Carnegie Mellon University

MSE

Data Analytics for Materials Science

A.D. (Tony) Rollett

& unknown author "TB"

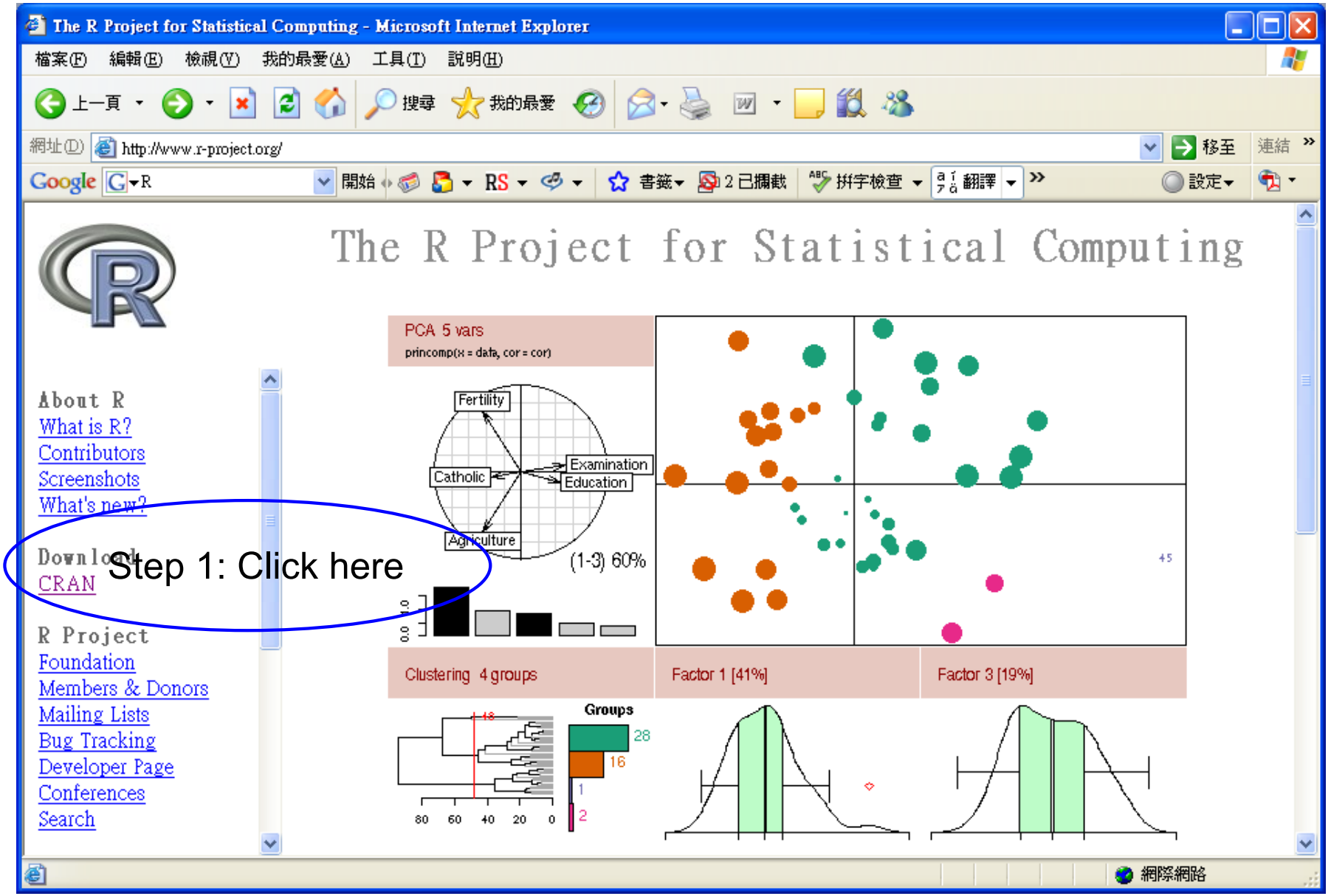
Dept. Materials Sci. Eng.
Carnegie Mellon University

