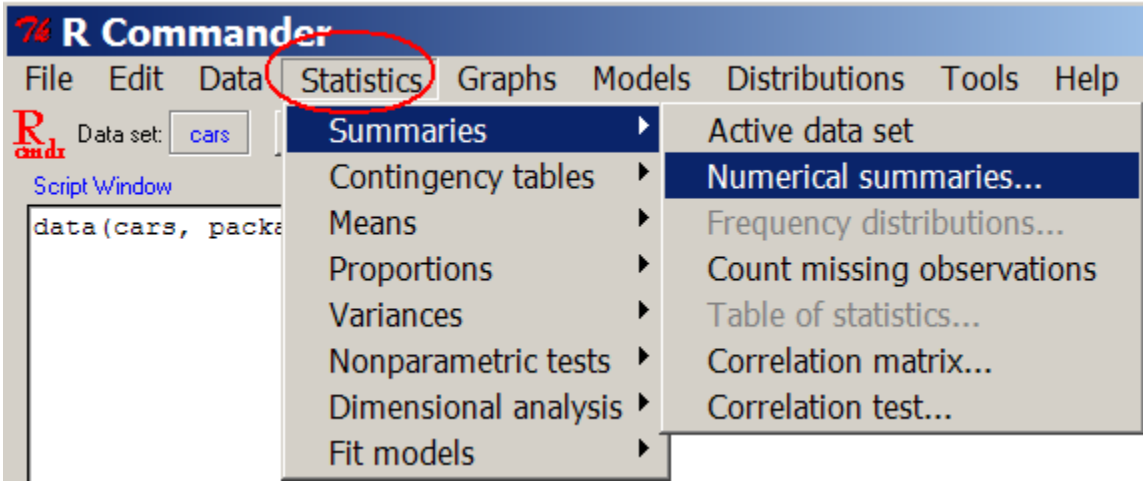
訊息區

Features (1/3)

Data:輸入/編輯資料

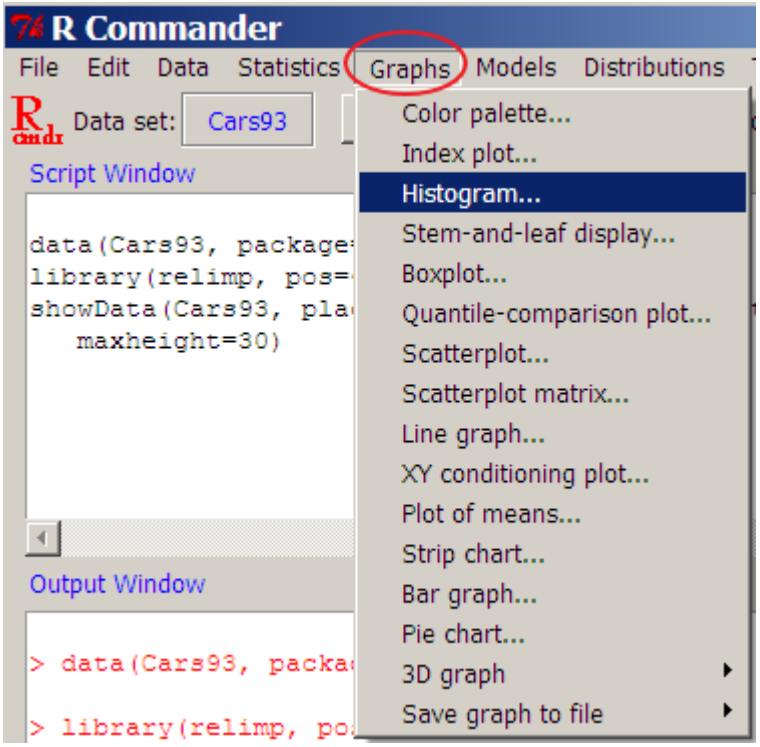


Statistics: 統計檢定分析

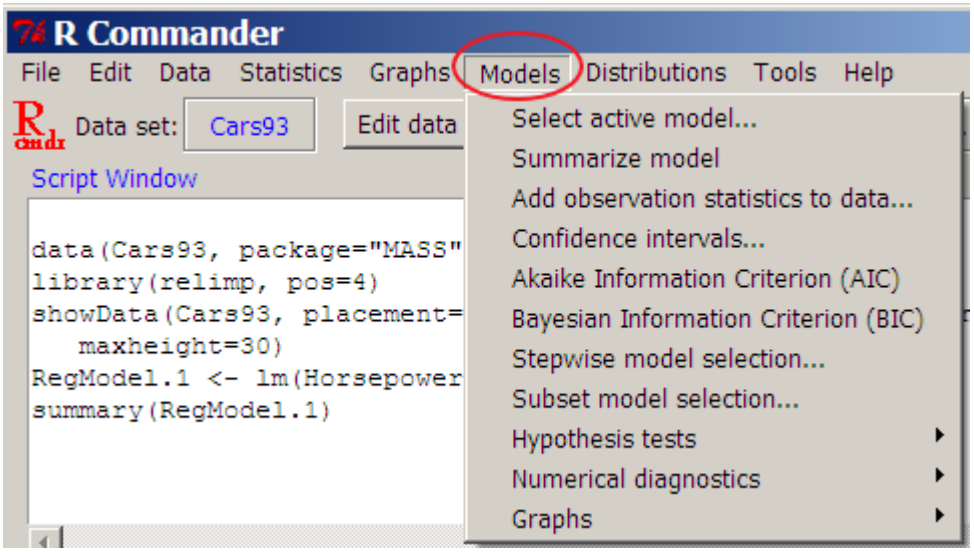


Features (2/3)

Graphs: 繪圖

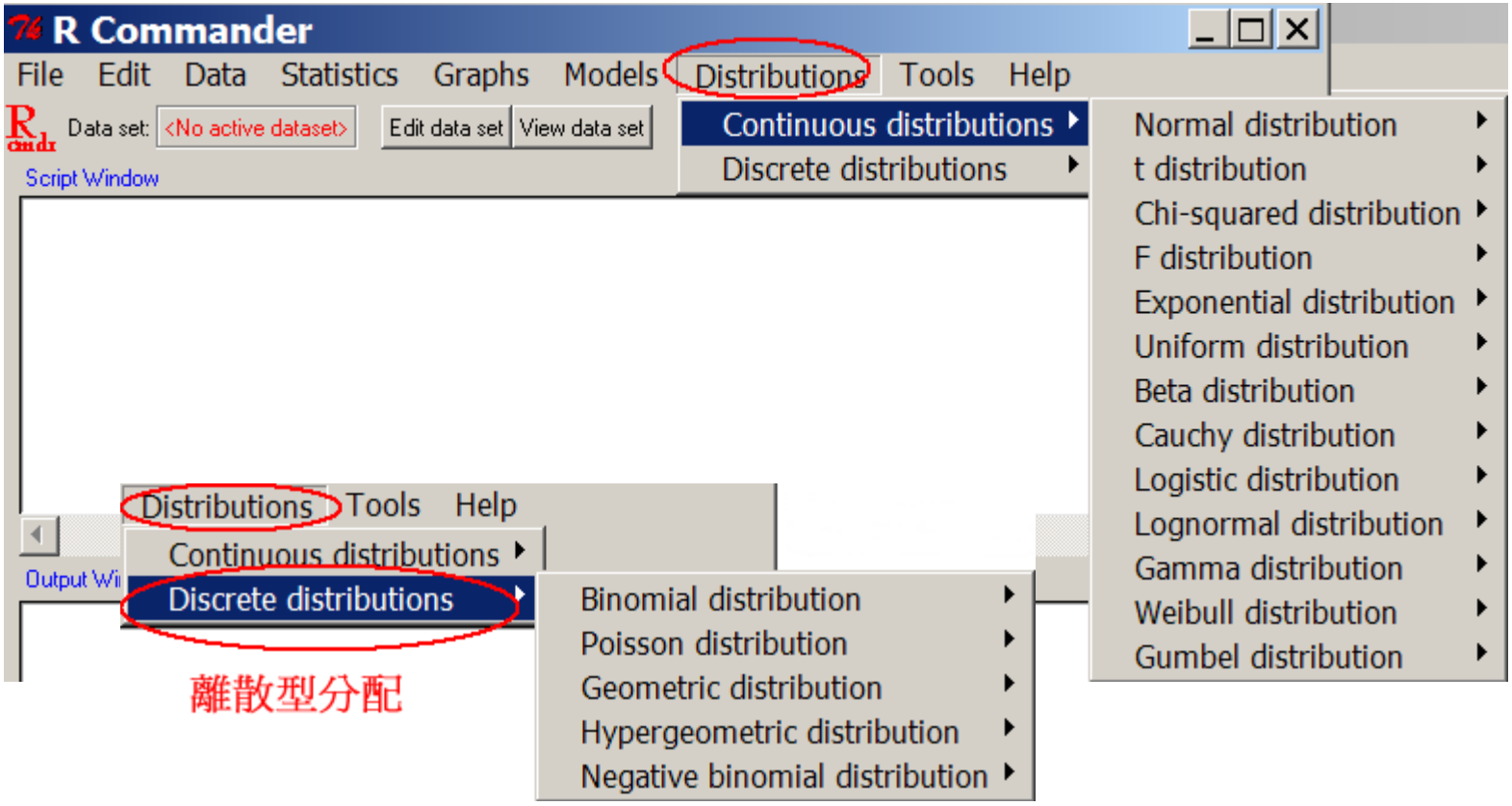


Models: 建立統計模型



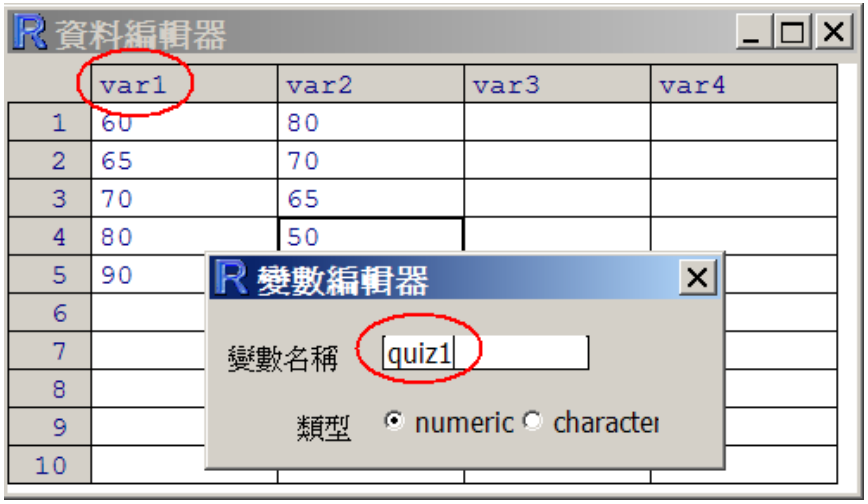
Features (3/3)

Distributions: 產生連續型/離散型機率分配 樣本



Input data (Input in Console)

- Data → New data set…
- 按 var1 可直接更改變數名稱,變數名稱不可有空白
- 適用於少量資料



Input data (Import data)

- 讀入文字檔:

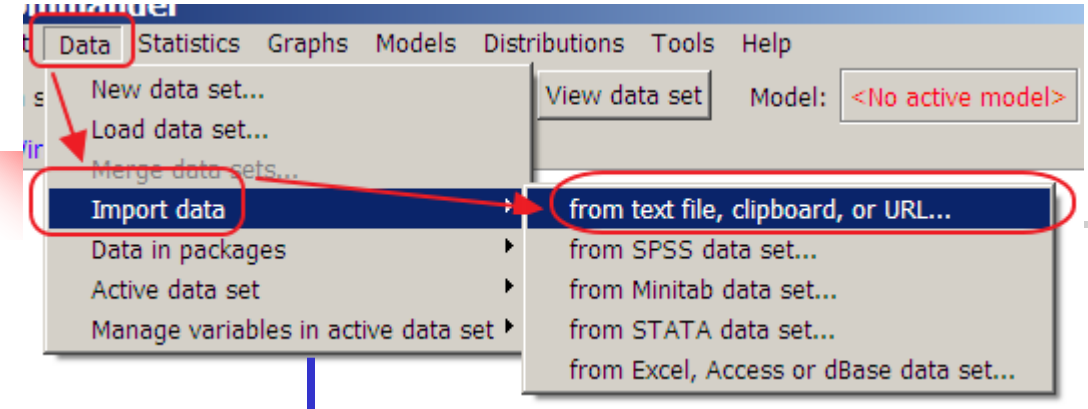
C:\Program Files\R\R-2.13.0\library\Rcmdr\etc

文字檔 第1列為 變數名稱,有5個變數,含遺漏值 NA

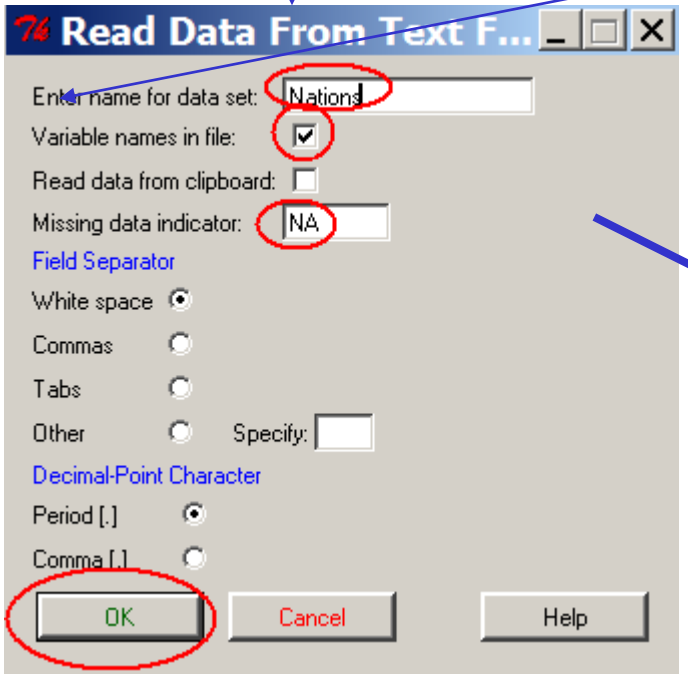
1. 數值-TFR (the total fertility rate, expressed as number of children per woman)
2. 數值- contraception (the rate of contraceptive use among married women, in percent)
3. 數值- infant.mortality (the infant-mortality rate per 1000 live births)
4. 數值- GDP (gross domestic product per capita, in U.S. dollars)
5. 字串- region.

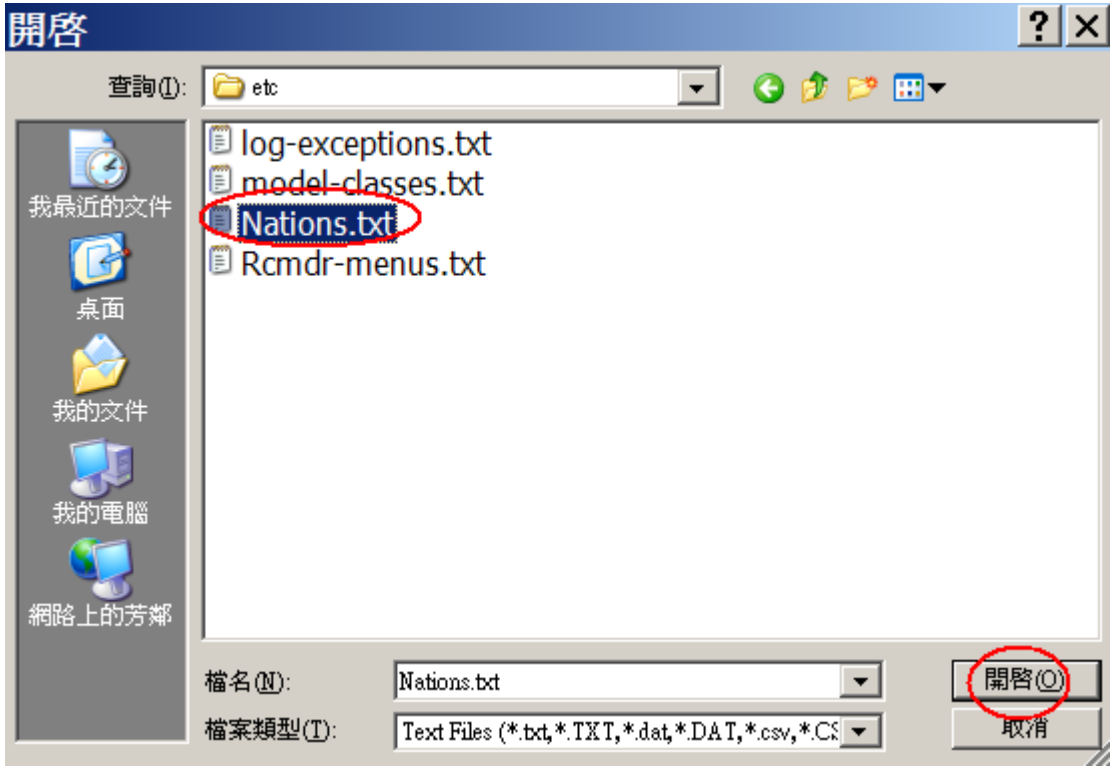
列名稱:國家名稱

	TFR	contraception	infant.mortality	GDP	region
Afghanistan	6.90	NA	154	2848	Asia
Albania	2.60	NA	32	863	Europe
Algeria	3.81	52	44	1531	Africa
American-Samoa	NA	NA	11	NA	Oceania
Andorra	NA	NA	NA	NA	Europe
Angola	6.69	NA	124	355	Africa

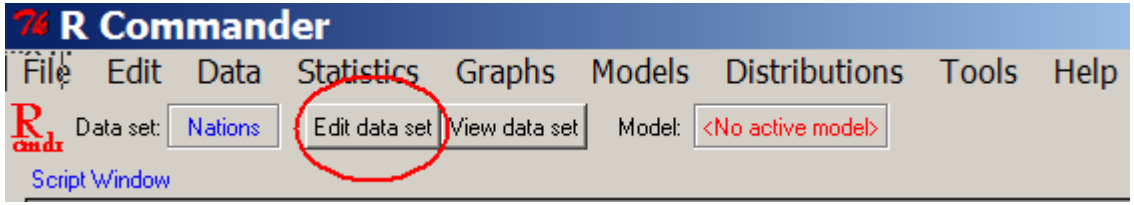


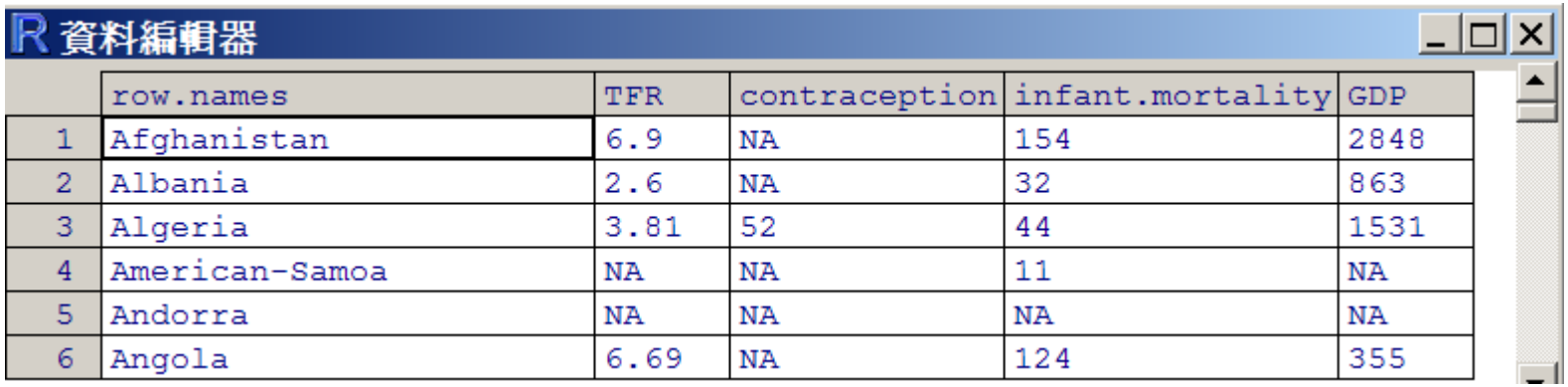
輸入資料集名稱: **Nations**





- 按 **Edit data set** 可編輯資料

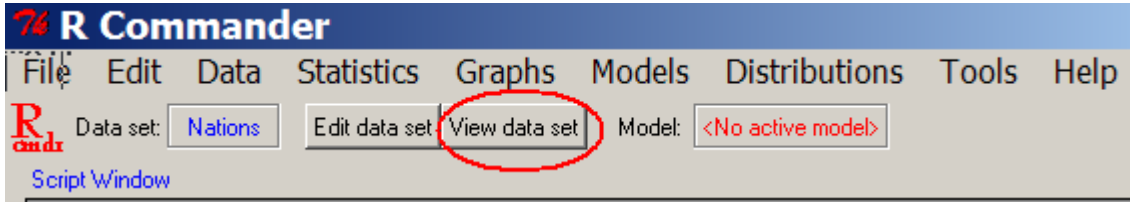


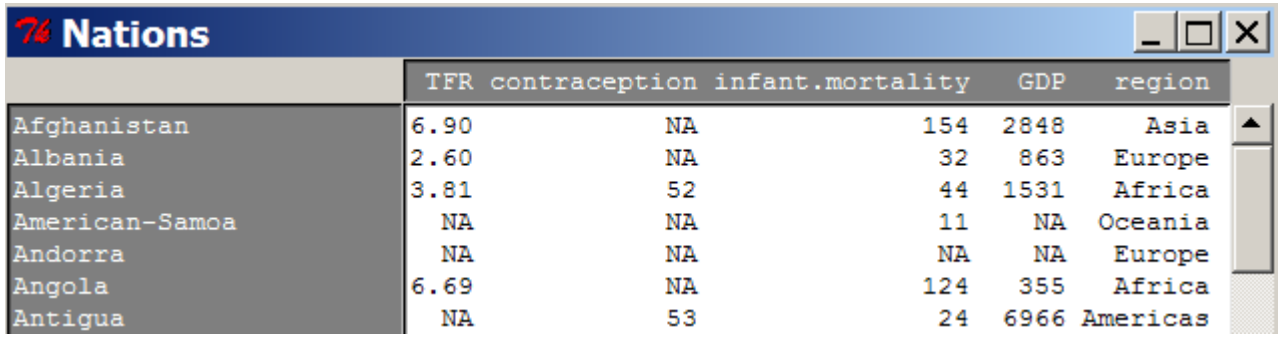


	row.names	TFR	contraception	infant.mortality	GDP
1	Afghanistan	6.9	NA	154	2848
2	Albania	2.6	NA	32	863
3	Algeria	3.81	52	44	1531
4	American-Samoa	NA	NA	11	NA
5	Andorra	NA	NA	NA	NA
6	Angola	6.69	NA	124	355