Introduction to R

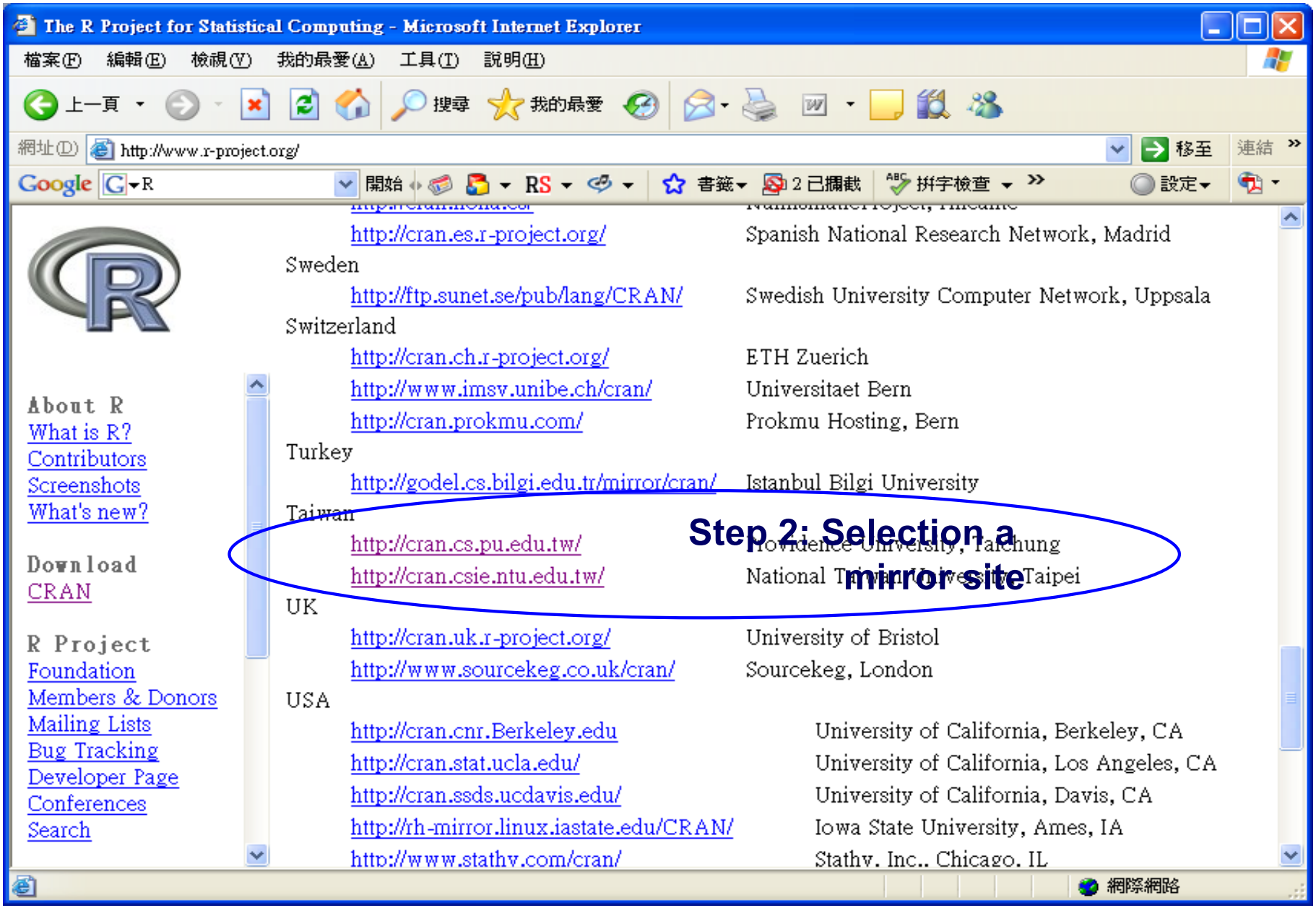
Revised: 31st Jan. 2021

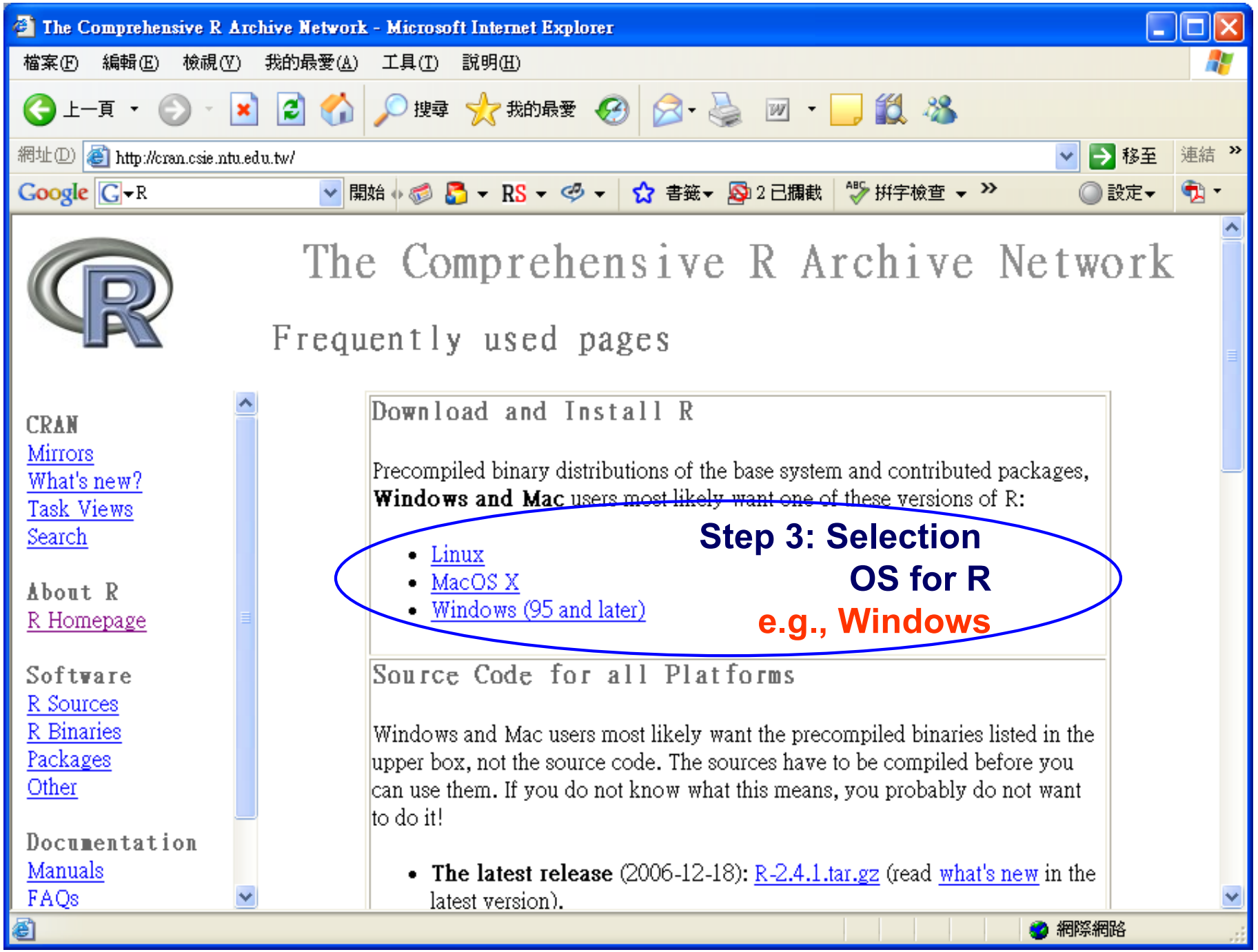
Installation of R

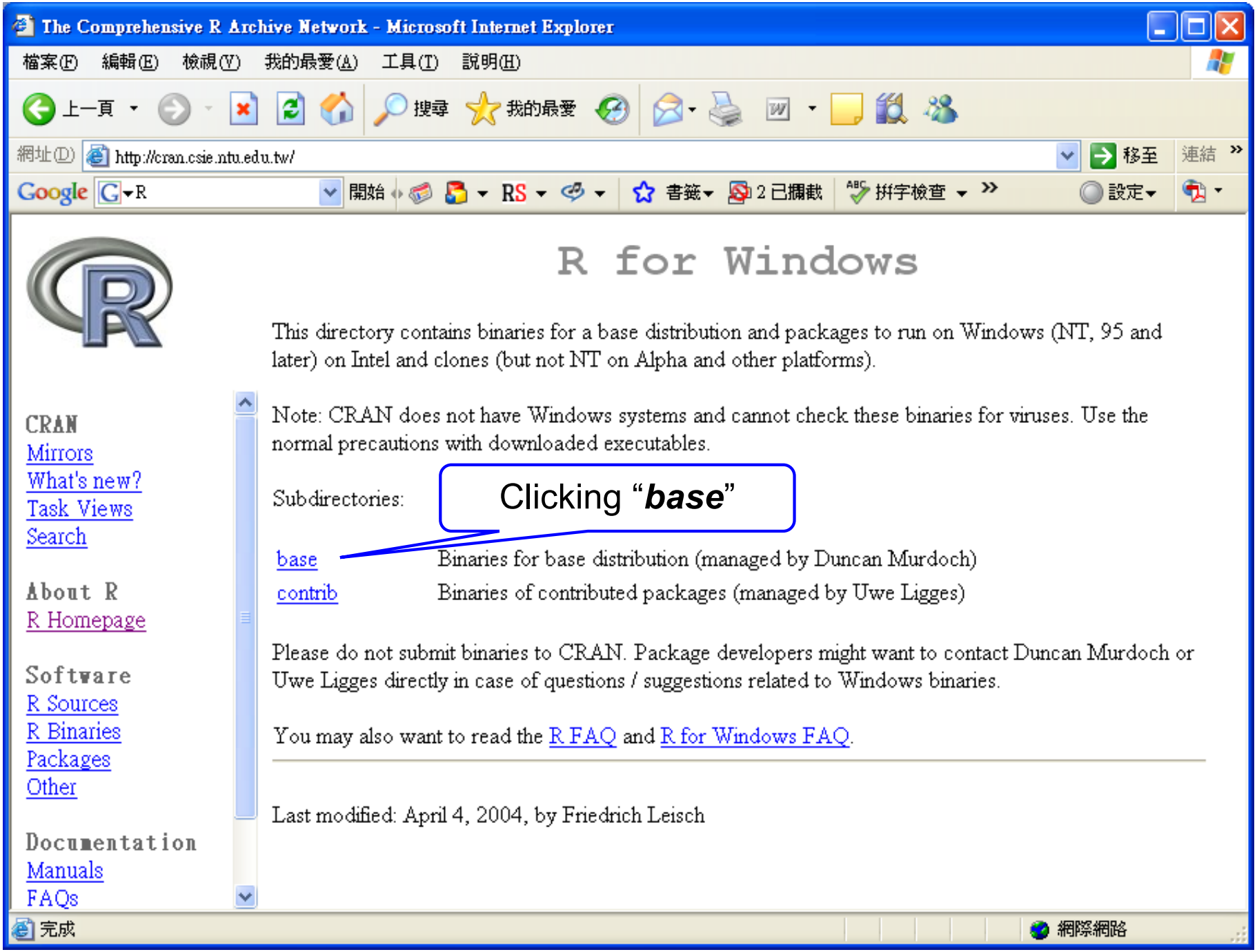
http://www.r-project.org/

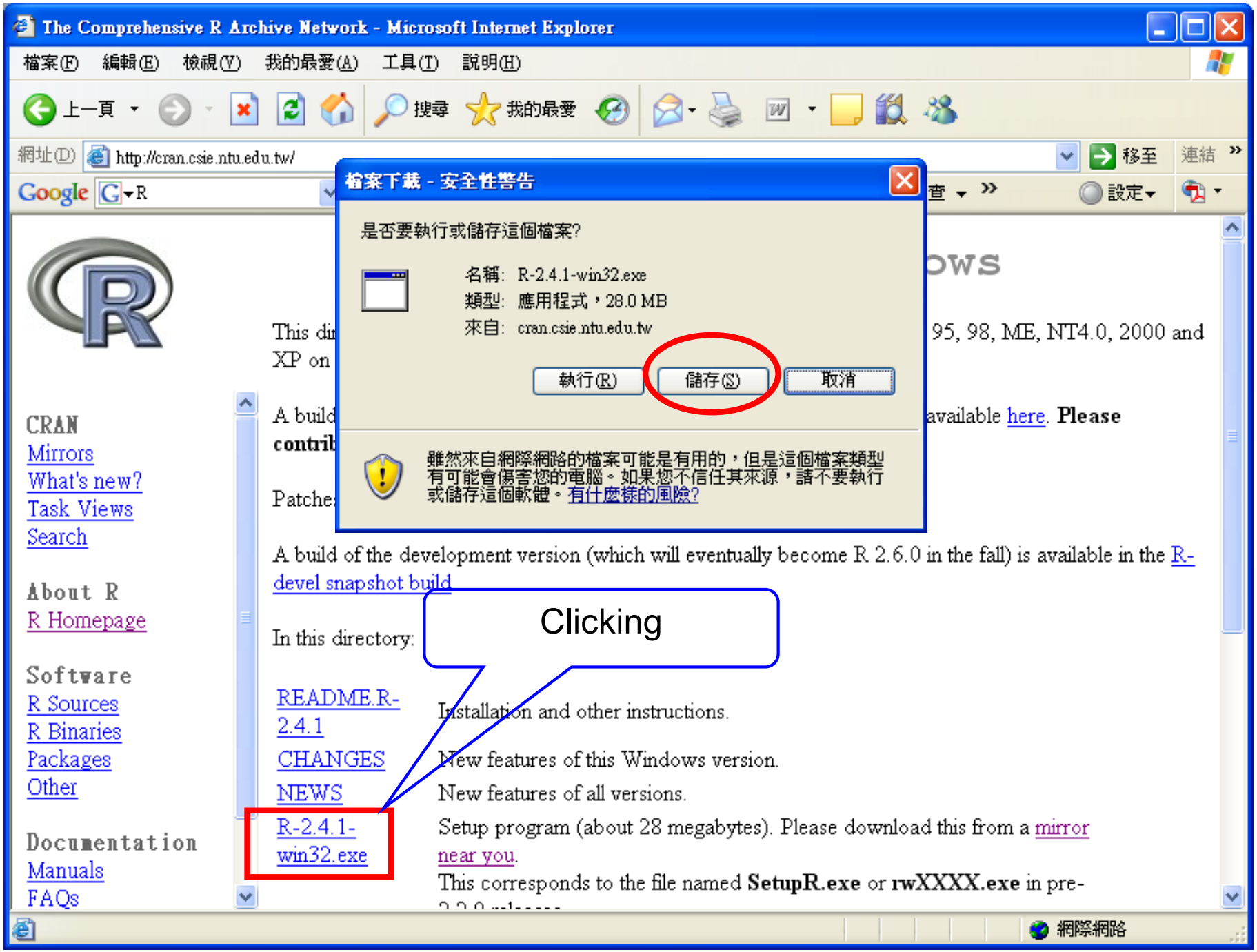


Selection a Mirror Site

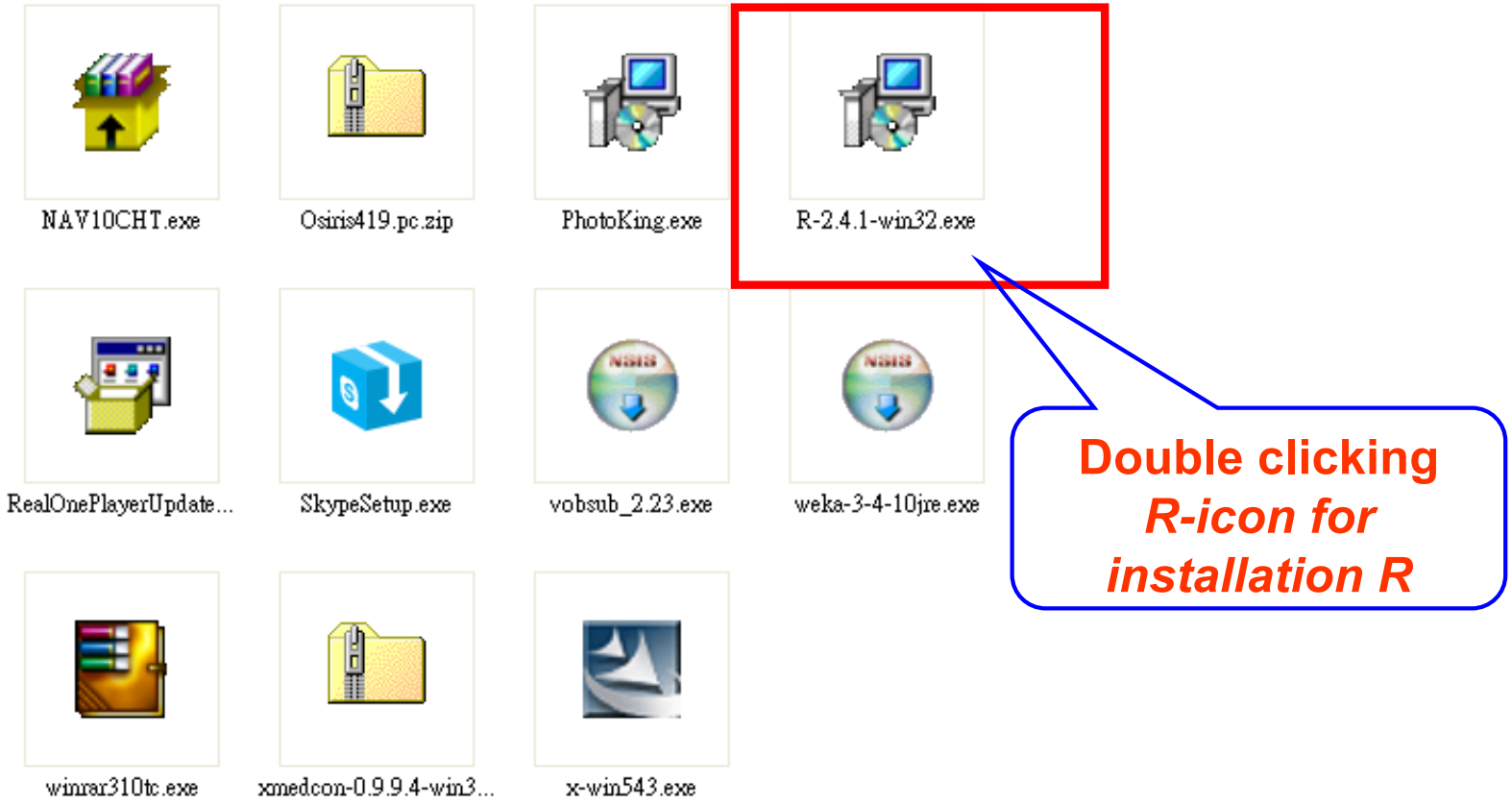



The Comprehensive R Archive Network - Microsoft Internet Explorer
http://cran.csie.ntu.edu.tw/
The Comprehensive R Archive Network
Frequently used pages
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Download and Install 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 (95 and later)
Step 3: Selection
OS for R
e.g., 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 (2006-12-18): R-2.4.1.tar.gz (read what's new in the latest version).

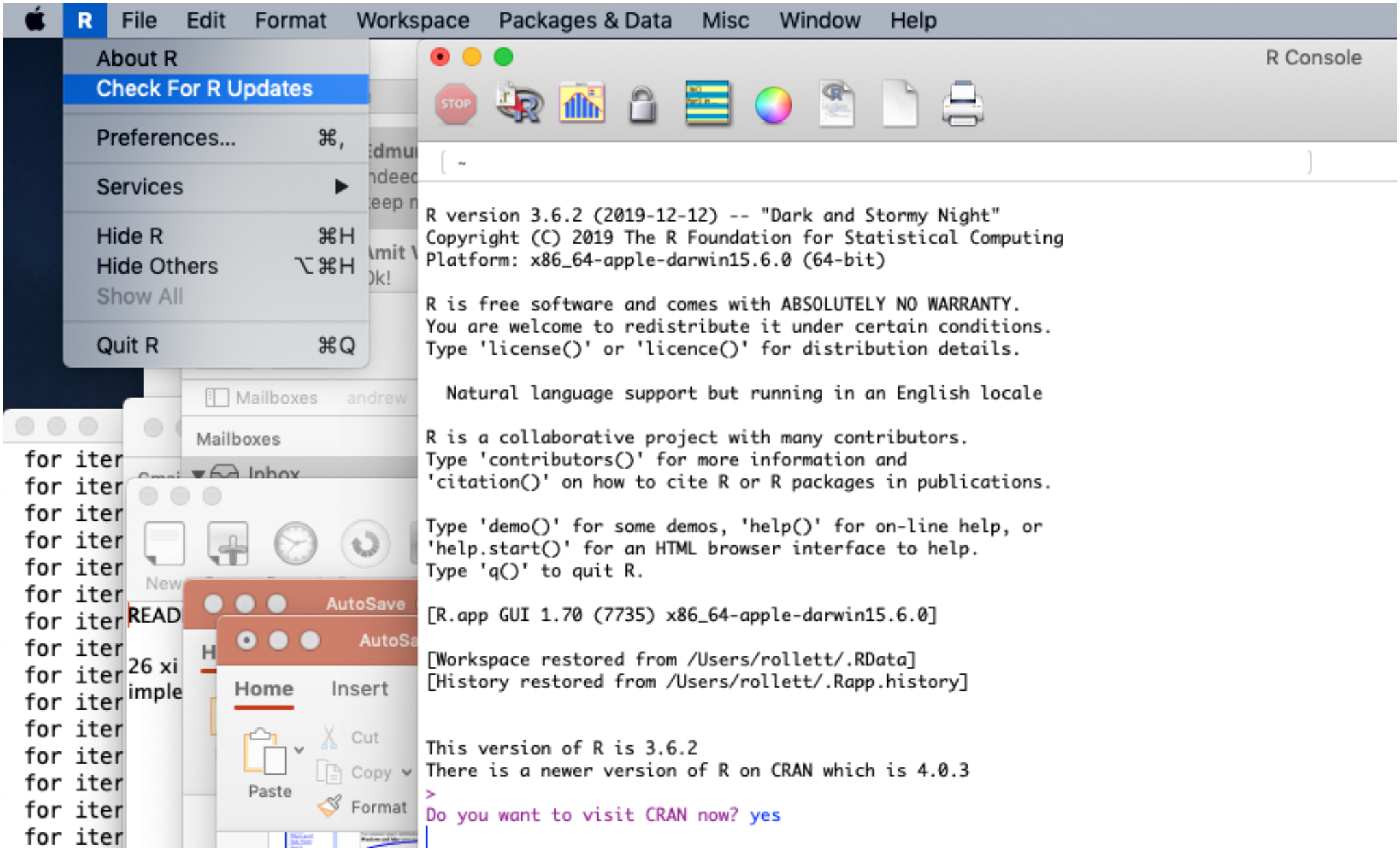

The Comprehensive R Archive Network - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛
網址(D) http://cran.csie.ntu.edu.tw/
移至 連結
Google R 開始 書籤 2 已攔截 拼字檢查 設定
R for Windows
This directory contains binaries for a base distribution and packages to run on Windows (NT, 95 and later) on Intel and clones (but not NT on Alpha and other platforms).
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Note: CRAN does not have Windows systems and cannot check these binaries for viruses. Use the normal precautions with downloaded executables.
Subdirectories:
Clicking "base"
base Binaries for base distribution (managed by Duncan Murdoch)
contrib Binaries of contributed packages (managed by Uwe Ligges)
Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.
You may also want to read the R FAQ and R for Windows FAQ.
Last modified: April 4, 2004, by Friedrich Leisch
完成
網際網路


The Comprehensive R Archive Network - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛
網址(D) http://cran.csie.ntu.edu.tw/
移至 連結
Google R
設定
檔案下載 - 安全性警告
是否要執行或儲存這個檔案?
名稱: R-2.4.1-win32.exe
類型: 應用程式，28.0 MB
來自: cran.csie.ntu.edu.tw
執行(R) 儲存(S) 取消
雖然來自網際網路的檔案可能是有用的，但是這個檔案類型有可能會傷害您的電腦。如果您不信任其來源，請不要執行或儲存這個軟體。有什麼樣的風險?
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
95, 98, ME, NT4.0, 2000 and
available here. Please
A build of the development version (which will eventually become R 2.6.0 in the fall) is available in the R-devel snapshot build
In this directory:
README.R-2.4.1 Installation and other instructions.
CHANGES New features of this Windows version.
NEWS New features of all versions.
R-2.4.1-win32.exe Setup program (about 28 megabytes). Please download this from a mirror near you.
This corresponds to the file named SetupR.exe or rwXXXX.exe in pre-
Clicking
網際網路

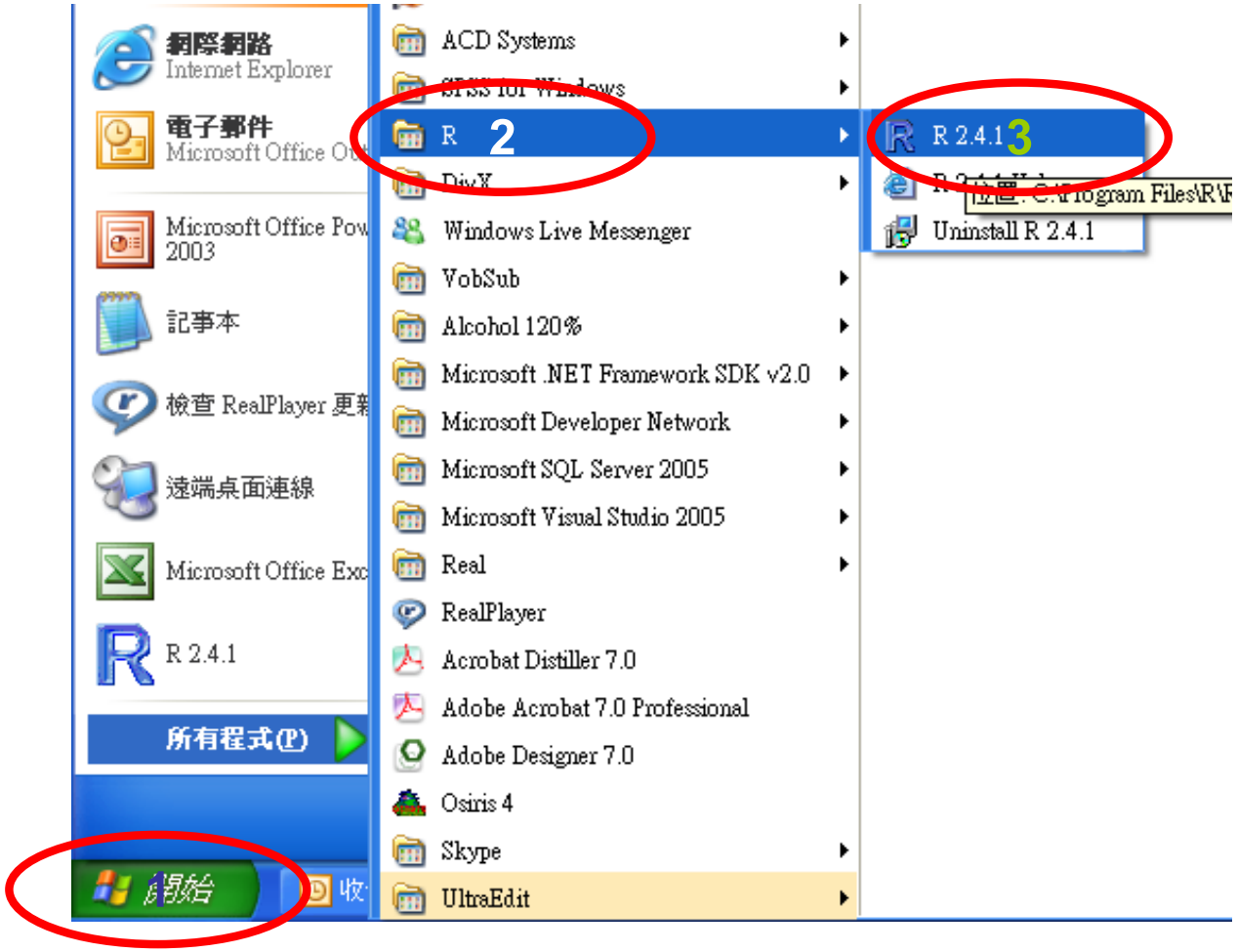
Installation of R



Update if You Already Have R



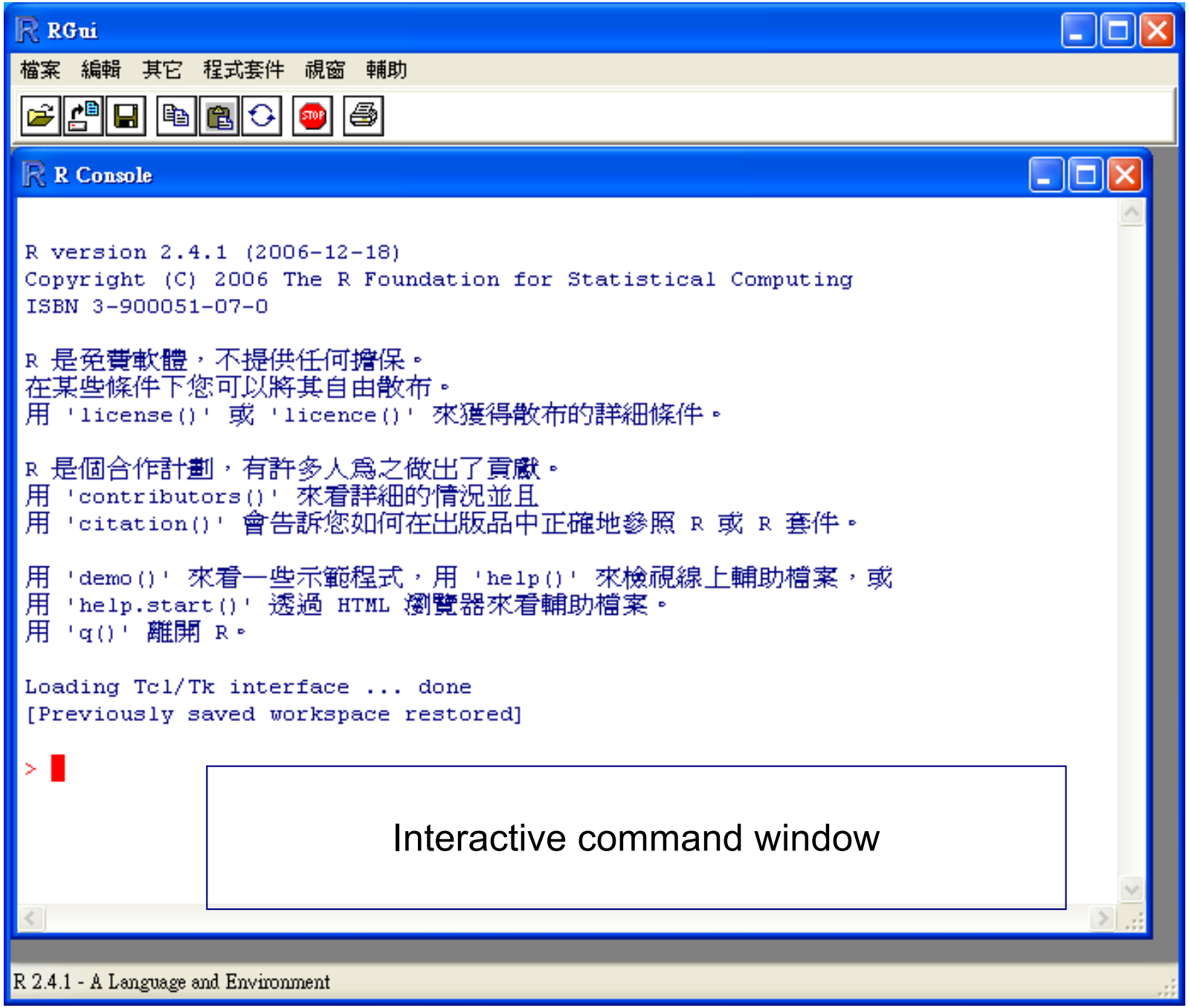
Using R



Or, on a Mac, double-click "R" in the Applications window.