- 修改資料後，按  即可關閉編輯視窗

- 按 **View data set** 可檢視資料





	TFR	contraception	infant.mortality	GDP	region
Afghanistan	6.90	NA	154	2848	Asia
Albania	2.60	NA	32	863	Europe
Algeria	3.81	52	44	1531	Africa
American-Samoa	NA	NA	11	NA	Oceania
Andorra	NA	NA	NA	NA	Europe
Angola	6.69	NA	124	355	Africa
Antigua	NA	53	24	6966	Americas

- 資料 “data frames” 型式
- 在Rcmdr編輯區中可顯示匯入資料指令 **read.table**

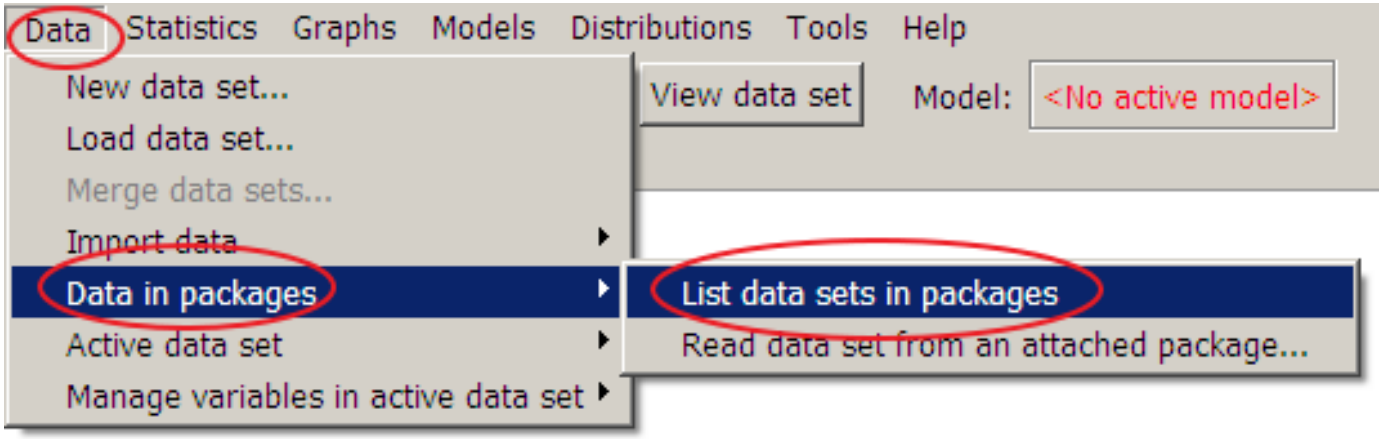
```
Nations <-
  read.table("C:/Program Files/R/R-2.13.0/library/Rcmdr/etc/Nations.txt",
  header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
```

Display Data in available package

- R 的 packages 中含有許多資料集可供測試.

  Data \ Data in packages \ List data sets in packages

  顯示所有可用資料集,其中 datasets package 有許多資料可供測試·




```
> # Liew all available data sets
> data()
> # end
```

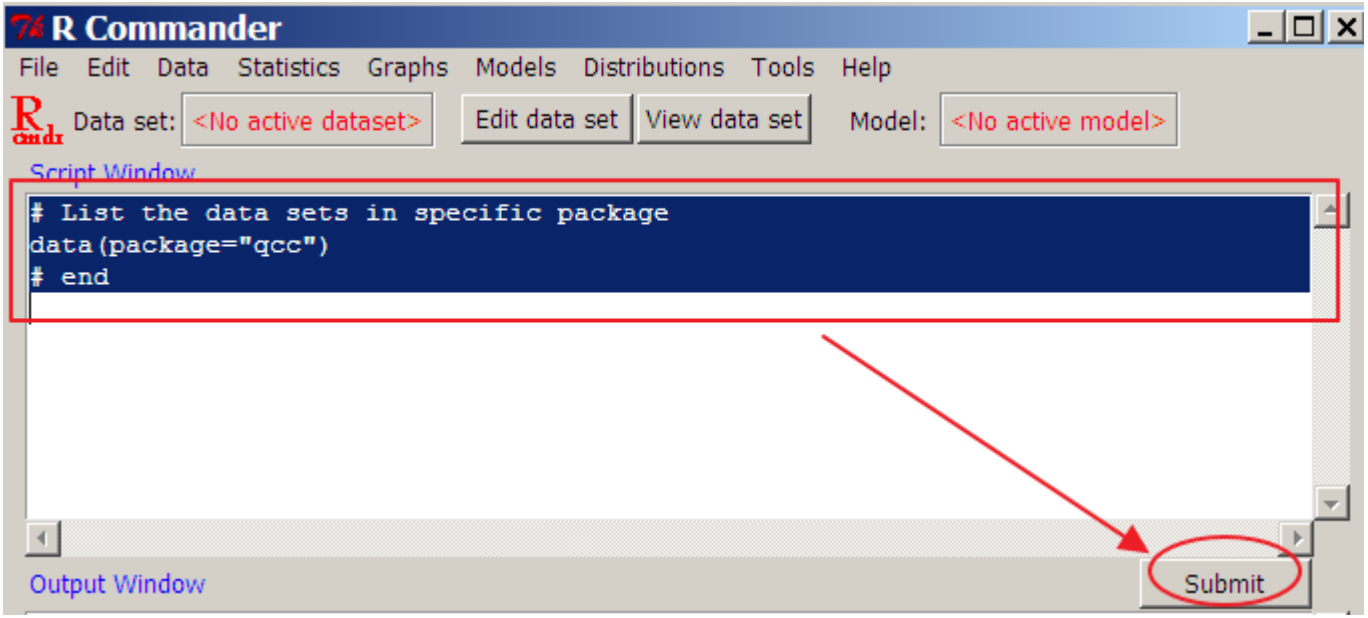
Display Data in available package (cont.)

RGui - [R data sets]

檔案 編輯 視窗

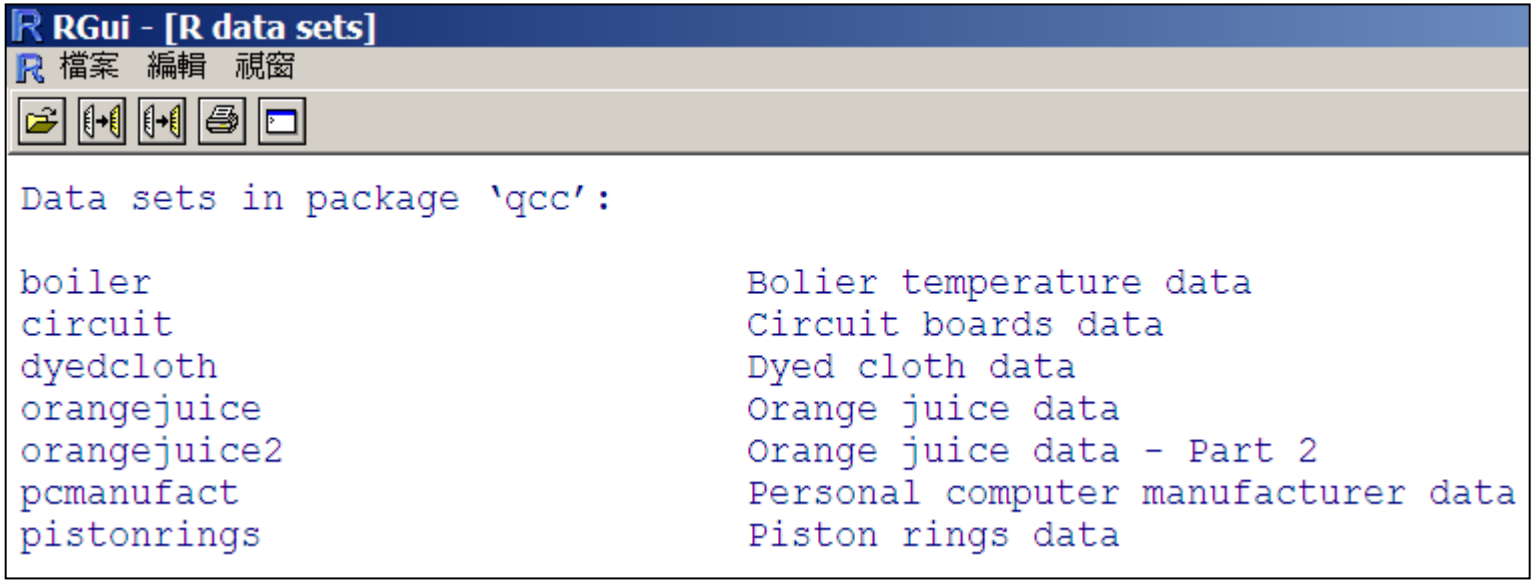
```
Data sets in package 'car':

AMSsurvey               American Math Society Survey Data
Adler                   Experimenter Expectations
Angell                  Moral Integration of American Cities
Anscombe                U. S. State Public-School Expenditures
Baumann                 Methods of Teaching Reading Comprehension
Bfox                    Canadian Women's Labour-Force Participation
Blackmoor               Exercise Histories of Eating-Disordered and
                        Control Subjects
Burt                    Fraudulent Data on IQs of Twins Raised Apart
CanPop                  Canadian Population Data
Chile                   Voting Intentions in the 1988 Chilean
                        Plebiscite
Chirot                  The 1907 Romanian Peasant Rebellion
Cowles                  Cowles and Davis's Data on Volunteering
Davis                   Self-Reports of Height and Weight
DavisThin               Davis's Data on Drive for Thinness
Depredations            Minnesota Wolf Depredation Data
Duncan                  Duncan's Occupational Prestige Data
Ericksen                The 1980 U.S. Census Undercount
Florida                 Florida County Voting
Freedman                Crowding and Crime in U. S. Metropolitan Areas
Friendly                Format Effects on Recall
Ginzberg                Data on Depression
Greene                  Refugee Appeals
Guyer                   Anonymity and Cooperation
Hartnagel               Canadian Crime-Rates Time Series
Highway1                Highway Accidents
Leinhardt               Data on Infant-Mortality
Mandel                  Contrived Collinear Data
Migration               Canadian Interprovincial Migration Data
Moore                   Status, Authoritarianism, and Conformity
Mroz                    U.S. Women's Labor-Force Participation
OBrienKaiser            O'Brien and Kaiser's Repeated-Measures Data
Ornstein                Interlocking Directorates Among Major Canadian
                        Firms
Pottery                 Chemical Composition of Pottery
Prestige                Prestige of Canadian Occupations
```

Display Data in specific package



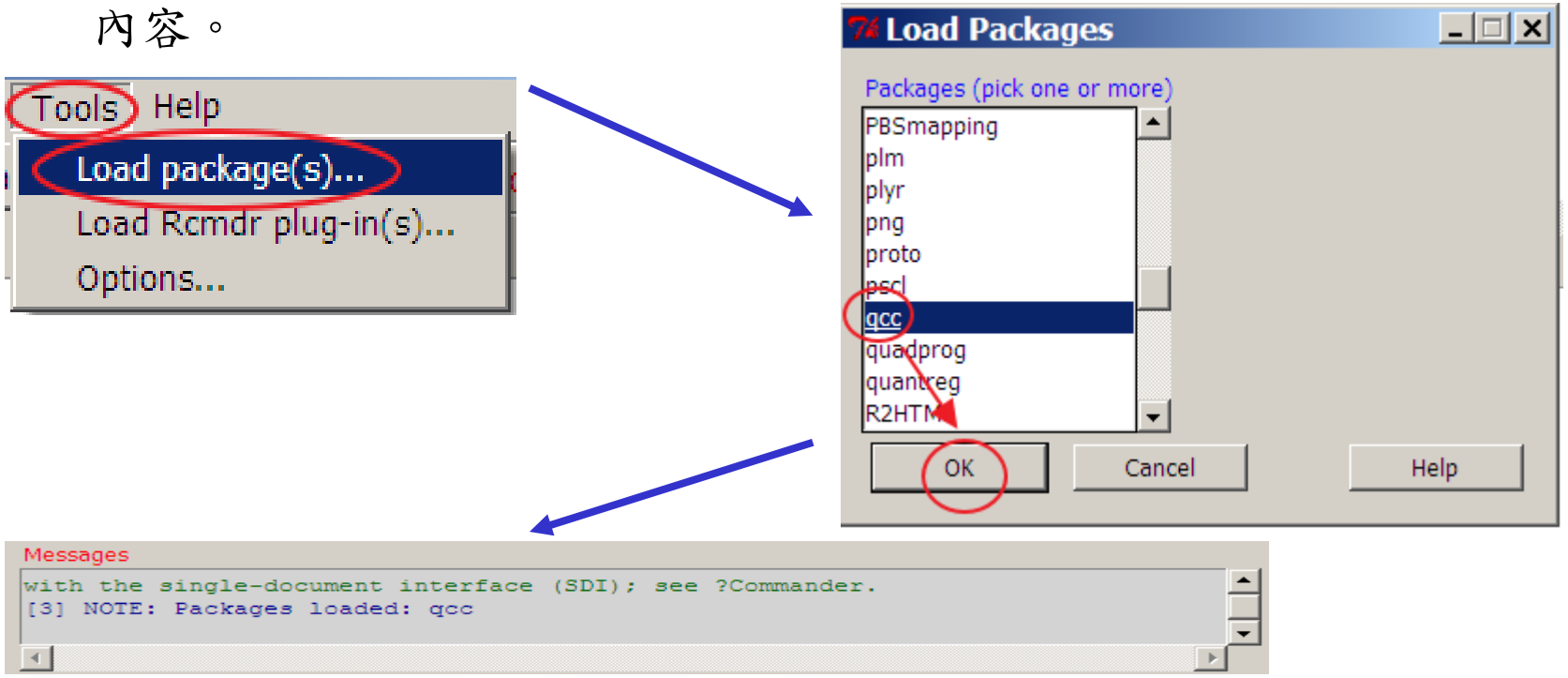

```
> # List the data sets in specific package
> data(package="qcc")
> # end
```

Display Data in specific package (cont.)



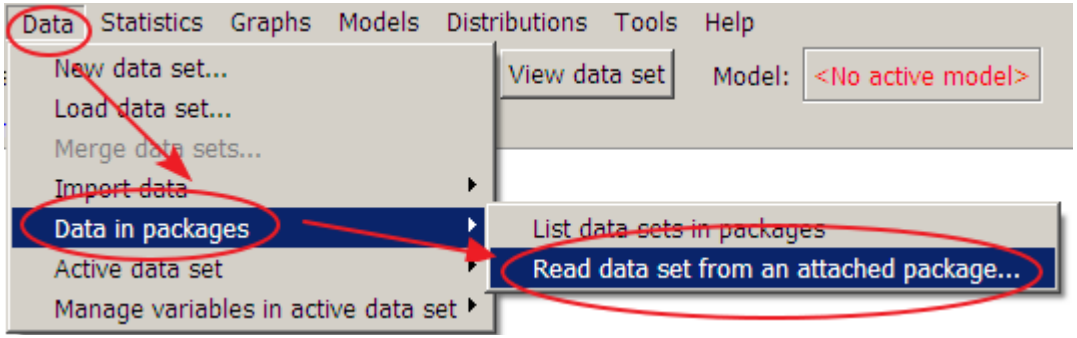
Load package and data

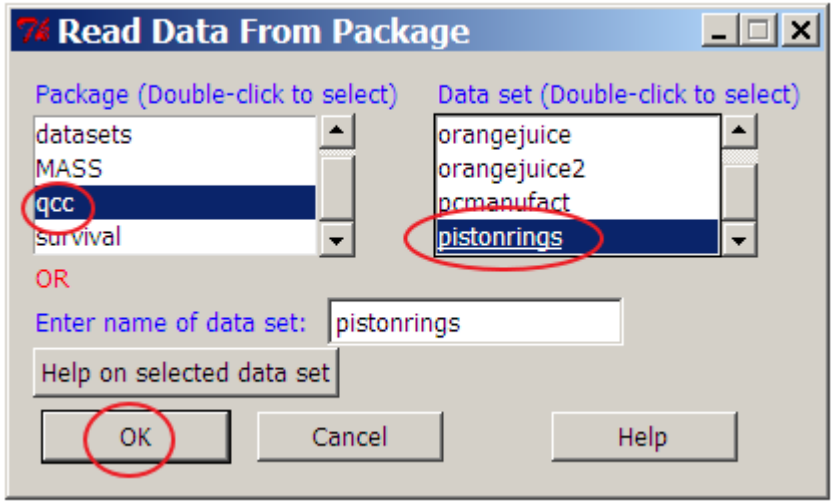
- 如果須使用的資料沒有在預設的資料集中,則可先載入package 再載入資料集。
- Step 1:
  Tools \ Load package(s)… \ 選取 qcc \ 按 ok, 訊息區會有載入package 內容。



Load package and data (cont.)