You can also put the R icon in your Dock or in the side-bar.

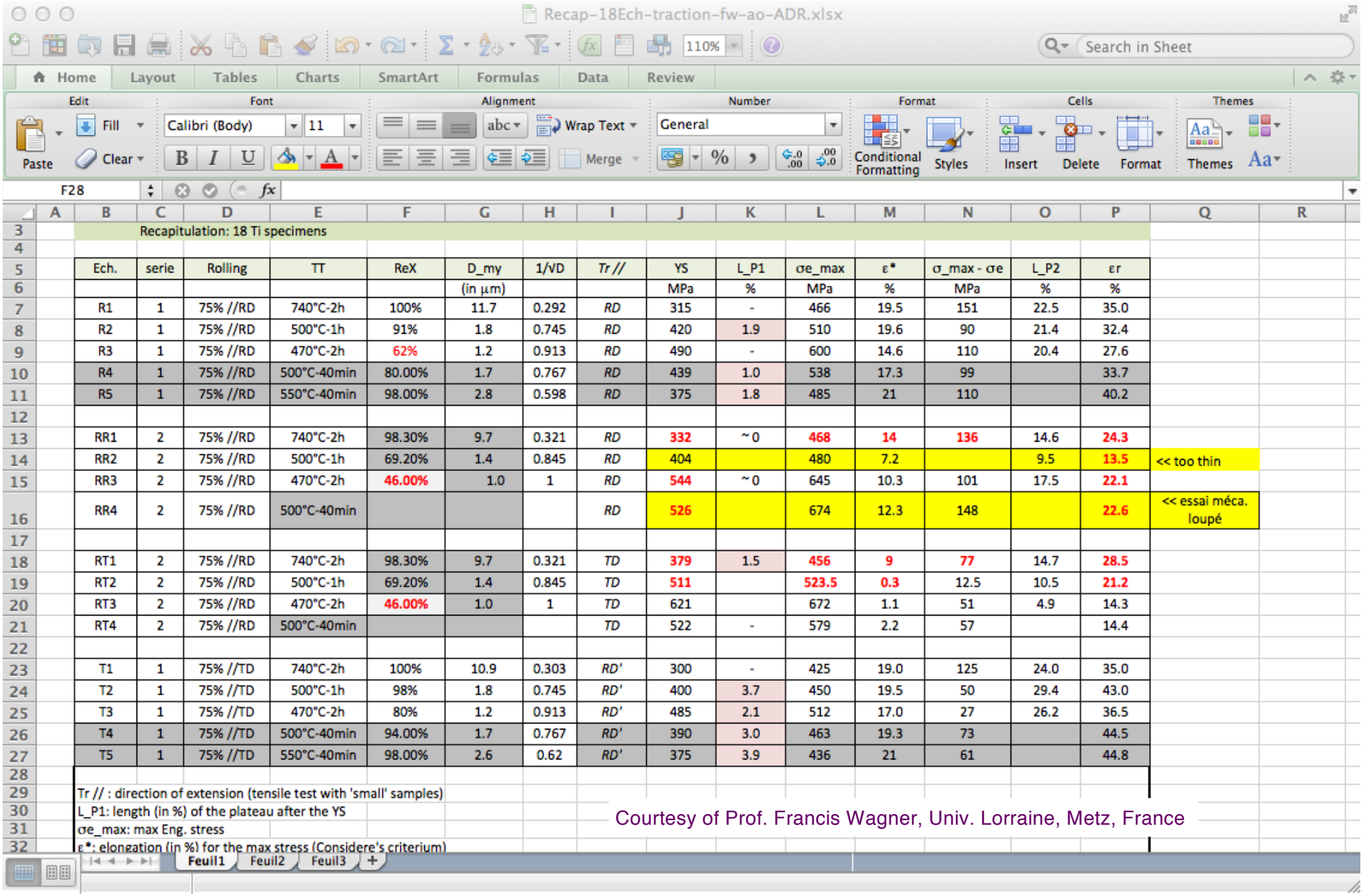
Same for R Studio.


RGui
檔案 編輯 其它 程式套件 視窗 輔助
R Console
R version 2.4.1 (2006-12-18)
Copyright (C) 2006 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
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。
Loading Tcl/Tk interface ... done
[Previously saved workspace restored]
>
Interactive command window
R 2.4.1 - A Language and Environment

Environment Commands

```
> search() # loaded packages
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"

> ls() # Used objects
 [1] "bb"              "col"             "colorlut"        "EdgeList"
 [5] "Edges"           "EXP"             "f“

> rm(bb) # remove object “bb”

> ?lm  # ? Function name = look function

> args(lm) # look arguments in “lm”

> help(lm) #See detail of function (lm)
```

From a spreadsheet of data …



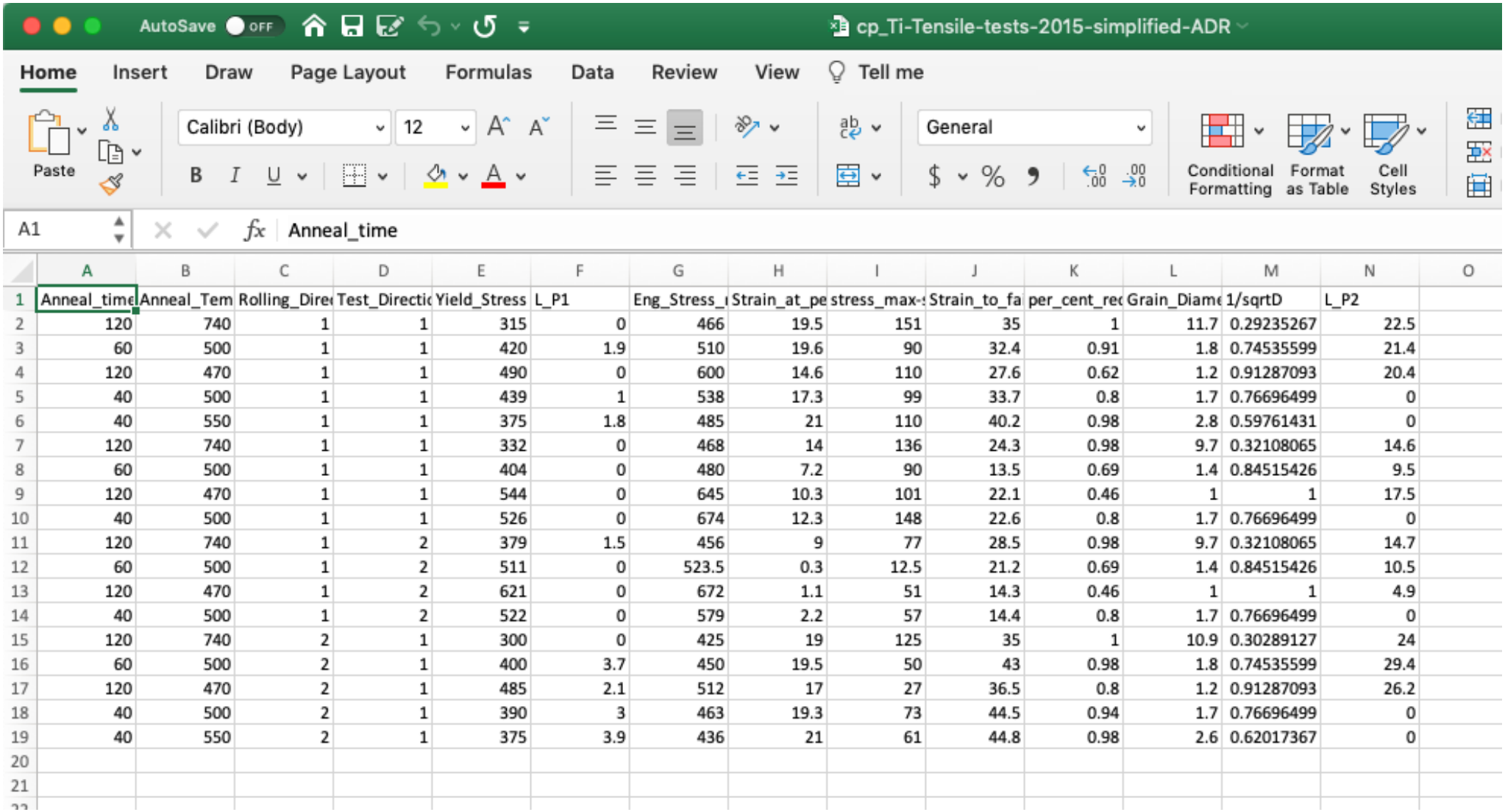
Recapitulation: 18 Ti specimens

Ech.	serie	Rolling	TT	ReX	D_my (in µm)	1/VD	Tr //	YS MPa	L_P1 %	σe_max MPa	ε* %	σ_max - σe MPa	L_P2 %	εr %	
R1	1	75% //RD	740°C-2h	100%	11.7	0.292	RD	315	-	466	19.5	151	22.5	35.0	
R2	1	75% //RD	500°C-1h	91%	1.8	0.745	RD	420	1.9	510	19.6	90	21.4	32.4	
R3	1	75% //RD	470°C-2h	62%	1.2	0.913	RD	490	-	600	14.6	110	20.4	27.6	
R4	1	75% //RD	500°C-40min	80.00%	1.7	0.767	RD	439	1.0	538	17.3	99		33.7	
R5	1	75% //RD	550°C-40min	98.00%	2.8	0.598	RD	375	1.8	485	21	110		40.2	
RR1	2	75% //RD	740°C-2h	98.30%	9.7	0.321	RD	332	~0	468	14	136	14.6	24.3	
RR2	2	75% //RD	500°C-1h	69.20%	1.4	0.845	RD	404		480	7.2		9.5	13.5	<< too thin
RR3	2	75% //RD	470°C-2h	46.00%	1.0	1	RD	544	~0	645	10.3	101	17.5	22.1	
RR4	2	75% //RD	500°C-40min				RD	526		674	12.3	148		22.6	<< essai méca. loupé
RT1	2	75% //RD	740°C-2h	98.30%	9.7	0.321	TD	379	1.5	456	9	77	14.7	28.5	
RT2	2	75% //RD	500°C-1h	69.20%	1.4	0.845	TD	511		523.5	0.3	12.5	10.5	21.2	
RT3	2	75% //RD	470°C-2h	46.00%	1.0	1	TD	621		672	1.1	51	4.9	14.3	
RT4	2	75% //RD	500°C-40min				TD	522	-	579	2.2	57		14.4	
T1	1	75% //TD	740°C-2h	100%	10.9	0.303	RD'	300	-	425	19.0	125	24.0	35.0	
T2	1	75% //TD	500°C-1h	98%	1.8	0.745	RD'	400	3.7	450	19.5	50	29.4	43.0	
T3	1	75% //TD	470°C-2h	80%	1.2	0.913	RD'	485	2.1	512	17.0	27	26.2	36.5	
T4	1	75% //TD	500°C-40min	94.00%	1.7	0.767	RD'	390	3.0	463	19.3	73		44.5	
T5	1	75% //TD	550°C-40min	98.00%	2.6	0.62	RD'	375	3.9	436	21	61		44.8	

Tr // : direction of extension (tensile test with 'small' samples)
L_P1: length (in %) of the plateau after the YS
σe_max: max Eng. stress
ε*: elongation (in %) for the max stress (Considere's criterium)

Courtesy of Prof. Francis Wagner, Univ. Lorraine, Metz, France

Such a spreadsheet requires "clean up"

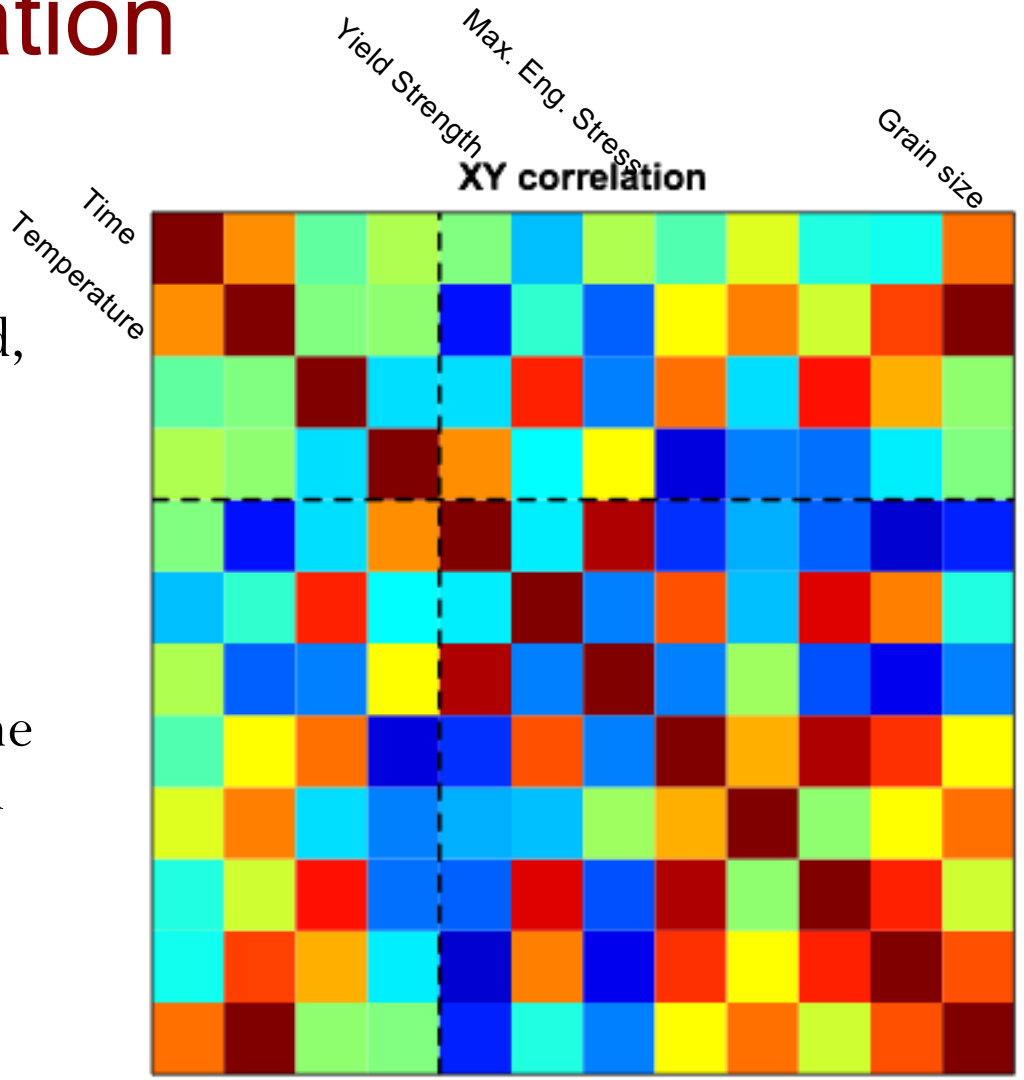


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Anneal_time	Anneal_Tem	Rolling_Direc	Test_Directio	Yield_Stress	L_P1	Eng_Stress_i	Strain_at_pe	stress_max-s	Strain_to_fa	per_cent_red	Grain_Diame	1/sqrtD	L_P2	
2	120	740	1	1	315	0	466	19.5	151	35	1	11.7	0.29235267	22.5	
3	60	500	1	1	420	1.9	510	19.6	90	32.4	0.91	1.8	0.74535599	21.4	
4	120	470	1	1	490	0	600	14.6	110	27.6	0.62	1.2	0.91287093	20.4	
5	40	500	1	1	439	1	538	17.3	99	33.7	0.8	1.7	0.76696499	0	
6	40	550	1	1	375	1.8	485	21	110	40.2	0.98	2.8	0.59761431	0	
7	120	740	1	1	332	0	468	14	136	24.3	0.98	9.7	0.32108065	14.6	
8	60	500	1	1	404	0	480	7.2	90	13.5	0.69	1.4	0.84515426	9.5	
9	120	470	1	1	544	0	645	10.3	101	22.1	0.46	1	1	17.5	
10	40	500	1	1	526	0	674	12.3	148	22.6	0.8	1.7	0.76696499	0	
11	120	740	1	2	379	1.5	456	9	77	28.5	0.98	9.7	0.32108065	14.7	
12	60	500	1	2	511	0	523.5	0.3	12.5	21.2	0.69	1.4	0.84515426	10.5	
13	120	470	1	2	621	0	672	1.1	51	14.3	0.46	1	1	4.9	
14	40	500	1	2	522	0	579	2.2	57	14.4	0.8	1.7	0.76696499	0	
15	120	740	2	1	300	0	425	19	125	35	1	10.9	0.30289127	24	
16	60	500	2	1	400	3.7	450	19.5	50	43	0.98	1.8	0.74535599	29.4	
17	120	470	2	1	485	2.1	512	17	27	36.5	0.8	1.2	0.91287093	26.2	
18	40	500	2	1	390	3	463	19.3	73	44.5	0.94	1.7	0.76696499	0	
19	40	550	2	1	375	3.9	436	21	61	44.8	0.98	2.6	0.62017367	0	
20															
21															

Column names must be simplified. Check that each column is numeric, not text. No comments or extraneous text allowed.