- Step 2. 載入 loaded package 的資料集
  Data \ Data in packages \ Read data set from an attached package…

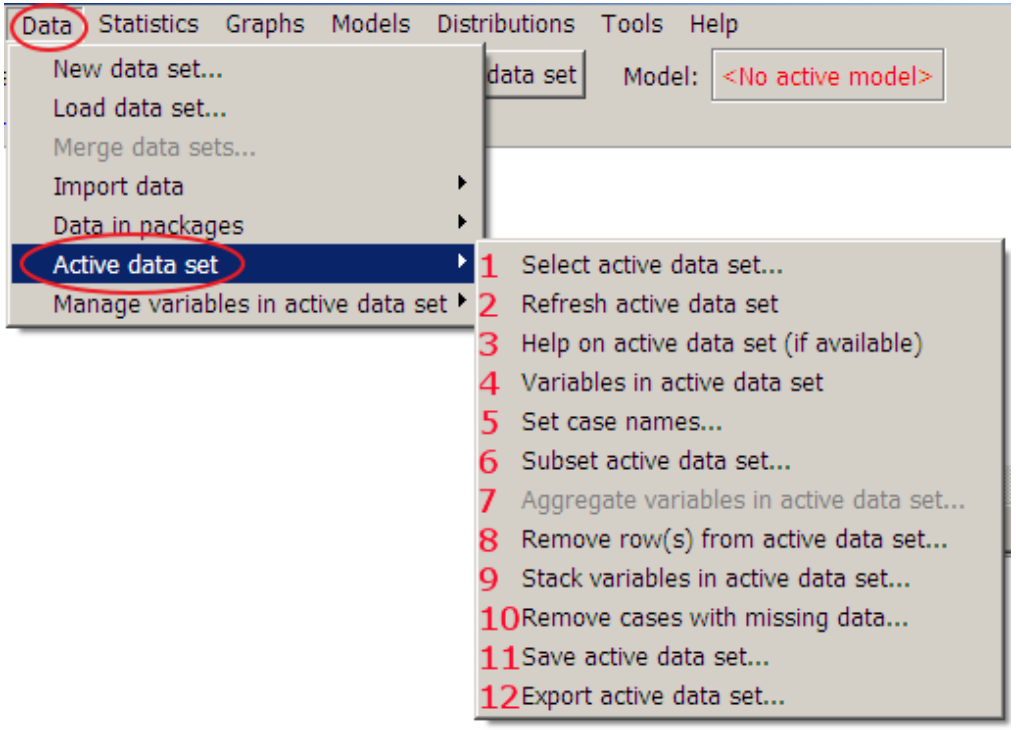





```
> # Load the data sets in specific package
> data(pistonrings, package="qcc")
> # end
```

```
> # TRY
> # Select data with "sampe=1"?
> pistonrings.sample1 <-
> pistonrings.sample1
  diameter sample trial
1   74.030      1  TRUE
2   74.002      1  TRUE
3   74.019      1  TRUE
4   73.992      1  TRUE
5   74.008      1  TRUE
> # end
```

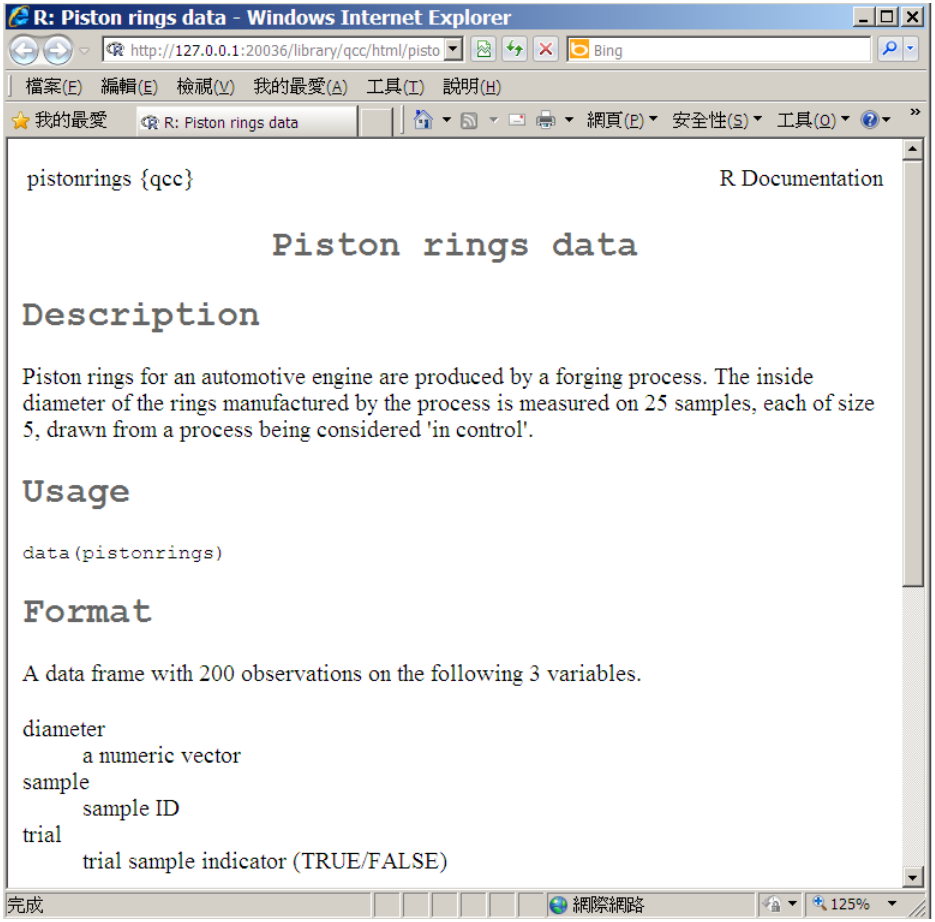
Active data set



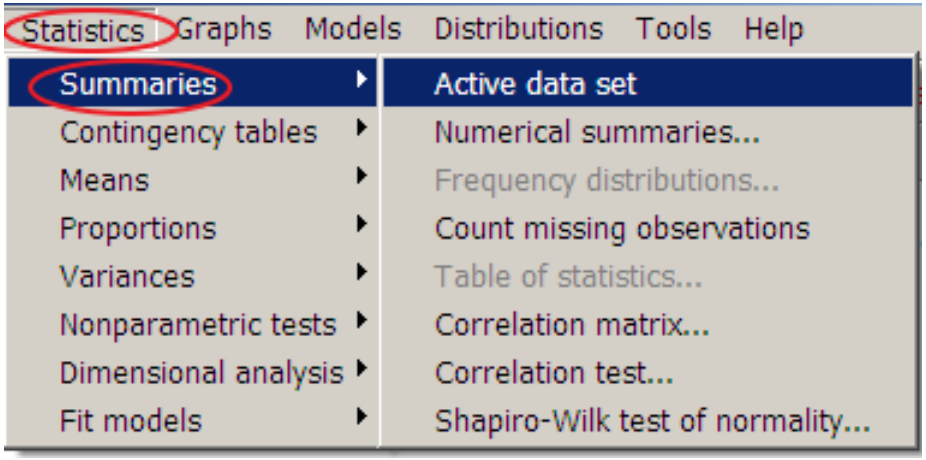
1. 如果有使用二個以上資料集, 選擇目前的主要資料集
2. 更新資料集
3. 查詢資料集的說明檔
4. 顯示資料集的變數名稱
5. 選取並設定資料列名稱
6. 選取現有資料集的子集合
7. 彙總變數
8. 刪除資料列
9. 串連資料集
10. 刪除含遺漏值的資料
11. 儲存資料集 (.rda)
12. 匯出資料

Help for data set

```
> # Help and column names
> help("pistonrings")
> names(pistonrings)
[1] "diameter" "sample"   "trial"
> # END
```

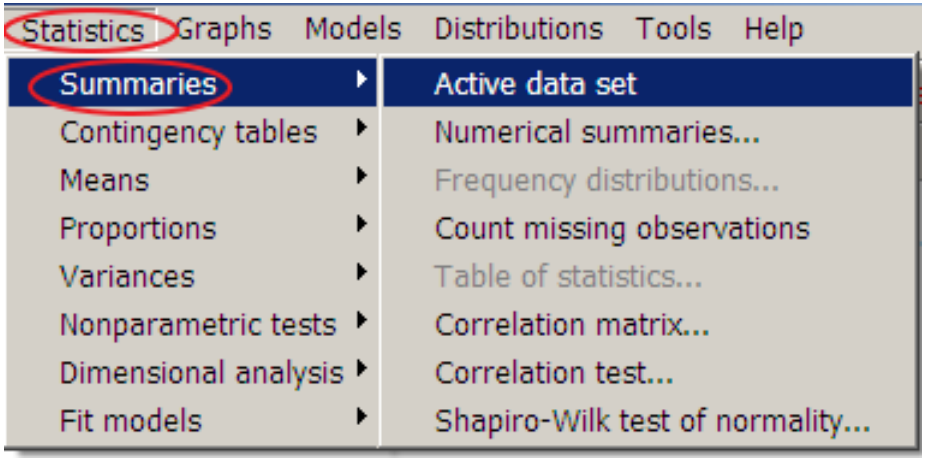


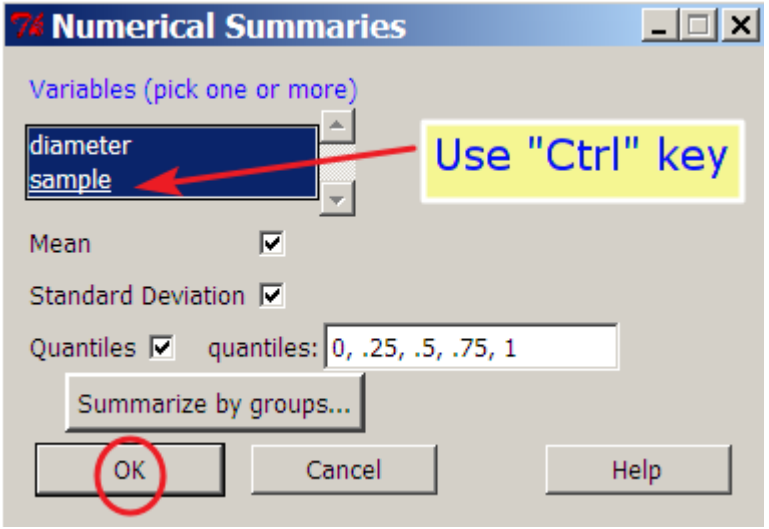
Summaries: Active data set



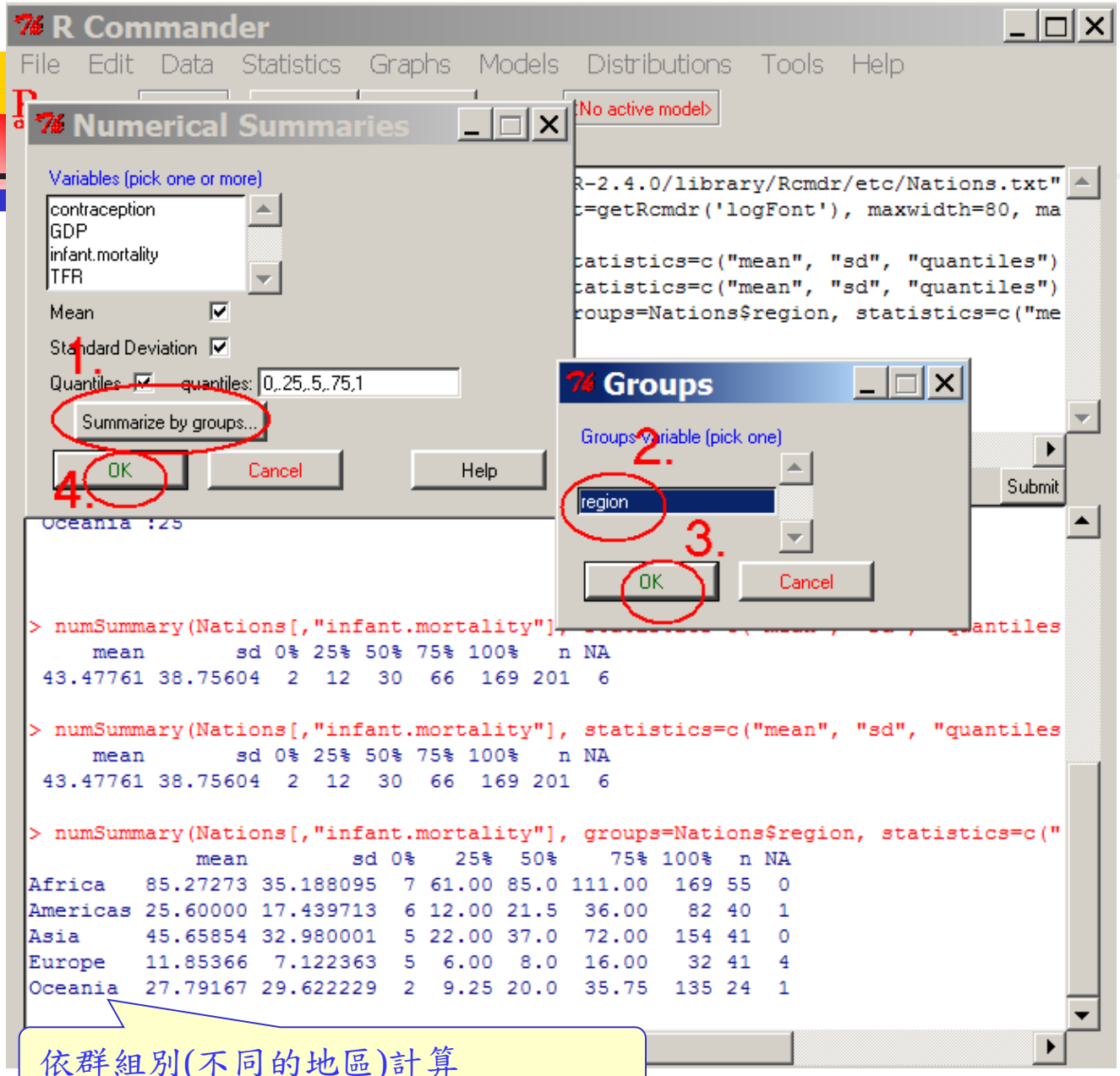

```
> summary(pistonrings)
    diameter          sample          trial
 Min.   :73.97   Min.   : 1.00   Mode :logical
 1st Qu.:74.00   1st Qu.:10.75   FALSE:75
 Median :74.00   Median :20.50   TRUE :125
 Mean   :74.00   Mean   :20.50   NA's :0
 3rd Qu.:74.01   3rd Qu.:30.25
 Max.   :74.04   Max.   :40.00
```

Summaries: Numerical summaries



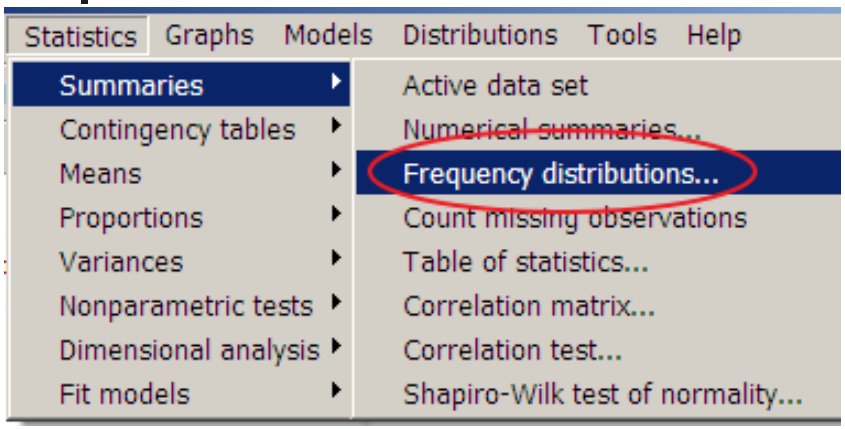


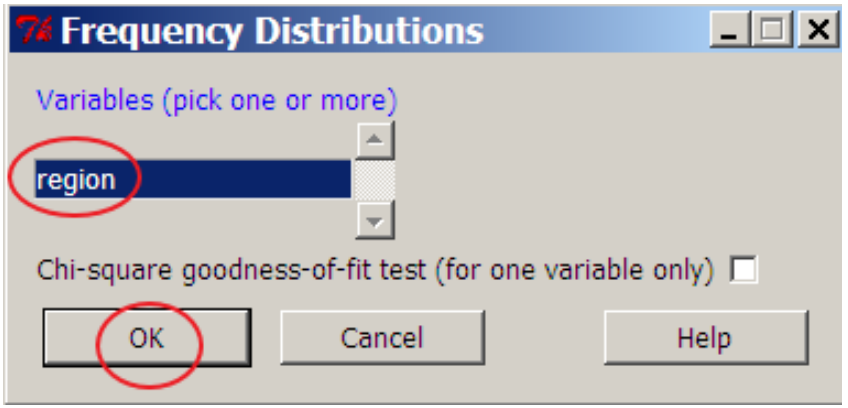

```
> numSummary(pistonrings[,c("diameter", "sample")], statistics=c("mean", "sd",
+     "quantiles"), quantiles=c(0,.25,.5,.75,1))
             mean           sd     0%    25%    50%   75%   100%   n
diameter 74.0036   0.01141712 73.967 73.995 74.003 74.01 74.036 200
sample   20.5000 11.57236354  1.000 10.750 20.500 30.25 40.000 200
```



依群組別(不同的地區)計算

Summaries: Frequency distribution





Output Window

```
> load("C:/R.data/Nations.rda")

> .Table <- table(Nations$region)

> .Table  # counts for region

  Africa Americas     Asia   Europe  Oceania
      55       41       41       45       25

> round(100*.Table/sum(.Table), 2)  # percentages for region

  Africa Americas     Asia   Europe  Oceania
   26.57    19.81    19.81    21.74    12.08

> remove(.Table)
```

顯示相對次數分配表

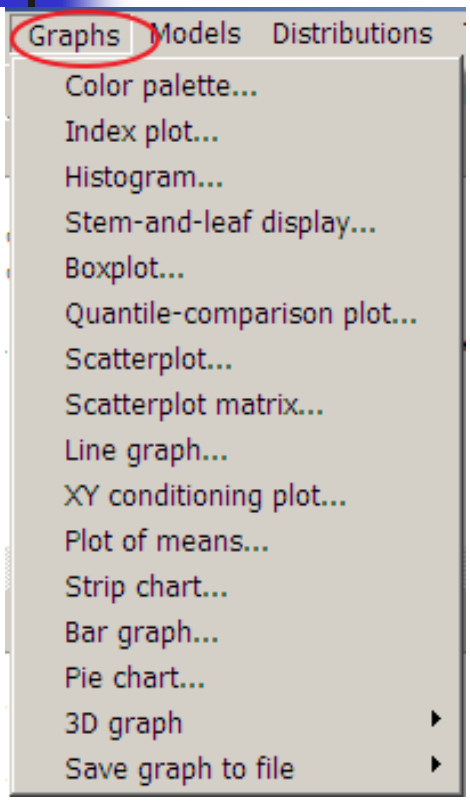
Set global options

- options(digits=16)
  - 設定 整數+小數點 顯示位數為16位
  - 預設值為7.

```
> # set global options
> x <- sqrt(2)
> x
[1] 1.414214
> options(digits=16)
> x
[1] 1.414213562373095
> # end
```

- ?options
  - 顯示options輔助說明

Graphics

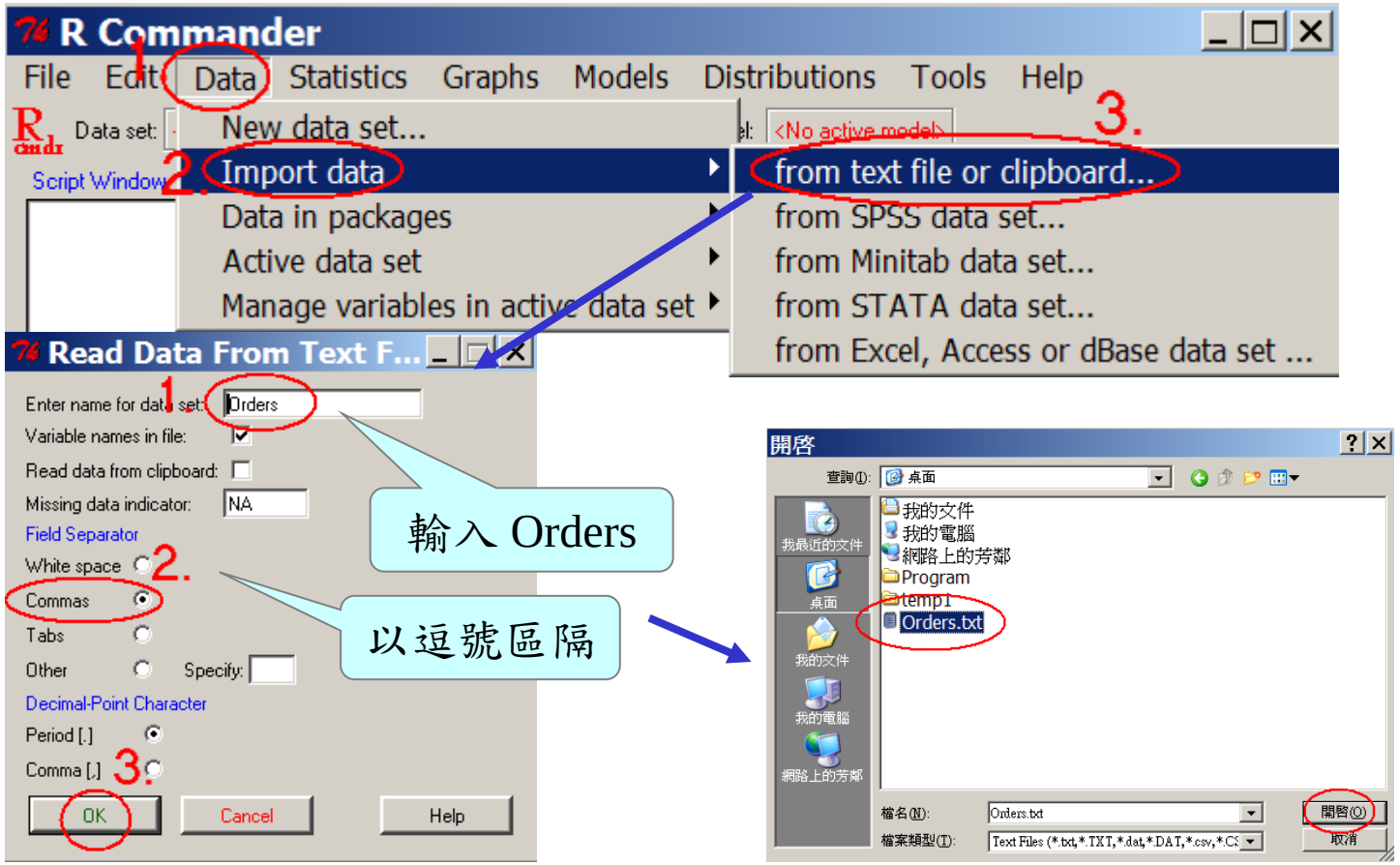


下載10萬筆測試資料- WEPA 2011 SINCE 2002

http://web.ydu.edu.tw/~alan9956/docu/refer/R03_Orders.txt

Histogram plot

- Orders 資料集 (十萬筆資料)



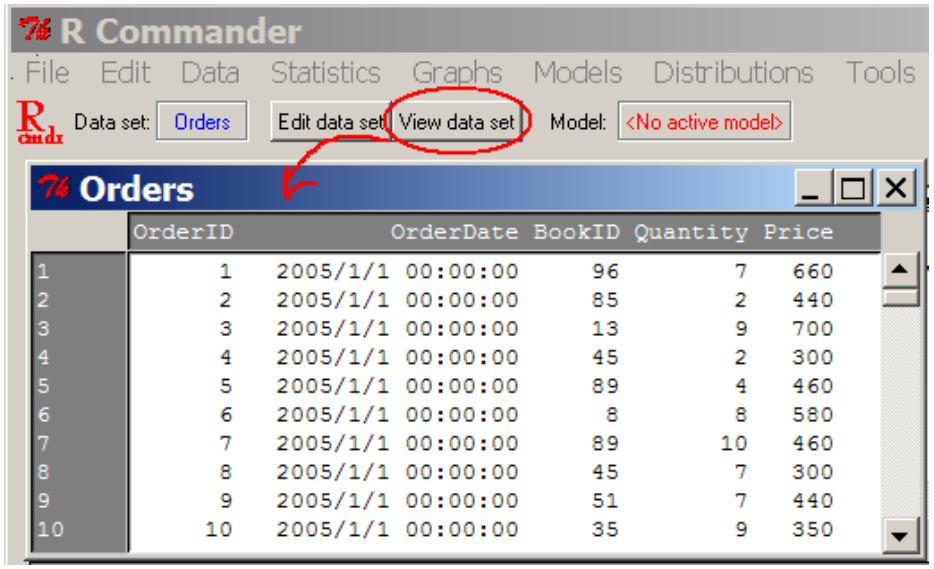
Histogram plot (cont.)