… to a correlation analysis

For a very straightforward, visual way to visualize correlations and cross-correlations, one can use *matcor*. The top right and bottom left quadrants are mirror images and show the cross-correlation between input & output.

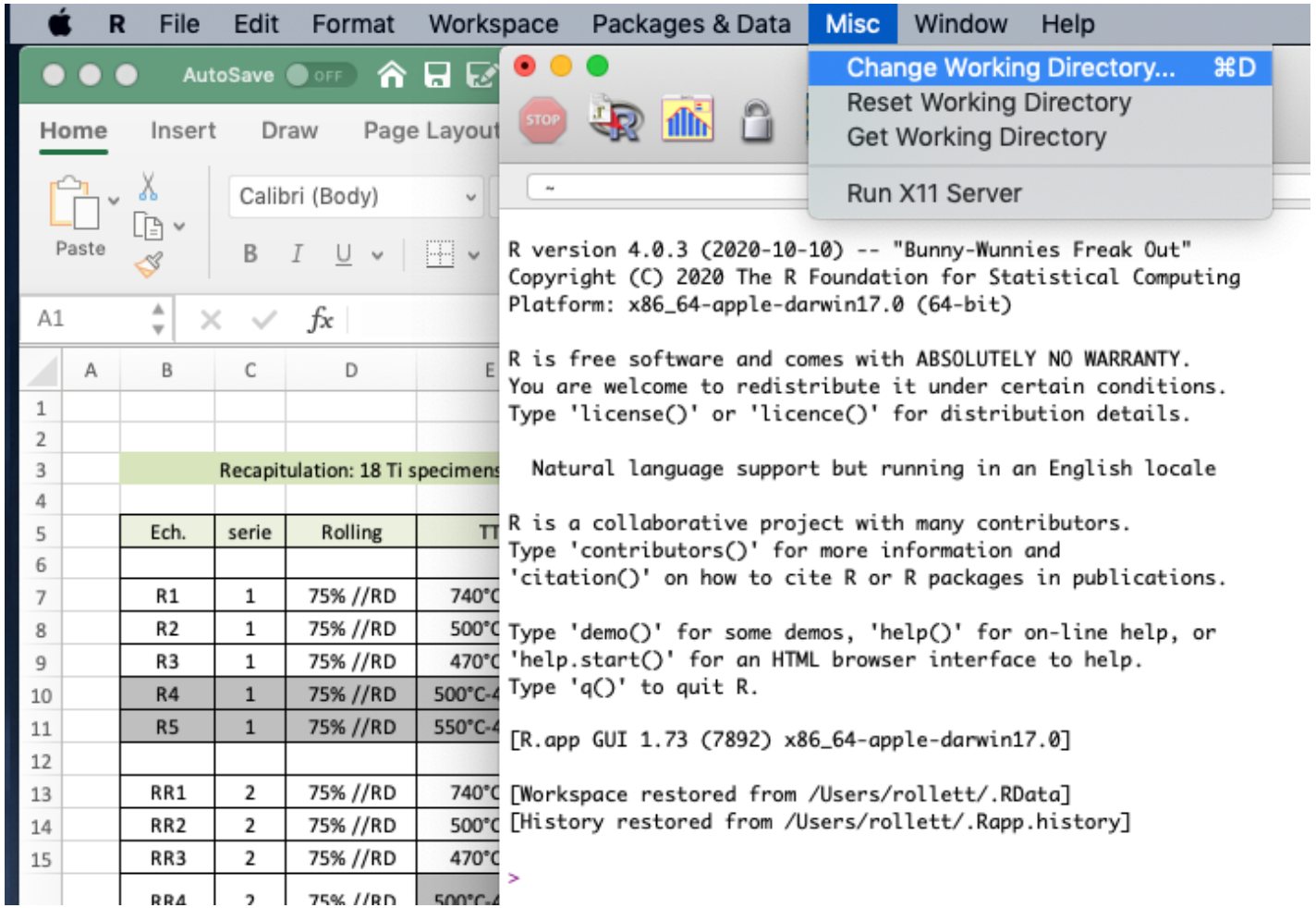



```
➢ Data = read.table('file')
➢ outvars=data[,1:2]
➢ invars = data[,3:12]
> simpleCorr=matcor(invars,outvars)
> img.matcor(simpleCorr)
```

Thanks to Francis Wagner, Univ. Metz, France

Change Working Directory

Navigate to where you have data* stored …



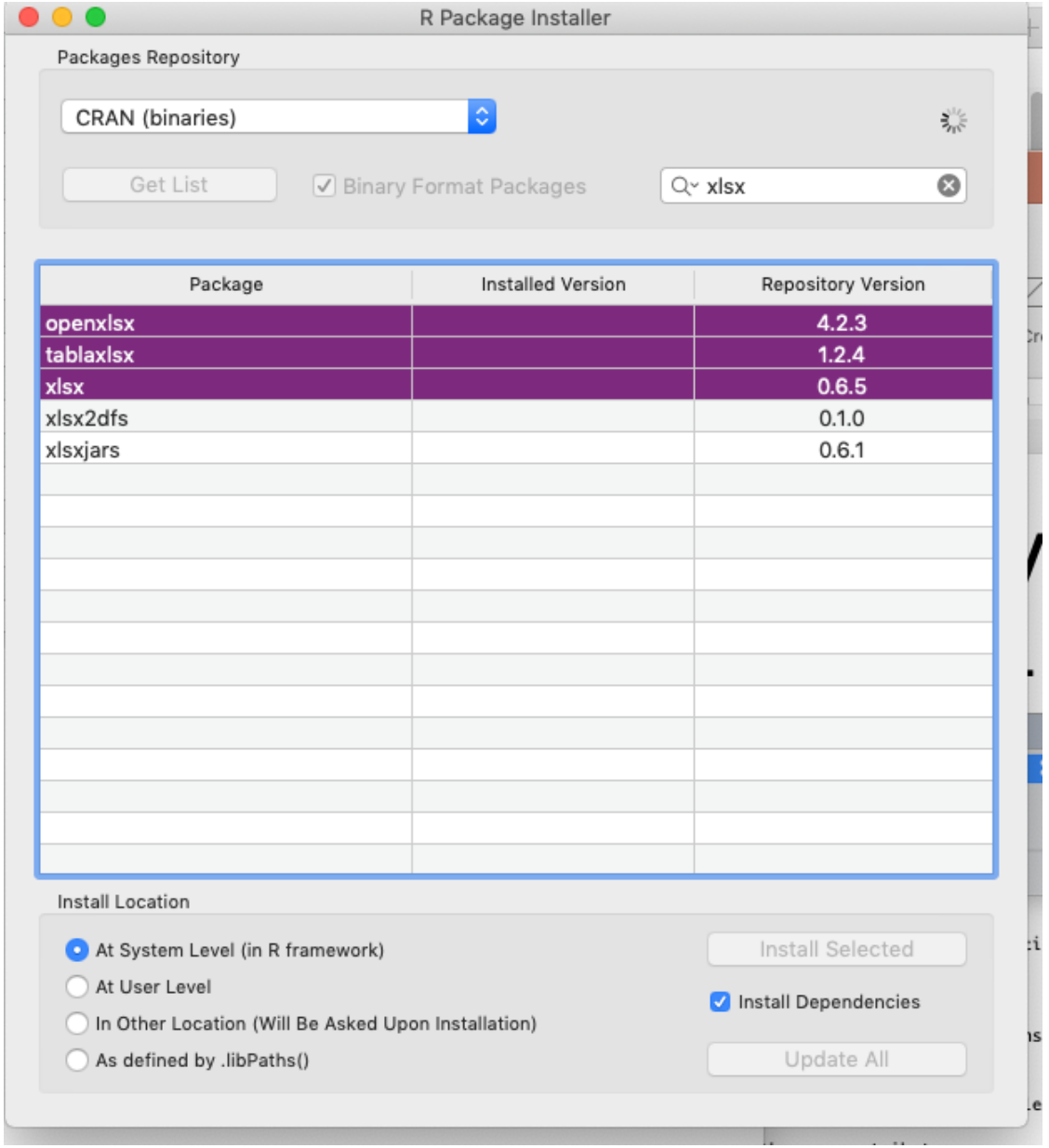
* cp_Ti-Tensile-tests-2015-simplified-ADR.xlsx

Read, Write XLSX

Use the Package Manager

…

to install useful packages such as "openxlsx", which allows you to read/write excel spreadsheets



Even so …

```
> Data = read.xlsx(‘cp_Ti-Tensile-tests-2015-simplified-ADR.xlsx')
Error: unexpected input in "Data = read.xlsx(‘"
> Data = read.xlsx(‘cp_Ti-Tensile-tests-2015-simplified-ADR.xlsx')
Error: unexpected input in "Data = read.xlsx(‘"
> Data = read.xlsx(cp_Ti-Tensile-tests-2015-simplified-ADR.xlsx)
Error in read.xlsx(cp_Ti - Tensile - tests - 2015 - simplified - ADR.xlsx) :
  object 'cp_Ti' not found
> Data = read.xlsx("cp_Ti-Tensile-tests-2015-simplified-ADR.xlsx")
```

In-class exercise: how did we circumvent these errors?!