- 完成後訊息視窗顯示資料集 Orders 包括100,000筆資料, 5個欄位.

Messages

```
NOTE: The dataset Orders has 100000 rows and 5 columns.
```

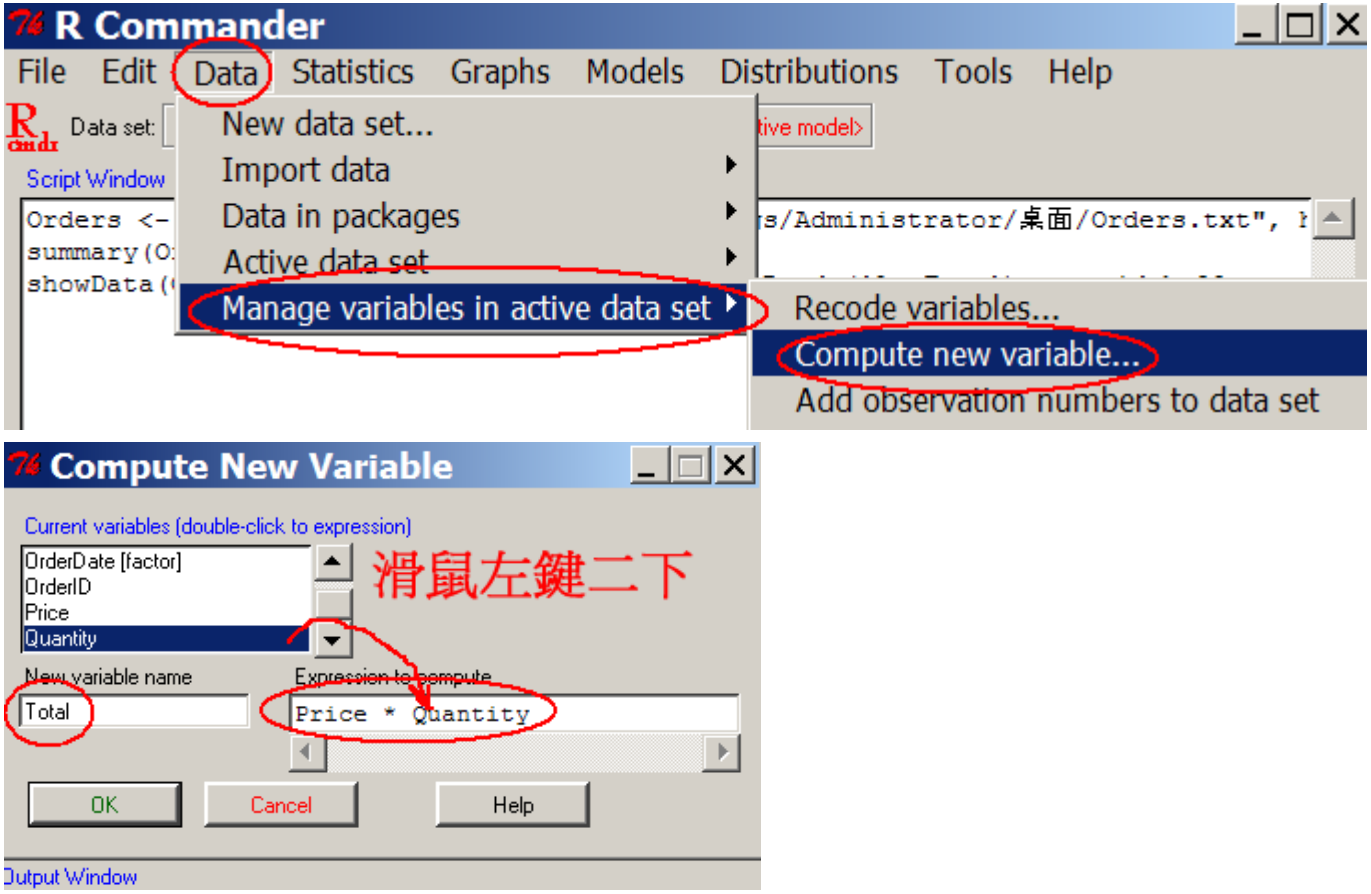
- 檢視資料集 View data set



	OrderID	OrderDate	BookID	Quantity	Price
1	1	2005/1/1 00:00:00	96	7	660
2	2	2005/1/1 00:00:00	85	2	440
3	3	2005/1/1 00:00:00	13	9	700
4	4	2005/1/1 00:00:00	45	2	300
5	5	2005/1/1 00:00:00	89	4	460
6	6	2005/1/1 00:00:00	8	8	580
7	7	2005/1/1 00:00:00	89	10	460
8	8	2005/1/1 00:00:00	45	7	300
9	9	2005/1/1 00:00:00	51	7	440
10	10	2005/1/1 00:00:00	35	9	350

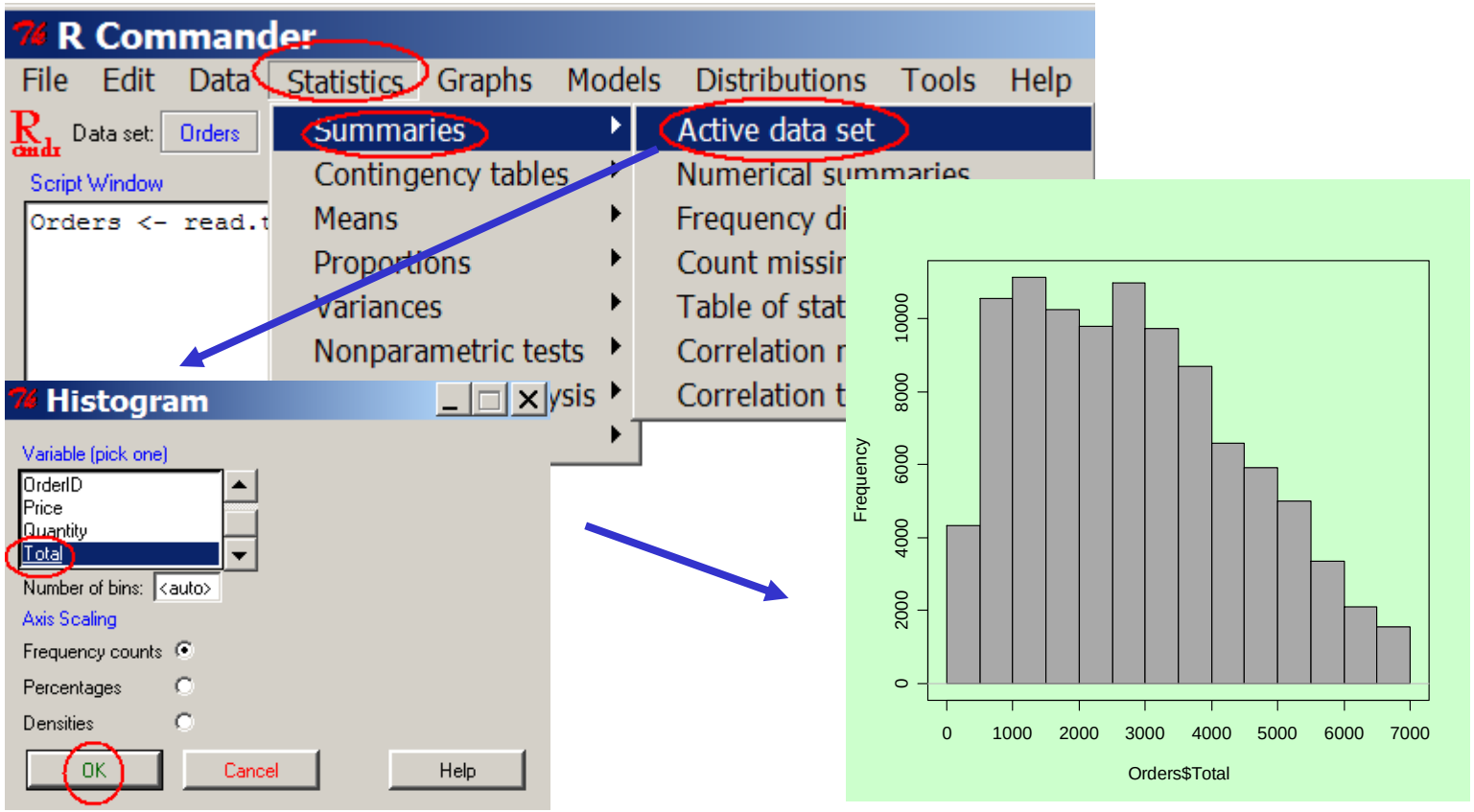
Histogram plot (cont.)

- 加入新變數 Total.



Histogram plot (cont.)

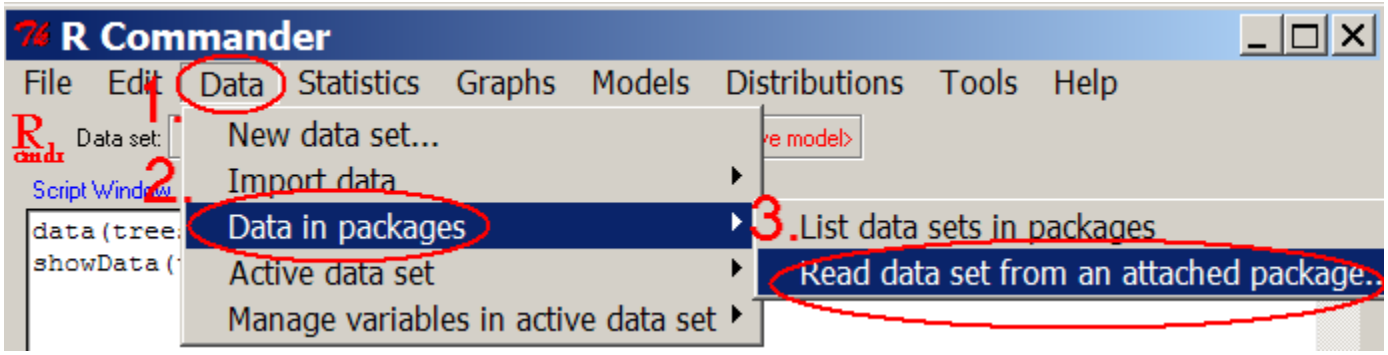
- 繪製直方圖 Histogram



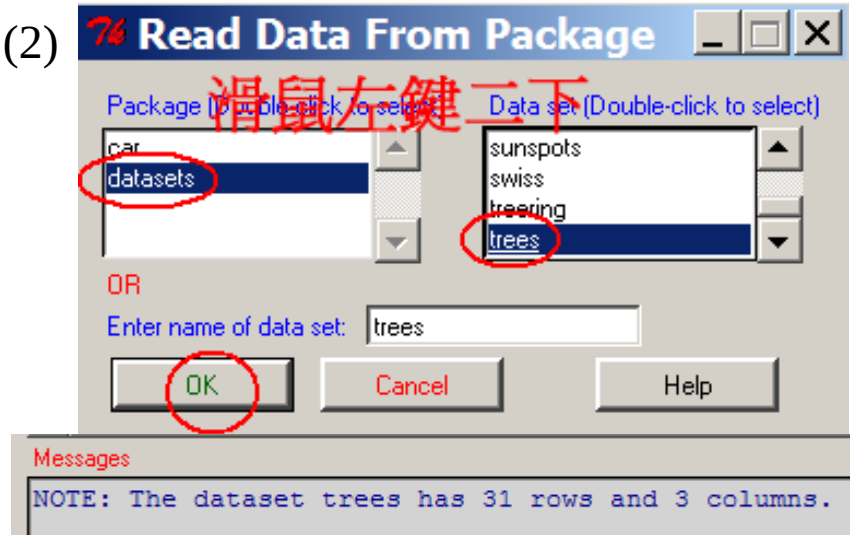
3D plot

- 匯入 trees 資料集

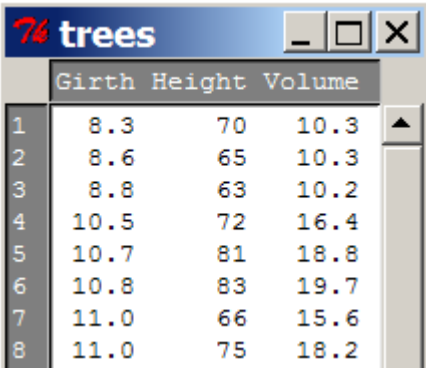
(1)



(2)

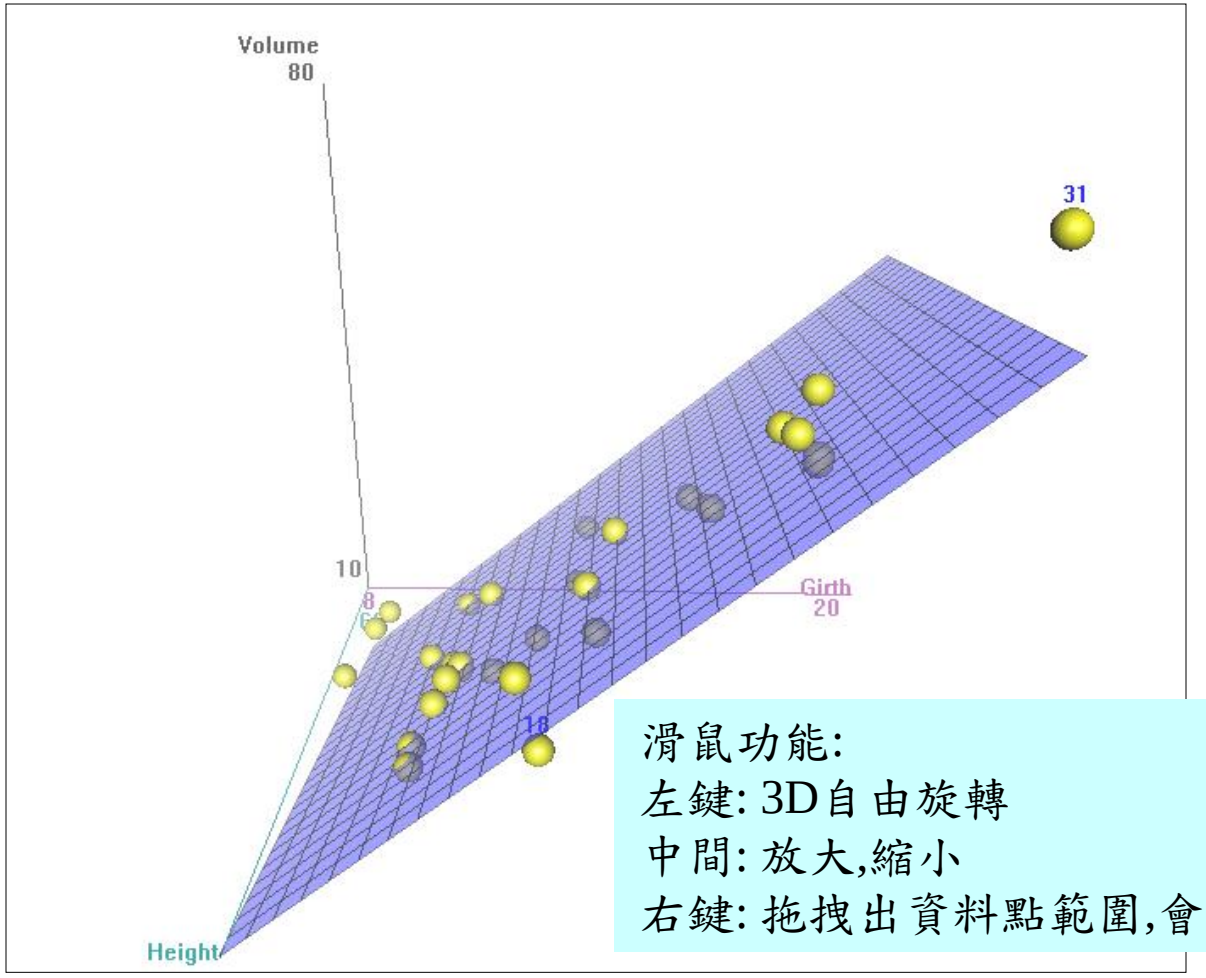


(3)



	Girth	Height	Volume
1	8.3	70	10.3
2	8.6	65	10.3
3	8.8	63	10.2
4	10.5	72	16.4
5	10.7	81	18.8
6	10.8	83	19.7
7	11.0	66	15.6
8	11.0	75	18.2

3D plot (cont.)



滑鼠功能:
左鍵: 3D自由旋轉
中間: 放大,縮小
右鍵: 拖拽出資料點範圍,會顯示資料集編號(disabled)

Probability distribution

分配	R 分配名稱	參數
Beta	***beta***	shapel, shape2
Binomial	***binom***	size, prob
Cauchy	***cauchy***	location, scale
Chi-squared	***chisq***	df, ncp
Exponential	***exp***	rate
F	***f***	dfl, df2, ncp
Gamma	***gamma***	shape, rate
Geometric	***geom***	prob
Hypergeometric	***hyper***	m, n, k
Log-normal	***lnorm***	meanlog, sdlog
Loistic	***logis***	location, scale
Negative binomial	***nbinom***	size, prob
Normal	***norm***	mean, sd
Poisson	***pois***	lambda
Student's t	***t***	df, ncp
Uniform	***unif***	min, max
Weibull	***weibull***	shape, scale
Wilcoxon	***wilcox***	m, n

Probability function - d, p, q, r

- The standard distributions:
  - d:機率密度函數(Probability Density Functions, pdf or p.d.f.)
  - p:累積分配函數 (Cumulative distribution function, CDF)
  - q:百分比函數 (Quantile function)
  - r:隨機產生分配的資料

$$F(x) = P(X \leq x)$$

Normal distribution (mean, sd)

Function	Usage
Density function	dnorm(x, mean=0, sd=1, log = FALSE)
distribution function	pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
quantile function	qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
random generation	rnorm(n, mean=0, sd=1)

x, q: vector of quantiles.

p: vector of probabilities.

n: number of observations.

mean: vector of means.

sd: vector of standard deviations.

log, log.p logical: if TRUE, probabilities p are given as log(p).

lower.tail logical: if TRUE (default), probabilities are P[X <= x],
otherwise, P[X > x].

Sample : dnorm(), pnorm(), qnorm(), rnorm()

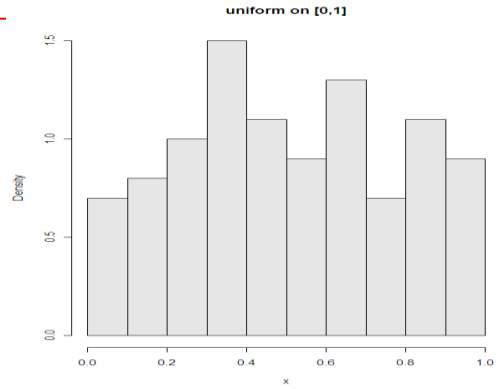
```
> # Probability function
> dnorm(1.96, 0, 1)
[1] 0.05844094
> pnorm(1.96, 0, 1)
[1] 0.9750021
> qnorm(0.975, 0, 1)
[1] 1.959964
> rnorm(5, 0, 1)
[1]  0.39845812 -1.88335789 -0.05334836  0.20127002  0.39038080
> # end
>
> dnorm(1.645)
[1] 0.1031108
> pnorm(1.645)
[1] 0.9500151
> pnorm(1.96)
[1] 0.9750021
> pnorm(2)
[1] 0.9772499
> qnorm(0.95, 0, 1)
[1] 1.644854
> # end
```

Microsoft Excel 函數

	A	B
1	0.058440944	=NORMDIST(1.96,0,1,FALSE)
2	0.975002105	=NORMDIST(1.96,0,1,TRUE)
3	1.959963985	=NORMINV(0.975,0,1)

Random generation

```
> # Random generation
> # also runif(1,min=0,max=2)
> runif(1,0,2)
[1] 1.854446
>
> runif(5,0,2)
[1] 1.7833049 1.6758303 1.4697508 1.0046285 0.3021581
>
> # 5 random numbers in [0,1]
> runif(5)
[1] 0.1896657 0.8363092 0.9523614 0.8769163 0.2453059
>
> x <- runif(100) # get the random numbers U(0,1)
> hist(x,probability=TRUE,col=gray(.9),main="uniform on [0,1]")
> # end
```



Sample - binomial distribution

- 已知某產品之不良率為0.1,隨機抽取10個產品檢查,至多有3個產品為不良品的機率為何?

Analysis:

方法1: 直接計算

$$P(X \le 3) = \sum_{i=0}^{3} f(x) = \sum_{x=0}^{3} C_x^{10} (0.1)^x (0.9)^{10-x}$$, 查表可得0.9872

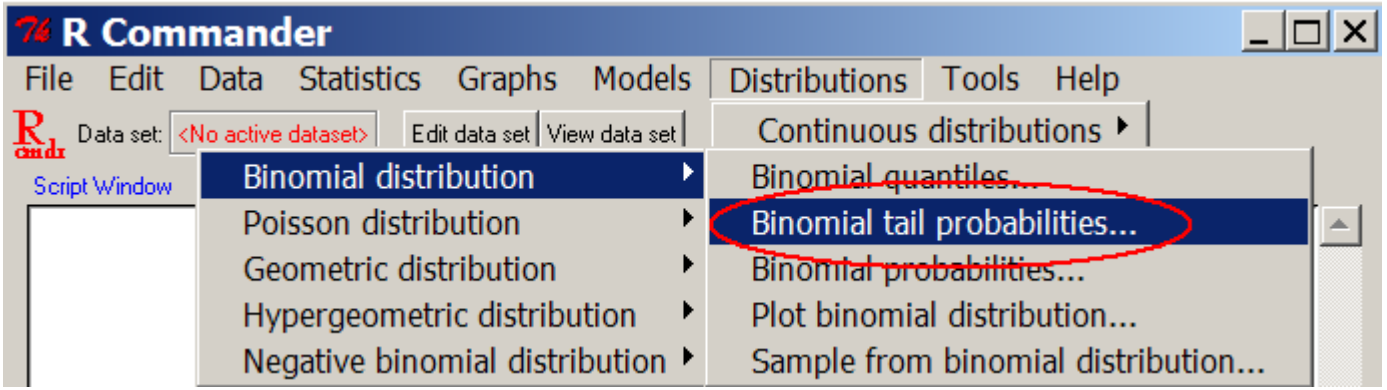
方法2: R

```
> # Sample - binomial distribution
> pbinom(3, 10, 0.1)
[1] 0.9872048
> # end
```

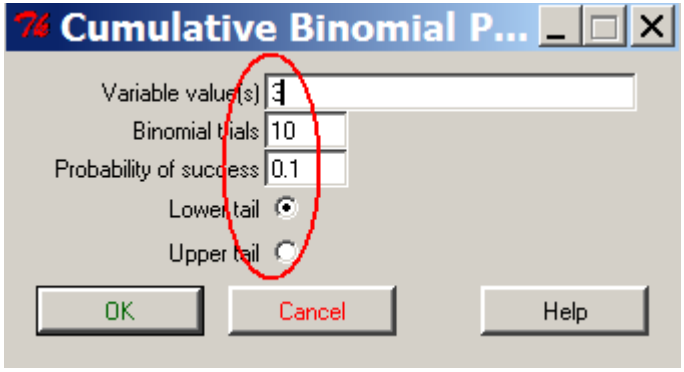
Sample - binomial distribution (cont.)

方法3: R Commander

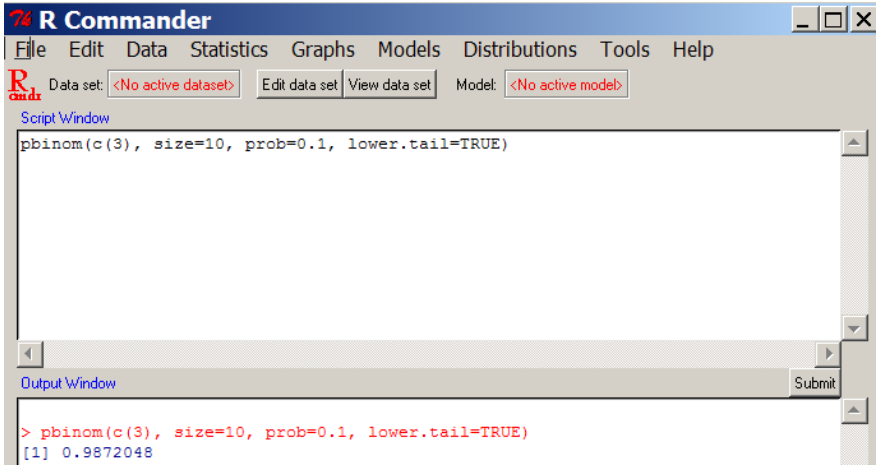
(1)



(2)



(3)

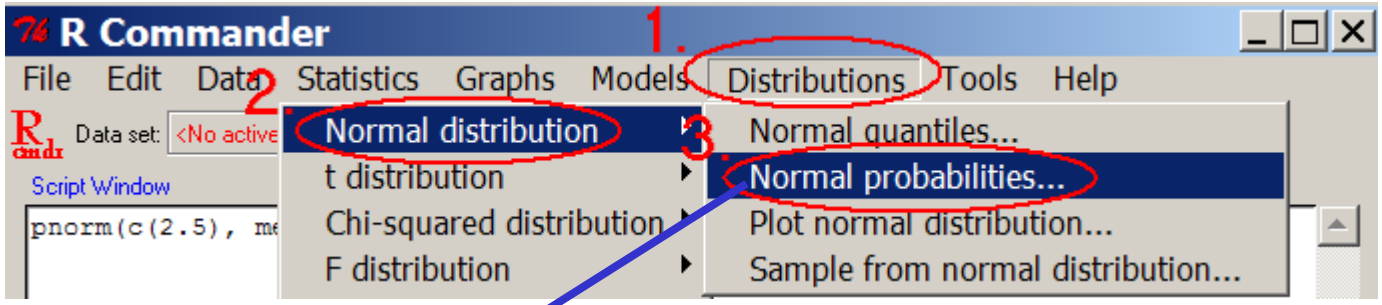


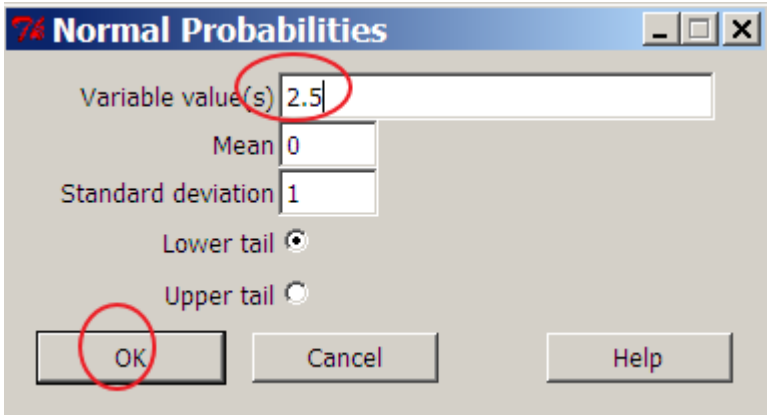
Sample – normal distribution

若 $R.V.\ Z \sim N(0,1)$，求(1)$P(Z \le 2.5) = ?$ ，$P(Z \le 2.41) = ?$

(2) $P(-2 \le Z \le 3) = ?$

(3)求常數 a ，使得$P(Z \le a) = 0.95$。






```
> # Sample - normal distribution
> pnorm(c(2.5), mean=0, sd=1, lower.tail=TRUE)
[1] 0.9937903
> # end
```

TRY !

(1) P(Z≦2.41)=?

(2) P(-2 ≦ Z ≦ 3)=?

(3) Find the constant *a*.

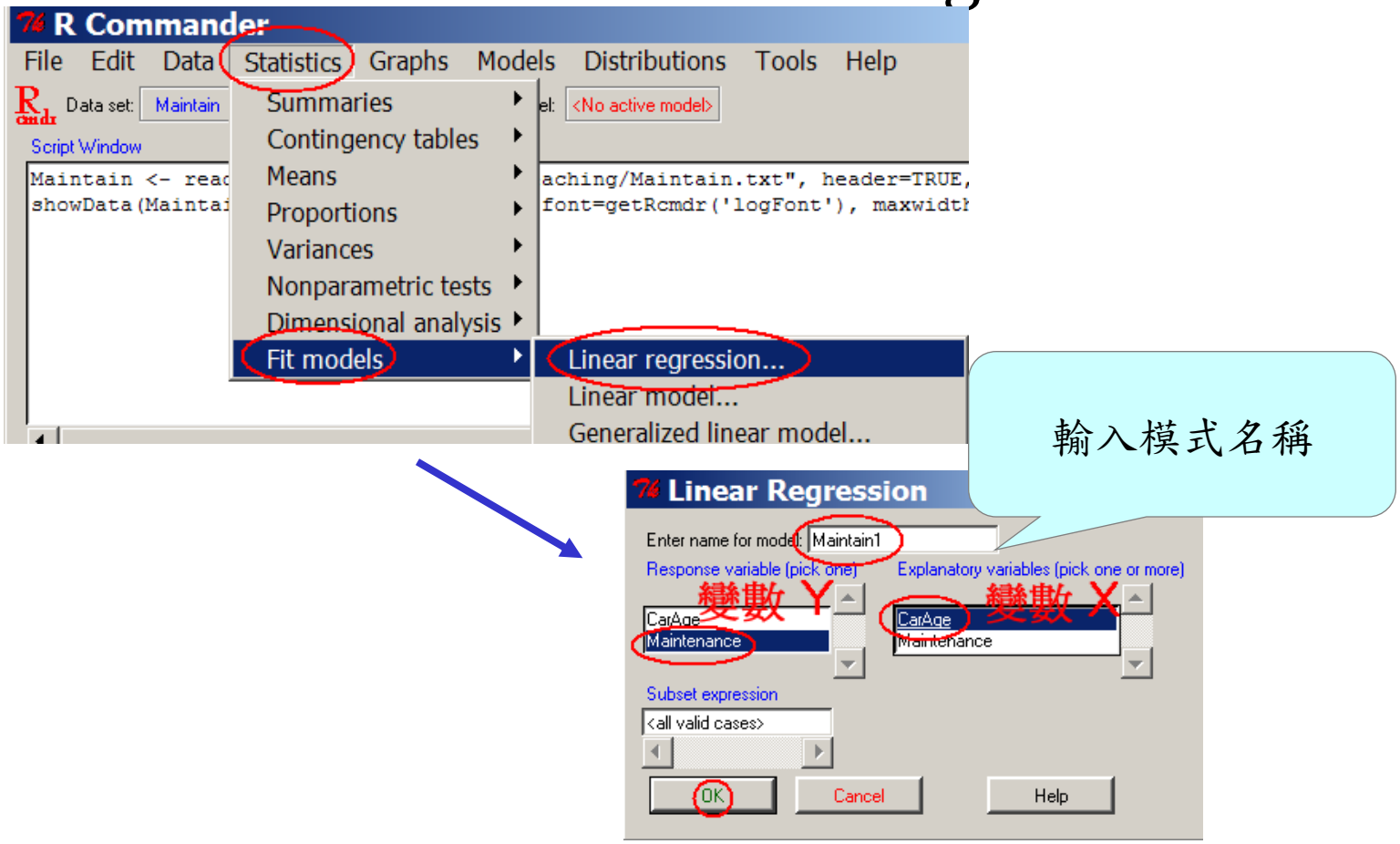
Regression analysis

- 考慮某品牌汽車之車齡與保養費用資料如下:

CarAge	1	2	2	3	3	4	4	5
Maintenance	6500	12000	13000	15000	20000	20000	25000	30000

Q: Find $\hat{y} = \hat{\alpha} + \hat{\beta}x, x$：車齡 , y：保養費用

- Statistics\Fit models\Linear regression



Output

```
> Maintain1 <- lm(Maintenance~CarAge, data=Maintain)
> summary(Maintain1)
```

Output Window

```
Call:
lm(formula = Maintenance ~ CarAge, data = Maintain)

Residuals:
    Min      1Q  Median      3Q     Max
-3270.8  -750.0   437.5  1291.7  2312.5

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    937.5     2034.6   0.461 0.661184
CarAge        5583.3      627.9   8.892 0.000113 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2175 on 6 degrees of freedom
Multiple R-Squared: 0.9295,     Adjusted R-squared: 0.9177
F-statistic: 79.07 on 1 and 6 DF,  p-value: 0.0001127
```

$$\hat{y} = 937.5 + 5583.3x,$$

$$\hat{\alpha} = 937.5$$

$$\hat{\beta} = 5583.3$$

$$R^2 = 0.9295$$

- R^2:判定係數, coefficient of determination

Linear model

```
> names(Maintain1)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

> Maintain1$coefficient
(Intercept)      CarAge
    937.500    5583.333

> Maintain1$fitted.values
       1          2          3          4          5          6          7          8
6520.833 12104.167 12104.167 17687.500 17687.500 23270.833 23270.833 28854.167
```

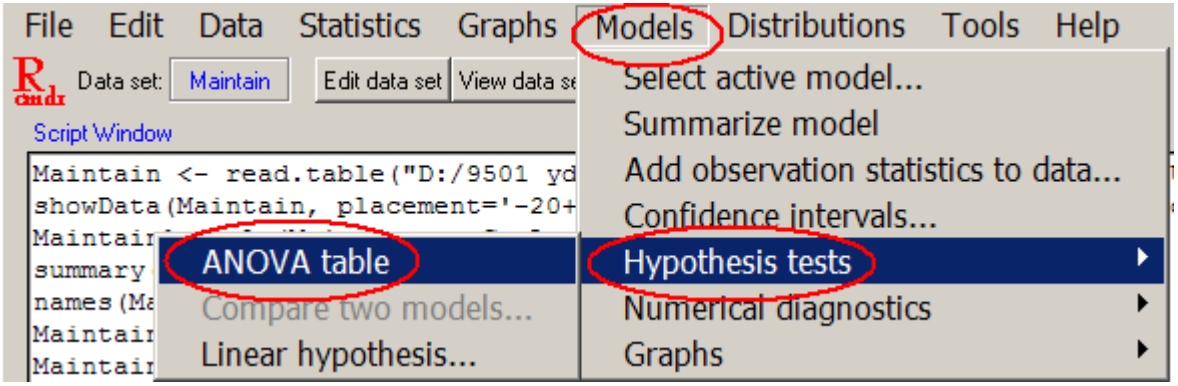
- names: 物件之內容
- Maintain1$coefficient: Maintain1物件中的 coefficient 值
- Maintain1$fitted.values: 迴歸模型的預測值

Model fit

- 利用ANOVA

表 13.3　檢定迴歸模型之變異數分析表

變異來源	平方和	自由度	均方	f 值
迴歸模型	SSR	1	$MSR=\frac{SSR}{1}$	$f_0=\frac{SSR}{S^2}$
隨機誤差	SSE	$n-2$	$MSE=S^2=\frac{SSE}{n-2}$	
總和	SST	$n-1$		



Model fit (cont.)

H_0:此模型不具解釋能力($\beta=0$)
H_1:此模型具解釋能力

```
> qf(c(0.05), df1=1, df2=6, lower.tail=FALSE)
[1] 5.987378
```

```
> Anova(Maintain1)
Anova Table (Type II tests)

Response: Maintenance
             Sum Sq Df F value    Pr(>F)
CarAge    374083333  1  79.072 0.0001127 ***
Residuals  28385417  6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

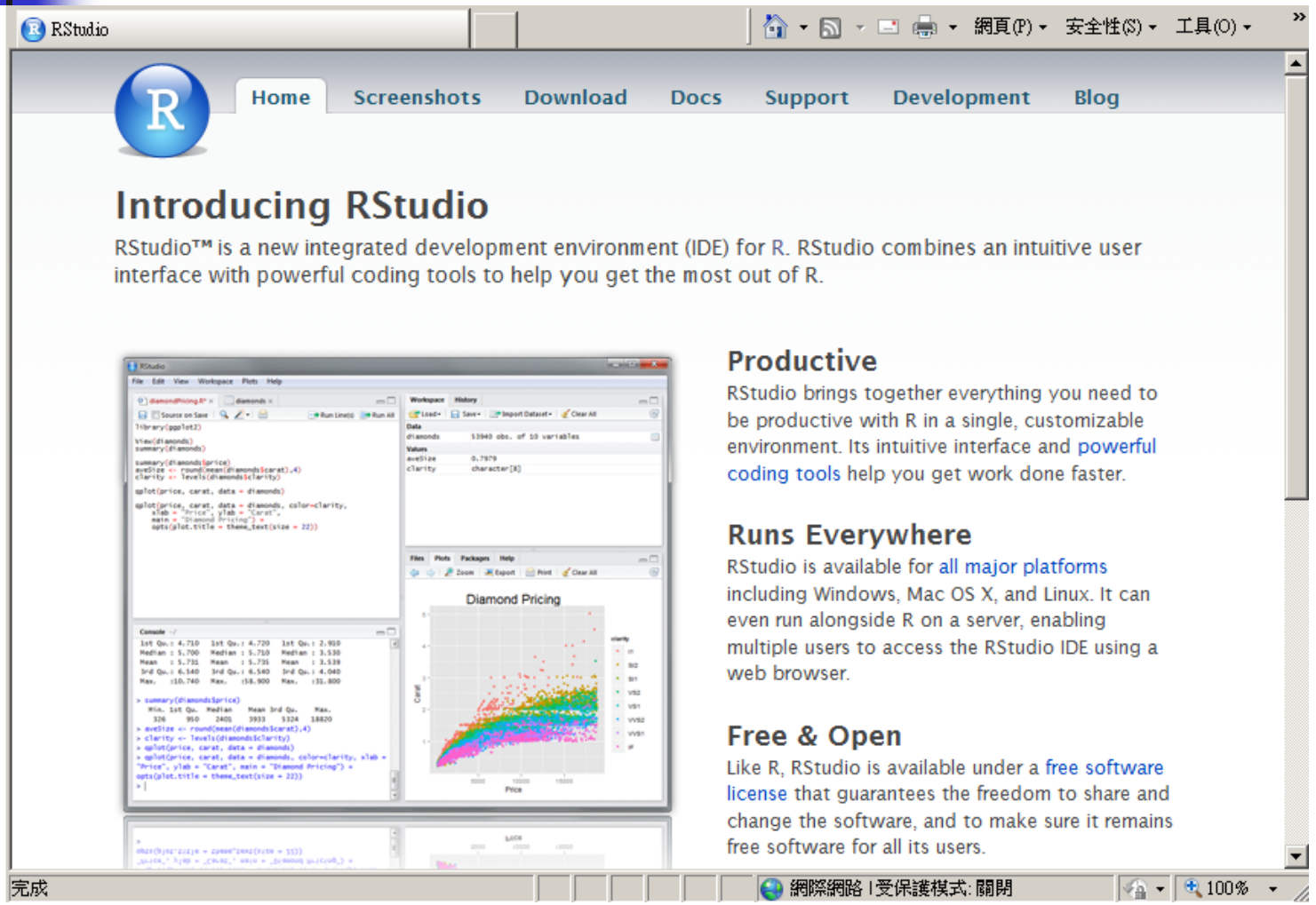
註: 查表 $F_{0.05}(1,6)=5.9874$

因 79.072 > 5.987378 , 所以拒絕 H_0, 即此迴歸模型具有解釋能力
由 *p*- value (0.0001127 很小)可直接觀察 reject H_0.

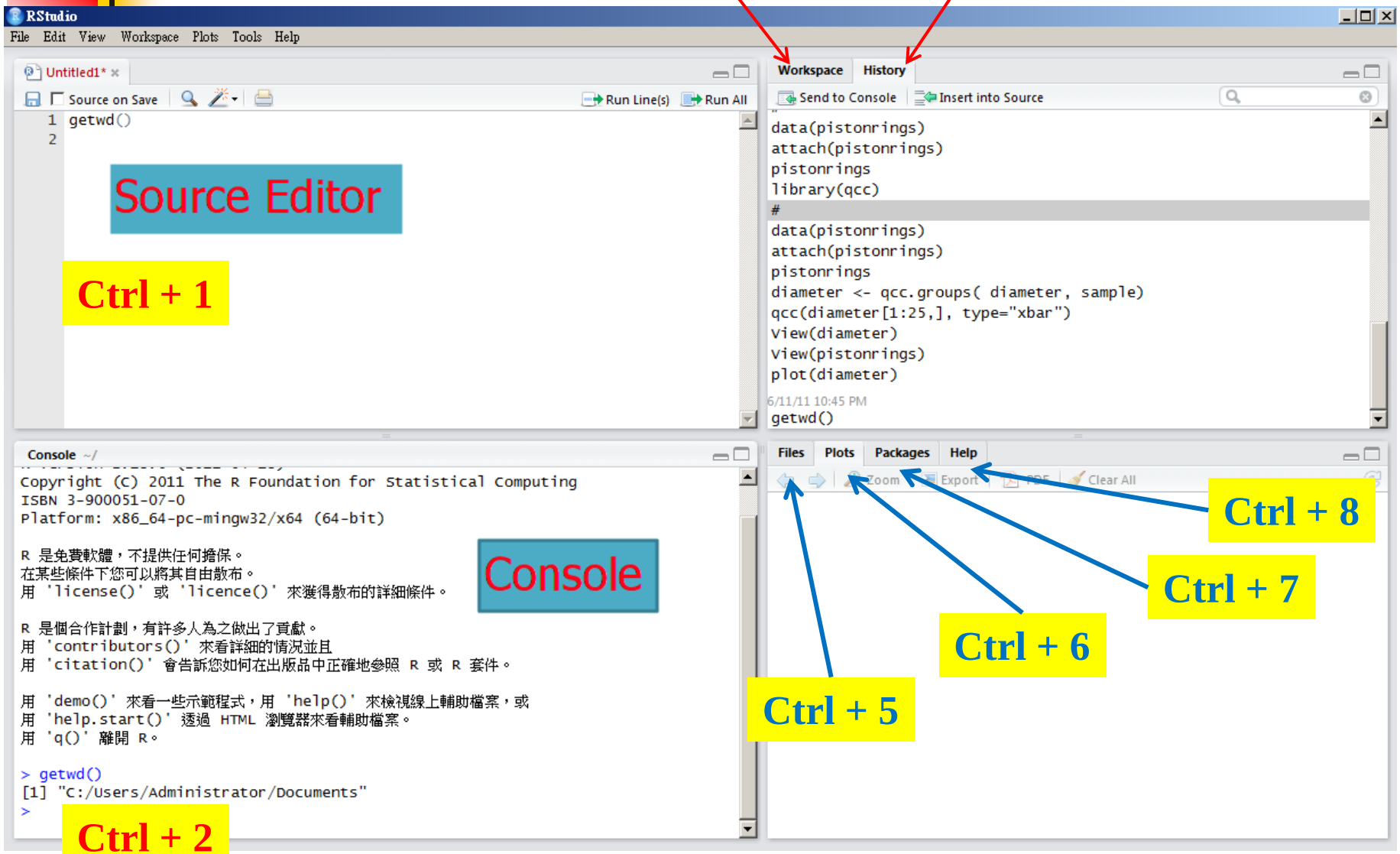
5.2 RStudio

- RStudio is a new integrated development environment (IDE) for R.
- RStudio combines an intuitive user interface with powerful coding tools to help you get the most out of R.
- RStudio can run alongside R on a server, enabling multiple users to access the RStudio IDE using a web browser.

http://www.rstudio.org/

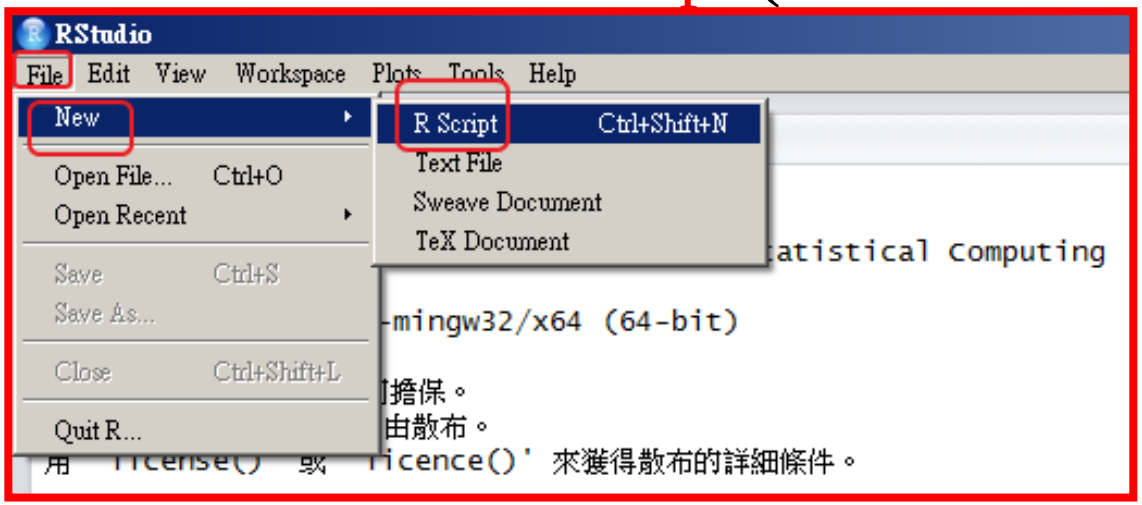


Download RStudio

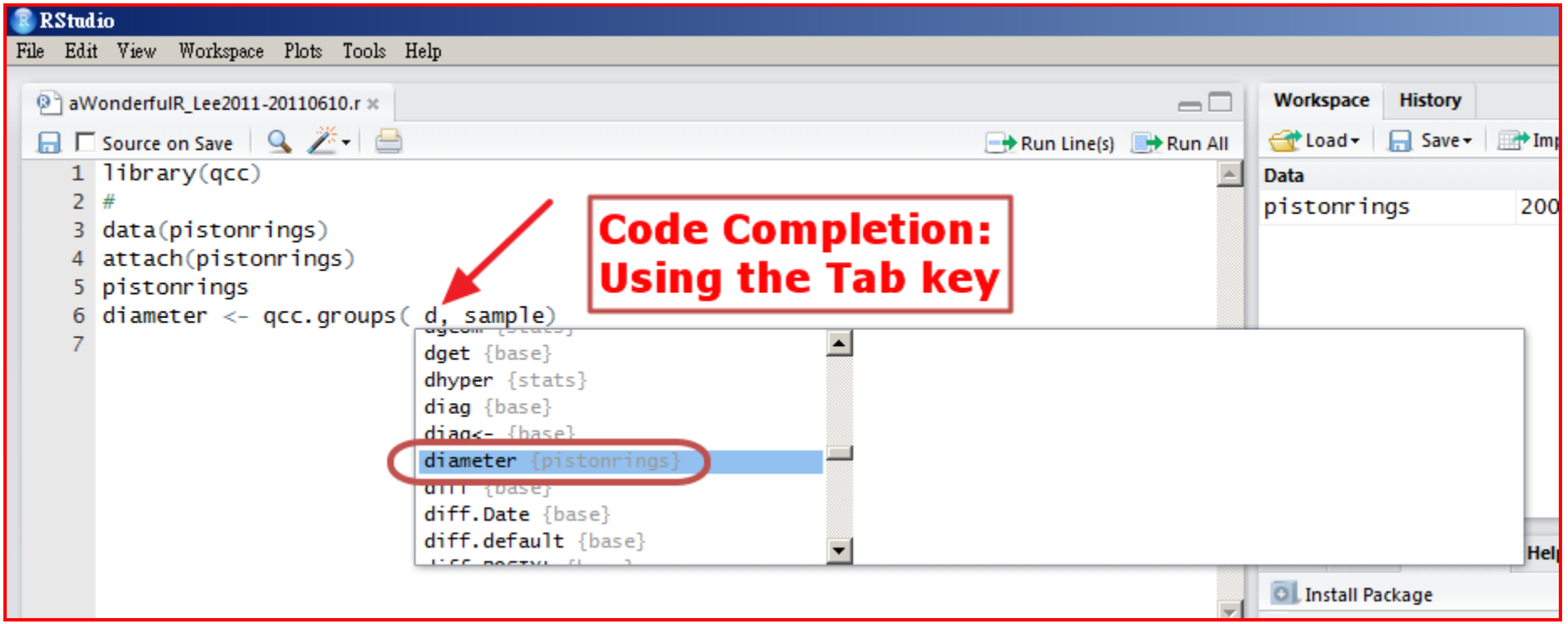

Ctrl + 3
Ctrl + 4
RStudio
File Edit View Workspace Plots Tools Help
Untitled1*
Source on Save
Run Line(s)
Run All
1 getwd()
2
Source Editor
Ctrl + 1
Workspace
History
Send to Console
Insert into Source
data(pistonrings)
attach(pistonrings)
pistonrings
library(qcc)
#
data(pistonrings)
attach(pistonrings)
pistonrings
diameter <- qcc.groups(diameter, sample)
qcc(diameter[1:25,], type="xbar")
view(diameter)
view(pistonrings)
plot(diameter)
6/11/11 10:45 PM
getwd()
Console ~/
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)
R 是免費軟體，不提供任何擔保。
在某些條件下您可以將其自由散布。
用 'license()' 或 'licence()' 來獲得散布的詳細條件。
Console
R 是個合作計劃，有許多人為之做出了貢獻。
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。
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。
> getwd()
[1] "C:/Users/Administrator/Documents"
>
Ctrl + 2
Files
Plots
Packages
Help
Zoom
Export
Clear All
Ctrl + 8
Ctrl + 7
Ctrl + 6
Ctrl + 5

Rstudio Features

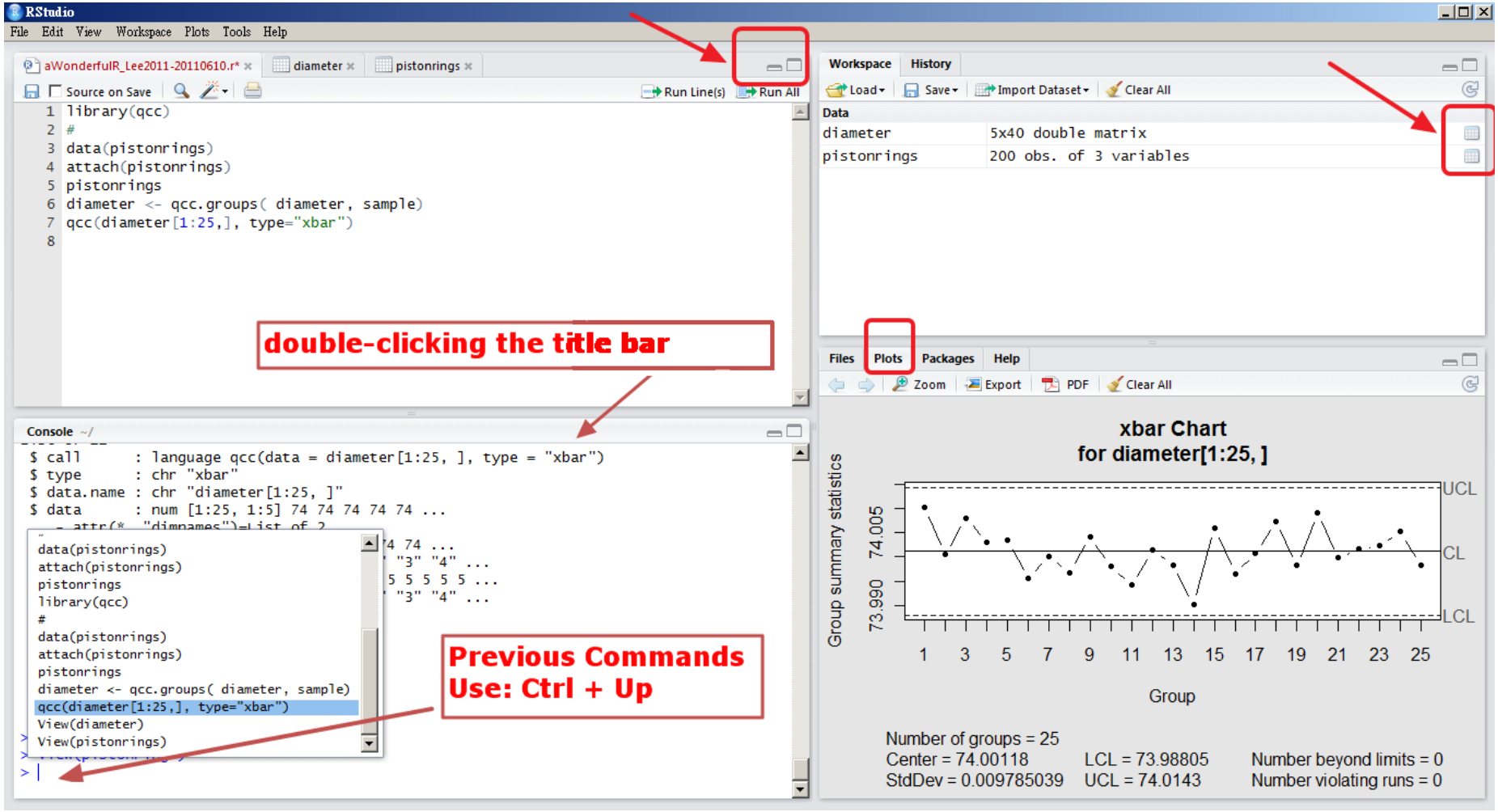
- Code completion
  - Tab
- Retrieving Previous Commands
  - Ctrl + Up
- File → New → R Script (Ctrl +Shift+N)



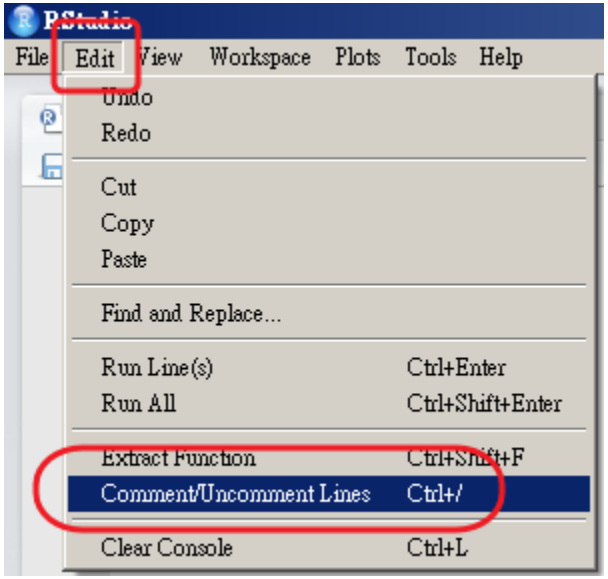
Code completion

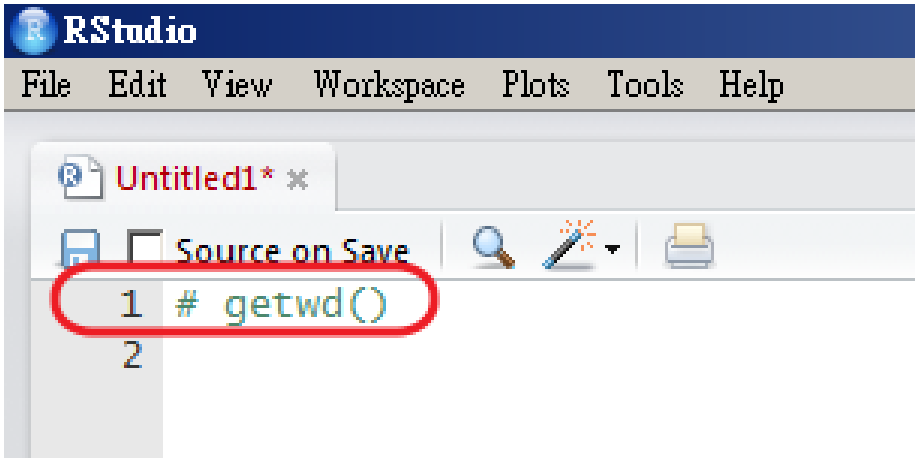


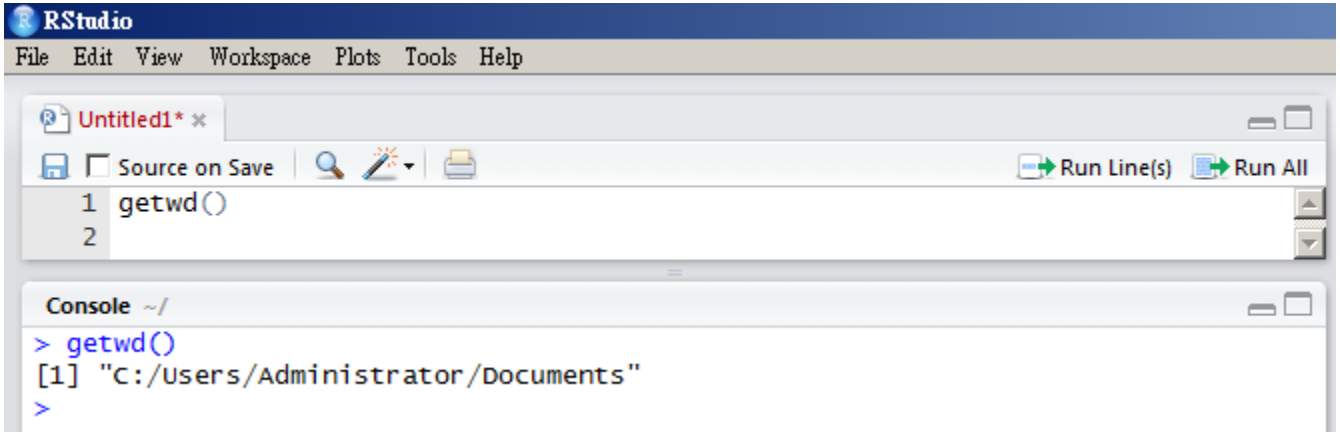
Command history



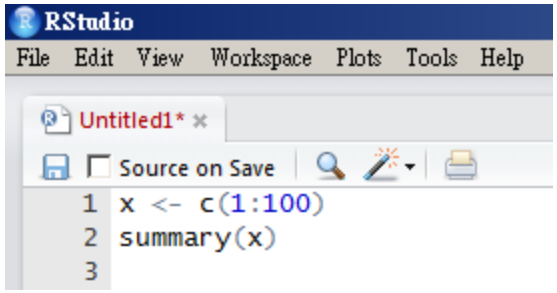
Comments/ Uncomments

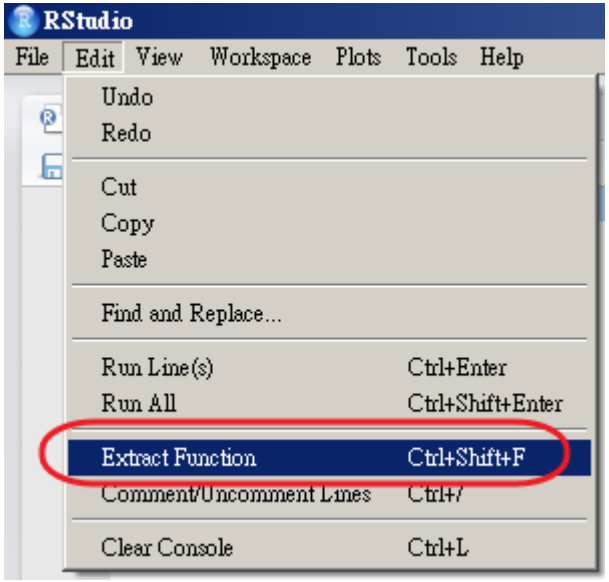


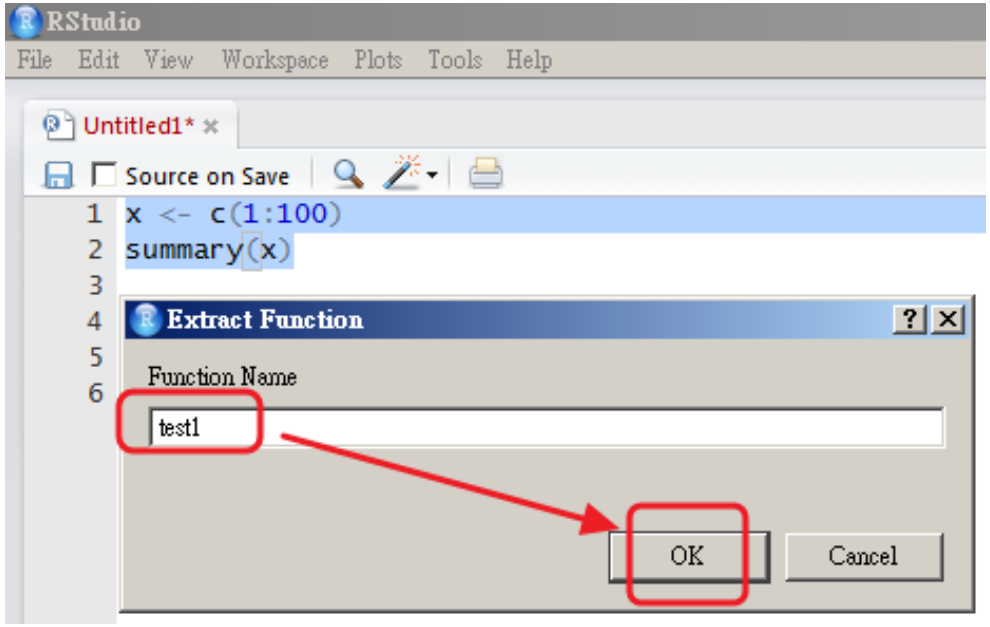




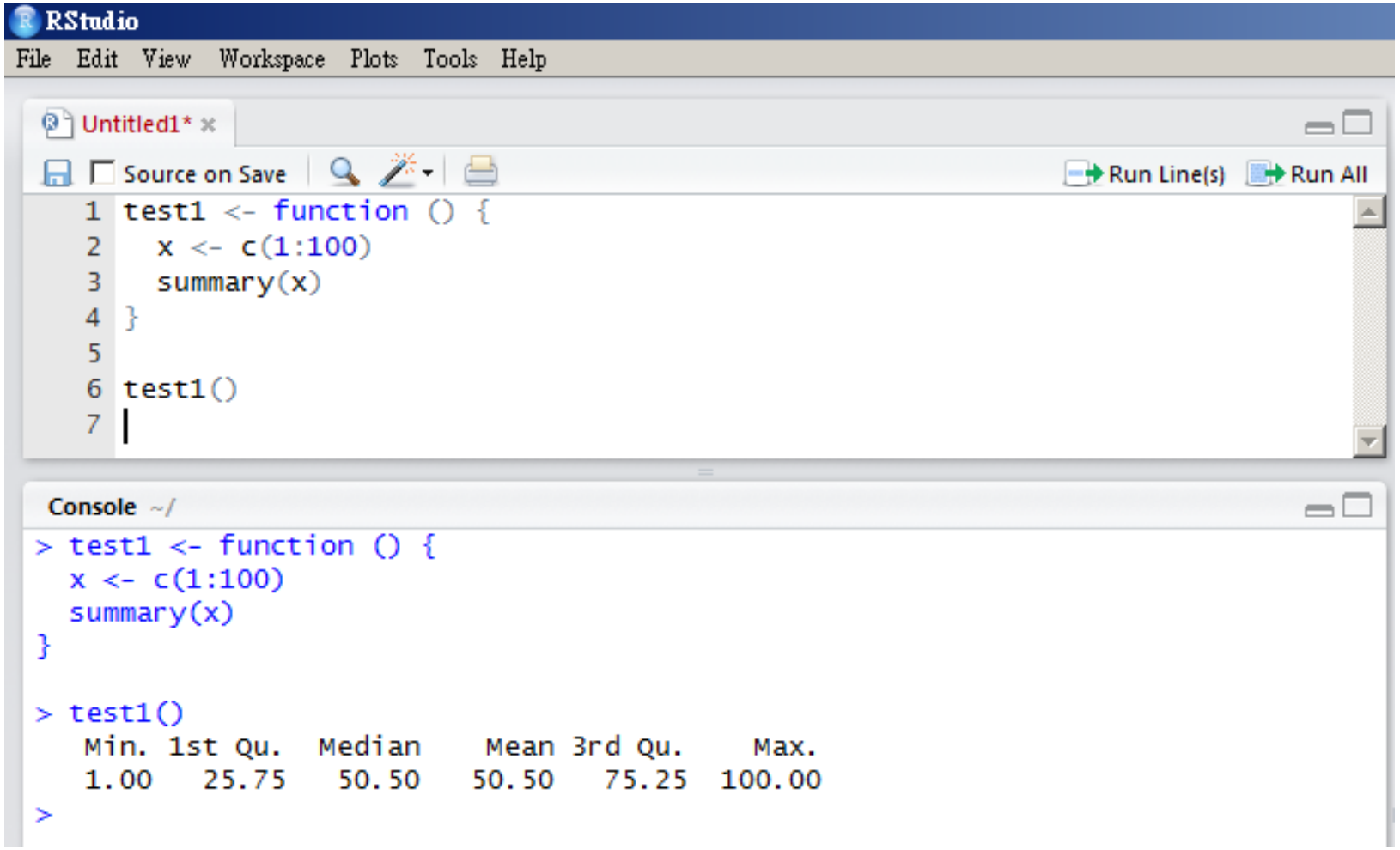
Extract function





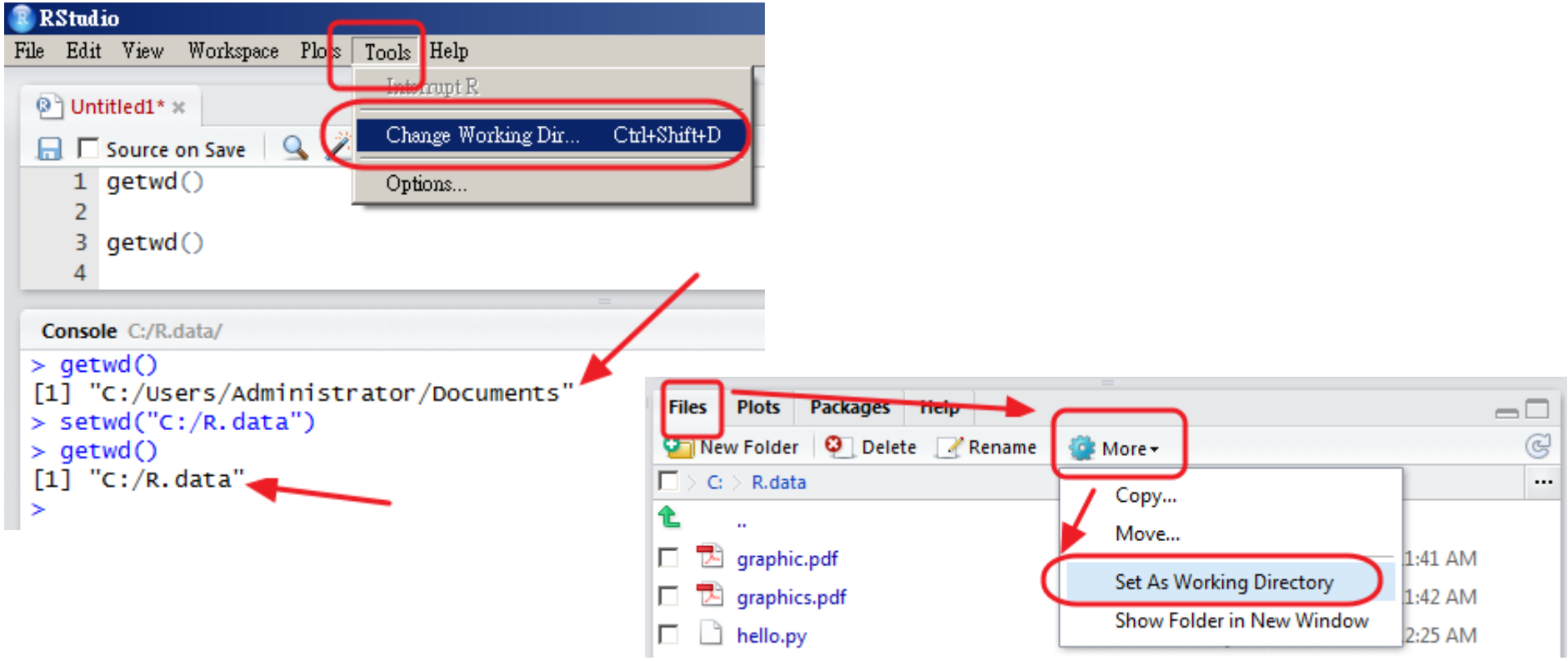


Extract function - output

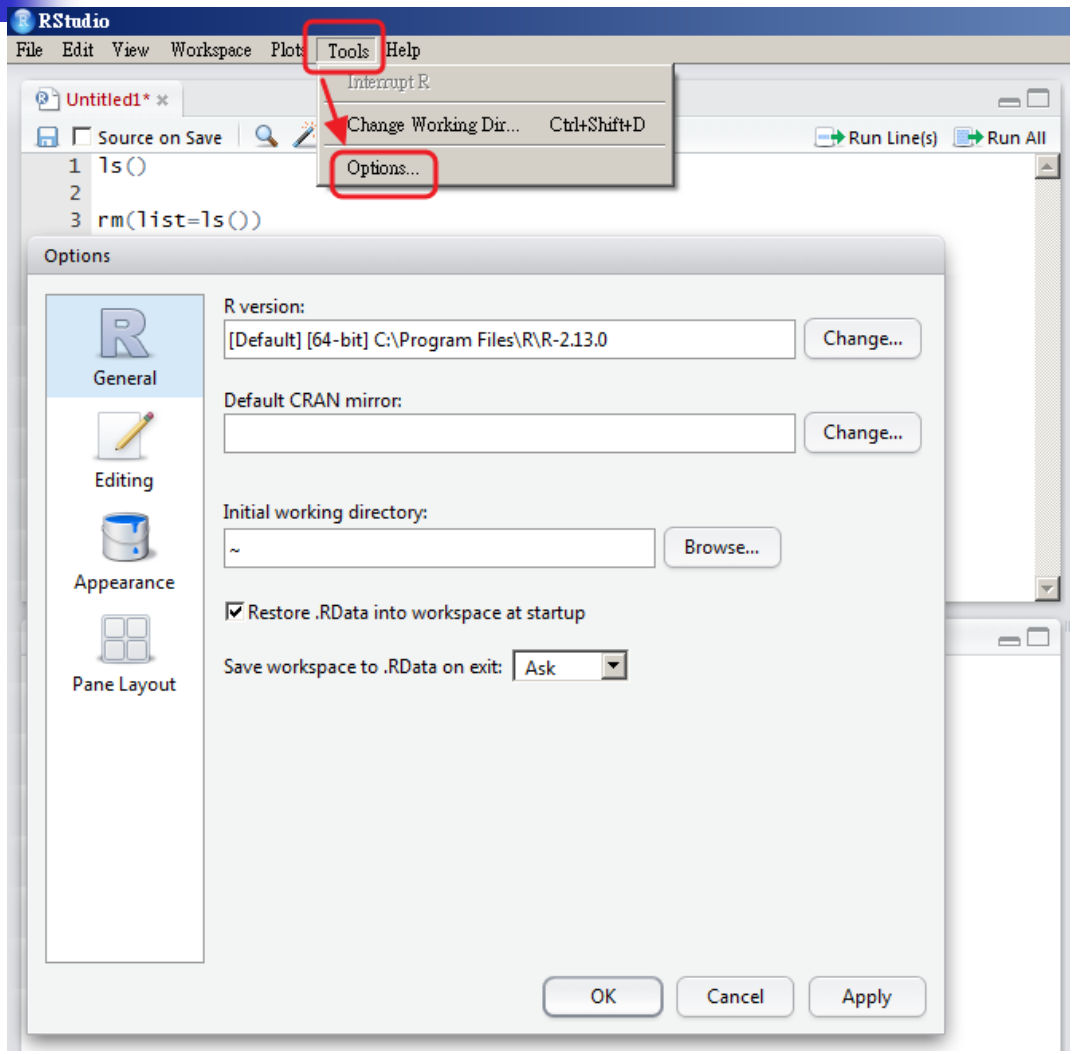


Change Working Directory

- Tools → Change Working Dir…
- setwd("c:/R.data")



RStudio Options



5.3 Quality Control Chart

- 計量值管制 (Control Charts for Variables)

Control Limits for the $\bar{x}$ Chart

$$\begin{aligned} \text{UCL} &= \bar{\bar{x}} + A_2\bar{R} \\ \text{Center line} &= \bar{\bar{x}} \\ \text{LCL} &= \bar{\bar{x}} - A_2\bar{R} \end{aligned} \qquad (5\text{-}4)$$

The constant A_2 is tabulated for various sample sizes in Appendix Table VI.

Control Limits for the *R* Chart

$$\begin{aligned} \text{UCL} &= D_4\bar{R} \\ \text{Center line} &= \bar{R} \\ \text{LCL} &= D_3\bar{R} \end{aligned} \qquad (5\text{-}5)$$

The constants D_3 and D_4 are tabulated for various values of n in Appendix Table VI.

Reference：Montgomery, D. C., Introduction to STATISTICAL QUALITY CONTROL 5e, John Wiley & Sons, Inc, 2005.

Control Charts for Variables

- A_2

$$W = \frac{R}{\sigma}, E(W) = d_2$$

$$\hat{\sigma} = \frac{\bar{R}}{d_2}$$

$$\because \ \sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}, \qquad \sigma = \frac{\bar{R}}{d_2}$$

$$\therefore \ \bar{x} \pm 3\sigma_{\bar{x}} = \bar{x} \pm \frac{3\sigma}{\sqrt{n}} = \bar{x} \pm \frac{3\bar{R}}{d_2\sqrt{n}} = \bar{x} \pm A_2\bar{R}$$

$$\Rightarrow A_2 = \frac{3}{d_2\sqrt{n}}$$

- D_3 、D_4

$$\because \ \sigma_R = d_3\sigma, \qquad \sigma = \frac{\bar{R}}{d_2}$$

$$UCL_R = \bar{R} + 3\sigma_R = \bar{R} + 3d_3\sigma = \bar{R} + \frac{3d_3\bar{R}}{d_2} = (1 + \frac{3d_3}{d_2})\bar{R}$$

$$LCL_R = \bar{R} - 3\sigma_R = \bar{R} - 3d_3\sigma = \bar{R} - \frac{3d_3\bar{R}}{d_2} = (1 - \frac{3d_3}{d_2})\bar{R}$$

$$\Rightarrow D_3 = 1 + \frac{3d_3}{d_2} \qquad D_4 = 1 - \frac{3d_3}{d_2}$$

Sample: x-bar chart

	A	B	C	D
1	No	x1	x2	x3
2	1	0.0629	0.0636	0.064
3	2	0.063	0.0631	0.0622
4	3	0.0628	0.0631	0.0633
5	4	0.0634	0.063	0.0631
6	5	0.0619	0.0628	0.063
7	6	0.0613	0.0629	0.0634
8	7	0.063	0.0639	0.0625
9	8	0.0628	0.0627	0.0622
10	9	0.0623	0.0626	0.0633
11	10	0.0631	0.0631	0.0633
12	11	0.0635	0.063	0.0638
13	12	0.0623	0.063	0.063
14	13	0.0635	0.0631	0.063
15	14	0.0645	0.064	0.0631
16	15	0.0619	0.0644	0.0632
17	16	0.0631	0.0627	0.063
18	17	0.0616	0.0623	0.0631
19	18	0.063	0.063	0.0626
20	19	0.0636	0.0631	0.0629
21	20	0.064	0.0635	0.0629
22	21	0.0628	0.0625	0.0616
23	22	0.0615	0.0625	0.0619
24	23	0.063	0.0632	0.063
25	24	0.0635	0.0629	0.0635
26	25	0.0623	0.0629	0.063

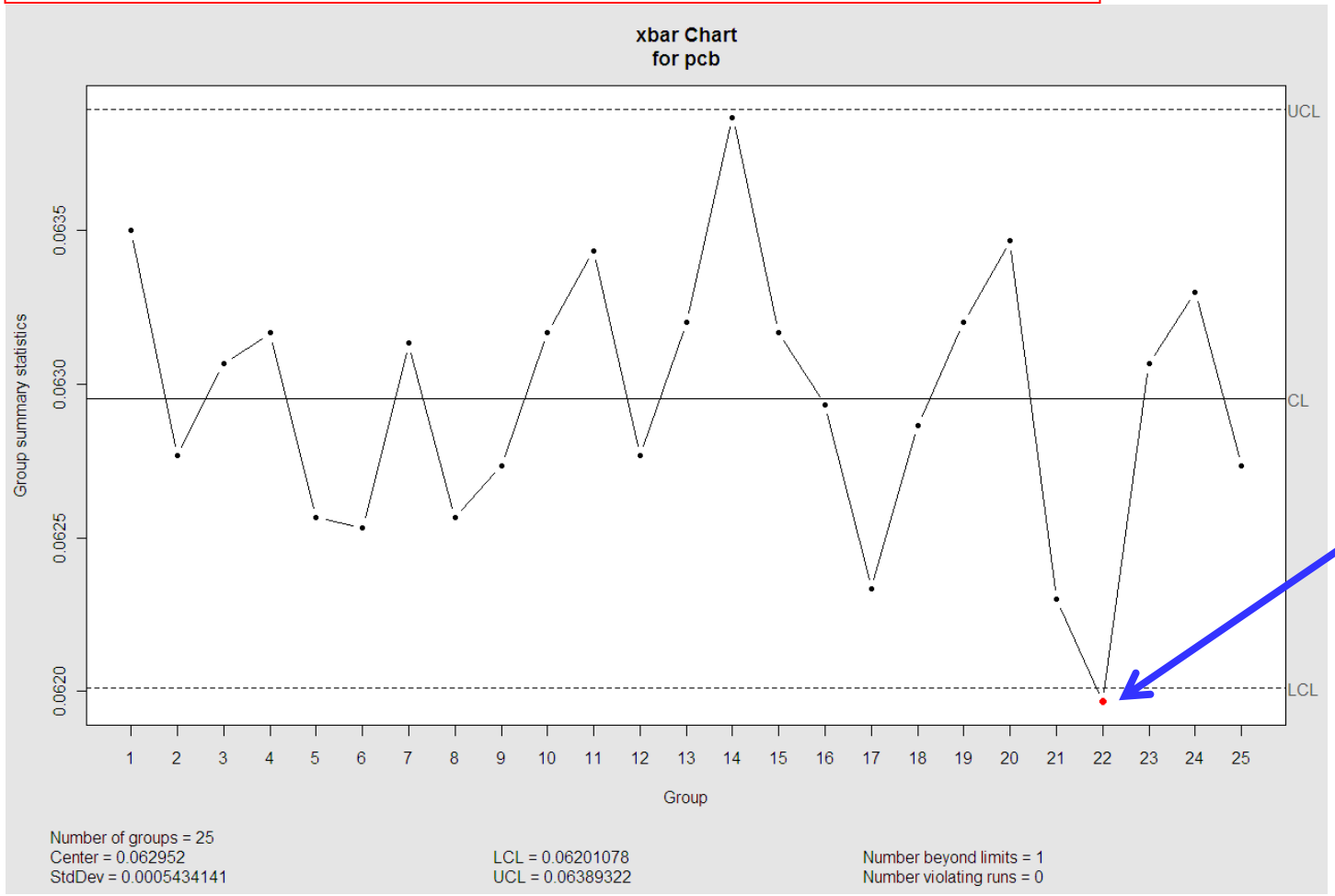
下載資料檔-
http://web.ydu.edu.tw/~alan9956/docu3/0992qm/hw1.zip

Sample: x-bar chart (cont.)

```
# Quality Control Chart
library(qcc)
workpath = "c:/R.data"
setwd(workpath)
# 先將資料複製到 C:\R.data 目錄
# 匯入資料
hw1 <- read.table("hw1.csv", header=TRUE, sep=",")
# 顯示資料
hw1

# 取出全部資料的第2-4欄
pcb <- hw1[, c(2:4)]; pcb

dim(pcb) # 包括 25列, 3行 資料

qcc(pcb, type="xbar") # 第22筆資料超出管制界限

qcc(pcb, type="R")    # 第15筆資料超出管制界限

pcb.modified <- pcb[-c(15,22), ] # 移除第15,22筆資料

dim(pcb.modified) # 包括 23列, 3行 資料

qcc(pcb.modified, type="xbar") # 製程有在管制界限之內嗎?

qcc(pcb.modified, type="R") # 製程有在管制界限之內嗎?
# end
```

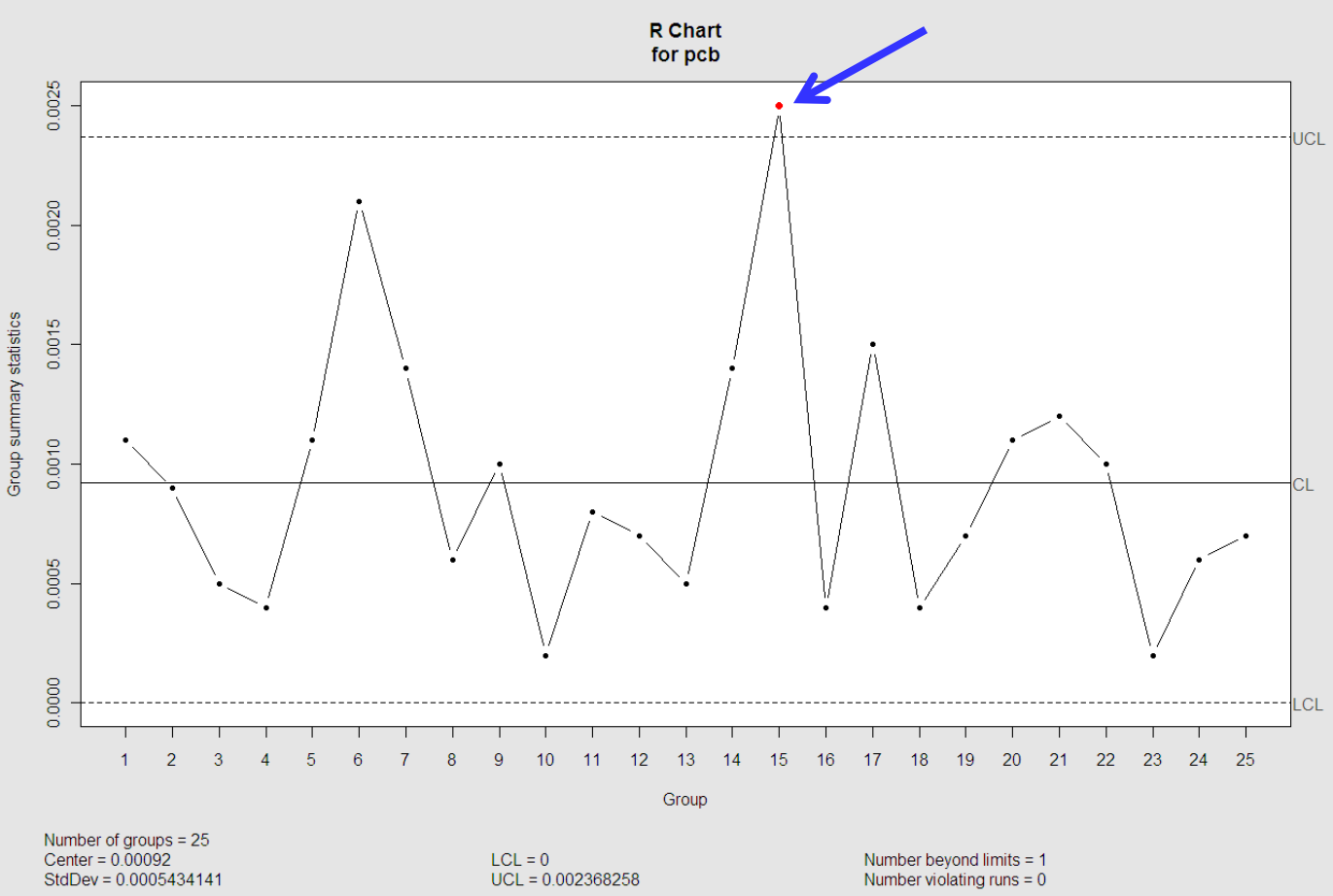
Sample: x-bar chart (cont.)

```
> qcc(pcb, type="xbar") # 第22筆資料超出管制界限
```



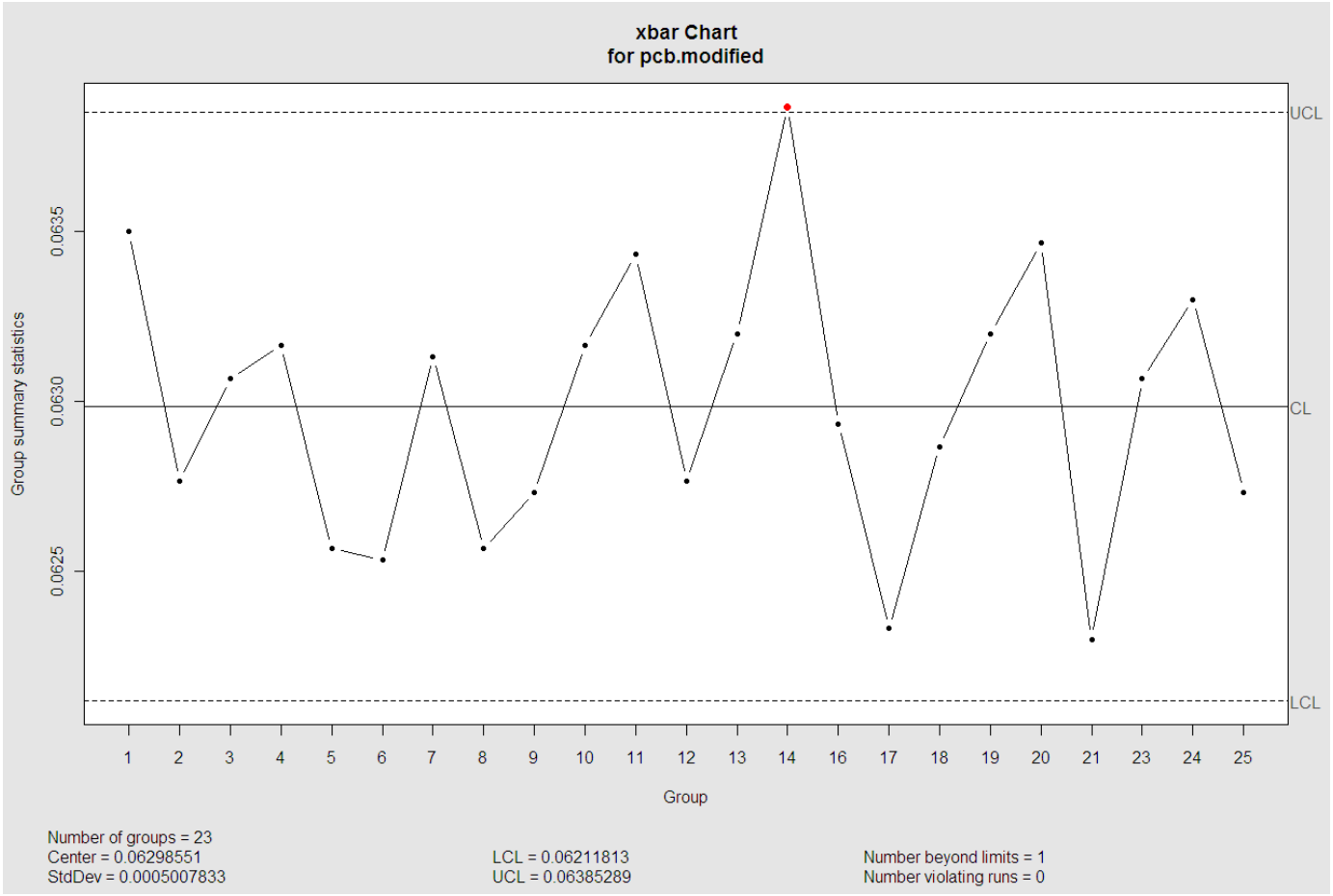
Sample: x-bar chart (cont.)

```
> qcc(pcb, type="R")      # 第15筆資料超出管制界限
```



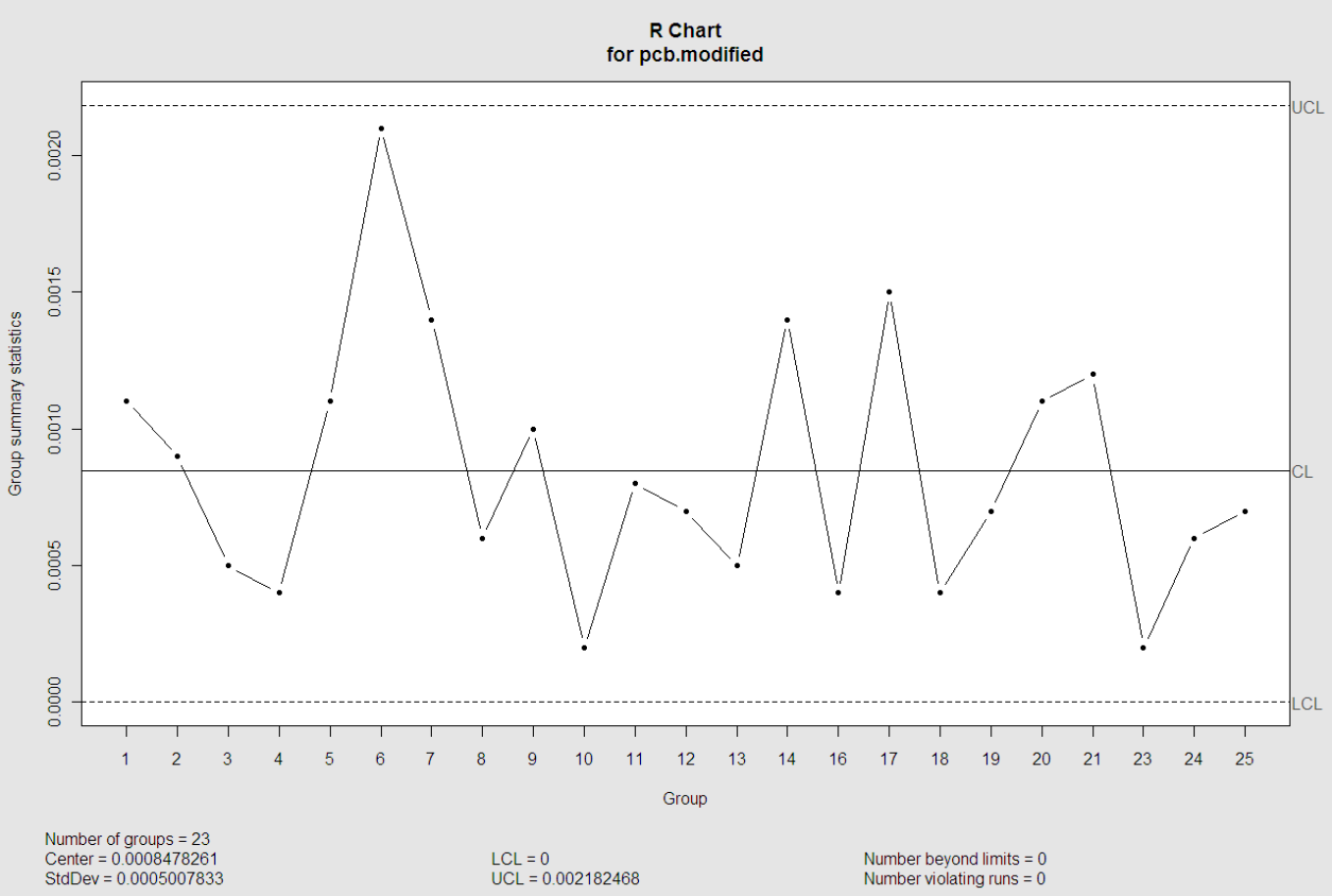
Sample: x-bar chart (cont.)

```
> pcb.modified <- pcb[-c(15,22), ] # 移除第15,22筆資料
```



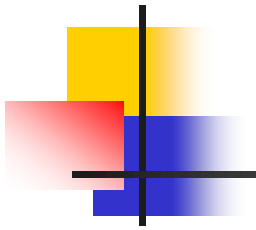
Sample: x-bar chart (cont.)

```
> pcb.modified <- pcb[-c(15,22), ] # 移除第15,22筆資料
```



5.4 SVM (Support Vector Machines)

- History :
  - Statistical Learning Theory (Vapnik 1998)
- Development:
  - Binary classification SVM
  - Multi-class SVM



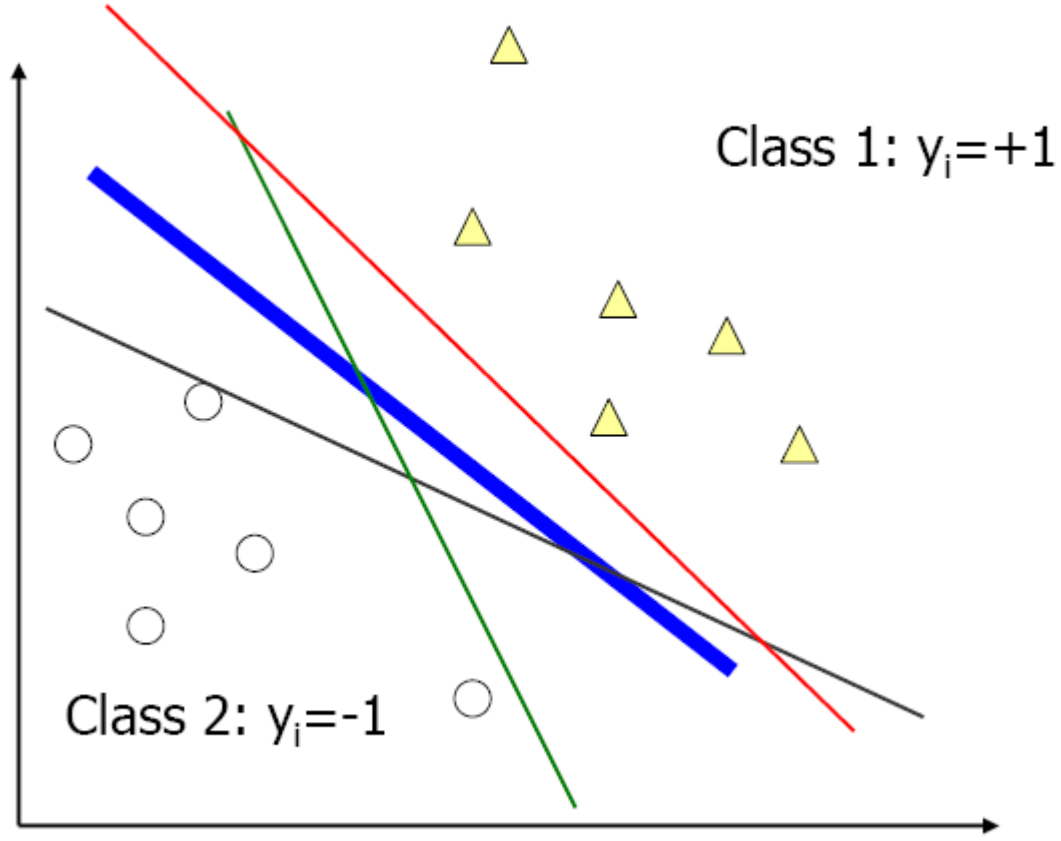
- Application:
  - Text categorization
  - Image recognition
  - Hand-written Digit Recognition
  - Bioinformatics
  - ……

Binary classification

- Consider a two-class, linearly separable classification problem :
  - training data (x_i, y_i), $i = 1,\ldots,l$. $\{x_i, y_i\}, x_i \in R^n$

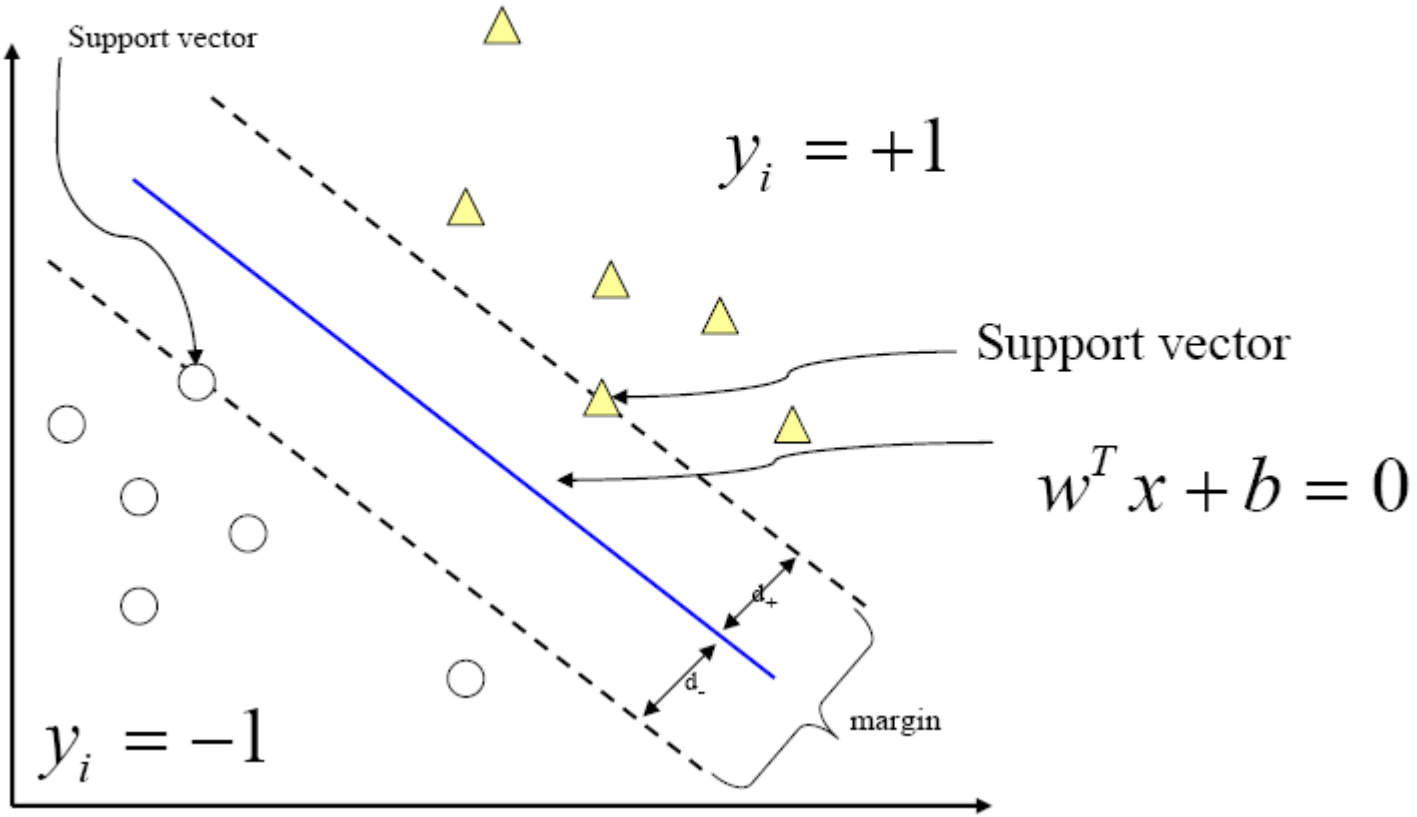
$$y_i = \begin{cases} +1, & x_i \in \text{class 1} \\ -1, & x_i \in \text{class 2} \end{cases}$$

Making decision boundary



Large-margin Decision Boundary

margin = (d+) 與(d-) → Goal: Max {margin}



Find maximum margin

$$w^T x_i + b \geq 1 \quad if \quad y_i = +1$$
$$w^T x_i + b \leq 1 \quad if \quad y_i = -1$$
$$\Rightarrow y_i (w^T x_i + b) \geq 1 \quad \forall i = 1, \ldots, l$$

$$margin = (d_+) + (d_-) = \frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|} = \frac{2}{\sqrt{w^T w}}$$

$$Minimize \qquad cost\ function, \Phi(w) = \frac{1}{2} w^T w$$

$$Subject\ to \qquad y_i (w^T x_i + b) \geq 1 \quad \forall i = 1, \ldots, l.$$

Lagrange multipliers function

$$\textit{Minimize} \qquad J(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{l} \alpha_i \left[y_i (w^T x_i + b) - 1 \right]$$

$$\textit{Subject to} \qquad \alpha_i \geq 0 \quad \forall i = 1,\ldots,l.$$

$$\frac{dJ(w,b,\alpha)}{dw} = 0, \quad w = \sum_{i=1}^{l} \alpha_i y_i x_i$$

$$\frac{dJ(w,b,\alpha)}{db} = 0, \quad \sum_{i=1}^{l} \alpha_i y_i = 0$$

Dual problem

$$J(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{l} \alpha_i y_i w^T x_i - b \sum_{i=1}^{l} \alpha_i y_i + \sum_{i=1}^{l} \alpha_i$$

$$w^T w = \sum_{i=1}^{l} \alpha_i y_i w^T x_i = \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Maximize $$Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Subject to

$$(1) \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$(2) \alpha_i \geq 0 \ \forall i = 1, 2, \ldots, l$$

Solution - Support vectors

- Many of the α_i are zero
- Support vectors (SV) x_i :
  - Non-zero α_i
  - The decision boundary is determined only by the SV
  - $$w = \sum_{i=1}^{l} \alpha_i y_i x_i$$
- Testing with a new data x_{new} :
  - Compute $w^T x_{new} + b$
  - If the result is positive , then classify x_{new} as class 1 , class 2 otherwise

Sample: SVM

```
> # The example of R for Support Vector Machines(SVM)
> # 安裝SVM 套件 e1071
> # 載入套件 e1071
> library(e1071)
> library(mlbench)
> # 載入資料集 Glass in mlbench package
> # 資料集 214 個觀測值,9 個變數,第9 個數數名稱爲Type,
> # 有7 個種類(1:7)
> data(Glass)
> head(Glass)
        RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
>
```

Sample: SVM (cont.)

```
> # 設定變數index 爲編號1,2,…214.
> index <- 1:nrow(Glass)
>
> # 準備隨機抽樣並設定測試資料的編號
> # 利用sample 取樣,將資料的1/3 做爲測試資料的序號
> testindex <- sample(index, trunc(length(index)/3))
> # 設定測試資料 testset,共71 筆資料
> testset <- Glass[testindex, ]
> # > dim(testset) # 可知道有71 筆測試資料
> # 將其他資料設定訓練資料 trainset,共143 個筆資料
> trainset <- Glass[-testindex, ]
```

Sample: SVM (cont.)

```
> # 利用svm 執行並將結果存入變數svm.model
> svm.model <- svm(Type ~ ., data = trainset, cost = 100, gamma = 1)
>
> # 利用predict 執行測試資料的分類預測
> svm.pred <- predict(svm.model, testset[, -10])
> print(svm.pred)
154 128  52 120  39 171  42  50   8 141 207 118 127  17  73 104  34
  3   2   2   2   1   2   1   1   1   2   7   2   3   1   2   2   1
 87  71  20  63  24  97 114 124  31 164  25  72  28 175 160 197 163
  2   2   2   1   1   2   2   2   1   2   1   2   1   2   3   7   2
 89  41 177 144 150  11  16  65  56 205 210  30 103  62 135
  2   1   6   2   1   1   1   1   2   7   7   1   2   2   1
Levels: 1 2 3 5 6 7
>
> # 利用 write 輸出結果
> # 將結果輸出成CSV 檔案
> write.table(svm.pred, file = "svm.test.csv", sep = ",")
> # end
```

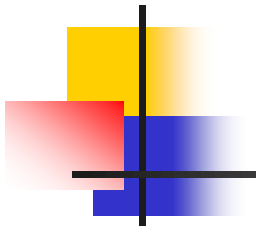
Sample: SVM (cont.)

svm.test.csv

	A	B
1	x	
2	144	2
3	145	1
4	127	3
5	193	7
6	45	2
7	55	1
8	140	2
9	161	3
10	182	2
11	68	1

References

- Adler, J., *R in a Nutshell*, O'Reilly Media, Inc., 2010.
- Fox, J., *The R Commander: A Basic-Statistics GUI for R*, 2006, URL: http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/
- Ihaka, R., Gentleman, R., *R: a language for data analysis and graphics*, Journal of Computational and Graphical Statistics 5: 299 -314, 1996.
- Maindonald, J., Braun, W.J., *Data Analysis and Graphics Using R – an Example-Based Approach, Third Edition*, Cambridge University Press, New York, 2010.
- Paradis, E., *R for Beginners*, 2002, URL: http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
- The R Project for Statistical Computing, URL: http://www.r-project.org/
- Venables, W. N., Smith, D.M., R Development Core Team, *An Introduction to R - Notes on R: A Programming Environment for Data Analysis and Graphics*, 2006, URL: http://cran.r-project.org/doc/manuals/R-intro.pdf
- Venables, W. N., Ripley, B.D., *Modern Applied Statistics with S*, Springer, 2010.
- *WEPA site*, 2011, URL: http://web.ydu.edu.tw/~alan9956/



THANKS

Q & A