Hint: look very carefully at all the characters used to specify the file

Get a "summary" of the Data

```
> summary(Data)
  Anneal_time  Anneal_Temperature_degrC Rolling_Direction Test_Direction   Yield_Stress
 Min.   : 40   Min.   :470.0            Min.   :1.000     Min.   :1.000   Min.   :300.0
 1st Qu.: 40   1st Qu.:500.0            1st Qu.:1.000     1st Qu.:1.000   1st Qu.:376.0
 Median : 60   Median :500.0            Median :1.000     Median :1.000   Median :412.0
 Mean   : 80   Mean   :552.2            Mean   :1.278     Mean   :1.222   Mean   :434.9
 3rd Qu.:120   3rd Qu.:550.0            3rd Qu.:1.750     3rd Qu.:1.000   3rd Qu.:505.8
 Max.   :120   Max.   :740.0            Max.   :2.000     Max.   :2.000   Max.   :621.0
      L_P1       Eng_Stress_max  Strain_at_peak_stress stress_max-s_yield Strain_to_failure
 Min.   :0.000   Min.   :425.0   Min.   : 0.300        Min.   : 12.50     Min.   :13.50
 1st Qu.:0.000   1st Qu.:463.8   1st Qu.: 9.325        1st Qu.: 58.00     1st Qu.:22.23
 Median :0.000   Median :497.5   Median :15.800        Median : 90.00     Median :30.45
 Mean   :1.050   Mean   :521.2   Mean   :13.567        Mean   : 87.14     Mean   :29.64
 3rd Qu.:1.875   3rd Qu.:568.8   3rd Qu.:19.450        3rd Qu.:110.00     3rd Qu.:36.12
 Max.   :3.900   Max.   :674.0   Max.   :21.000        Max.   :151.00     Max.   :44.80
 per_cent_recrystallized Grain_Diameter      1/sqrtD            L_P2
 Min.   :0.4600          Min.   : 1.000   Min.   :0.2924   Min.   : 0.00
 1st Qu.:0.7175          1st Qu.: 1.400   1st Qu.:0.6033   1st Qu.: 0.00
 Median :0.8550          Median : 1.700   Median :0.7670   Median :12.55
 Mean   :0.8261          Mean   : 3.611   Mean   :0.6961   Mean   :11.98
 3rd Qu.:0.9800          3rd Qu.: 2.750   3rd Qu.:0.8452   3rd Qu.:21.15
 Max.   :1.0000          Max.   :11.700   Max.   :1.0000   Max.   :29.40
```

Access a single column with "$"

Note that almost any statistical quantity that you need can be used as function, e.g., mean, median etc.

Suffixing with $ and the column name will access that part of the data

```
> mean(Data$Anneal_time)
[1] 80
```

Operators

- Mathematical operators: + - * / ^
  - Mod: %%
  - sqrt, exp, log, log10, sin, cos, tan, …..
- Other operators:
  - $ component selection HIGH
  - [, [[subscripts, elements
  - : sequence operator
  - %*% matrix algebra
  - <, >, <=, >= inequality
  - ==, != comparison
  - ! not
  - &, |, &&, || and, or
  - ~ formulas
  - <- assignment (or = 1.9.1 later)

Demo Algebra, Operators and Functions

```
> 1+2
[1] 3
> 1 > 2
[1] FALSE
> 1 > 2 | 2 > 1
[1] TRUE
> 1:3
[1] 1 2 3
> A = 1:3
> A
[1] 1 2 3
> A*6
[1]  6 12 18
> A/10
[1] 0.1 0.2 0.3
> A %% 2
[1] 1 0 1
```

```
> B=4:6
> A*B
[1]  4 10 18
> A%*%B
     [,1]
[1,]   32
> A %*% t(B)
     [,1] [,2] [,3]
[1,]    4    5    6
[2,]    8   10   12
[3,]   12   15   18
> A / B
[1] 0.25 0.40 0.50
> sqrt(A)
[1] 1.000000 1.414214 1.732051
> log(A)
[1] 0.0000000 0.6931472 1.0986123
```

```
> round(sqrt(A),2)
[1] 1.00 1.41 1.73
> ceiling(sqrt(A))
[1] 1 2 2
> floor(sqrt(A))
[1] 1 1 1
> eigen( A%*% t(B))
$values
[1] 3.200000e+01 5.835176e-16 2.480655e-16

$vectors
          [,1]       [,2]       [,3]
[1,] 0.2672612  0.3273463 -0.8890009
[2,] 0.5345225 -0.8217055  0.2540003
[3,] 0.8017837  0.4665237  0.3810004
> eigen( A%*% t(B))$values
[1] 3.200000e+01 5.835176e-16 2.480655e-16
```

Distributions in R

- Notation:
  - Probability Density Function: **p**
  - Distribution Function: **p**
  - Quantile function: **q**
  - Random generation for distribution: **r**
- Example:
  - Normal distribution:
    - dnorm(x, mean=0, sd=1, log = FALSE)
    - pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
    - qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
    - rnorm(n, mean=0, sd=1)

- Weibull Distribution
  - dweibull(x, shape, scale = 1, log = FALSE)
  - pweibull(q, shape, scale = 1, lower.tail = TRUE, log.p = FALSE)
  - qweibull(p, shape, scale = 1, lower.tail = TRUE, log.p = FALSE)
  - rweibull(n, shape, scale = 1)
- Log Normal Distribution
  - dlnorm(x, meanlog = 0, sdlog = 1, log = FALSE)
  - plnorm(q, meanlog = 0, sdlog = 1, lower.tail = TRUE, log.p = FALSE)
  - qlnorm(p, meanlog = 0, sdlog = 1, lower.tail = TRUE, log.p = FALSE)
  - rlnorm(n, meanlog = 0, sdlog = 1)

Statistical Functions

Excel	R
NORMSDIST	pnorm(7.2,mean=5,sd=2)
NORMSINV	qnorm(0.9,mean=5,sd=2)
LOGNORMDIST	plnorm(7.2,meanlog=5,sdlog=2)
LOGINV	qlnorm(0.9,meanlog=5,sdlog=2)
GAMMADIST	pgamma(31, shape=3, scale =5)
GAMMAINV	qgamma(0.95, shape=3, scale =5)
GAMMALN	lgamma(4)
WEIBULL	pweibull(6, shape=3, scale =5)
BINOMDIST	pbinom(2,size=20,p=0.3)
POISSON	ppois(2, lambda =3)

Write Data to a TXT File

- Usage:
- ***write***(x, file, …)

- x<-matrix(c(1.0, 2.0, 3.0, 4.0, 5.0, 6.0), 2, 3) ; x
- [,1] [,2] [,3]
- [1,] 1 3 5
- [2,] 2 4 6
- write(t(x), file="d:/out2.txt", ncolumns=3)
- write(x, file="d:/out3.txt", ncolumns=3)

d:/out2.txt
1 3 5
2 4 6

d:/out3.txt
1 2 3
4 5 6

Write Data to a CSV File

- Usage:
- ***write.table***(x, file = "foo.csv", sep=", ",…)
- Example:
- x<-matrix(c(1.0, 2.0, 3.0, 4.0, 5.0, 6.0), 2, 3) ; x
- [,1] [,2] [,3]
- [1,] 1 3 5
- [2,] 2 4 6
- write.table(**t(x)**, file="d:/out4.txt", sep=",", col.names=FALSE, row.names=FALSE)
- write.table(x, file="d:/out5.txt",sep=",", col.names=FALSE, row.names=FALSE)

  d:/out4.txt
  1,2
  3,4
  5,6

  d:/out5.txt
  1,3,5
  2,4,6

Read TXT and CSV File

- Usage:
- ***read.table*** (file,…)

- X=read.table(file="d:/out2.txt"); X
- V1 V2 V3
- 1 1 3 5
- 2 2 4 6
- X=read.table(file="d:/out5.txt",sep=", ", header=FALSE); X

d:/out2.txt
1 3 5
2 4 6

d:/out5.csv
1,3,5
2,4,6

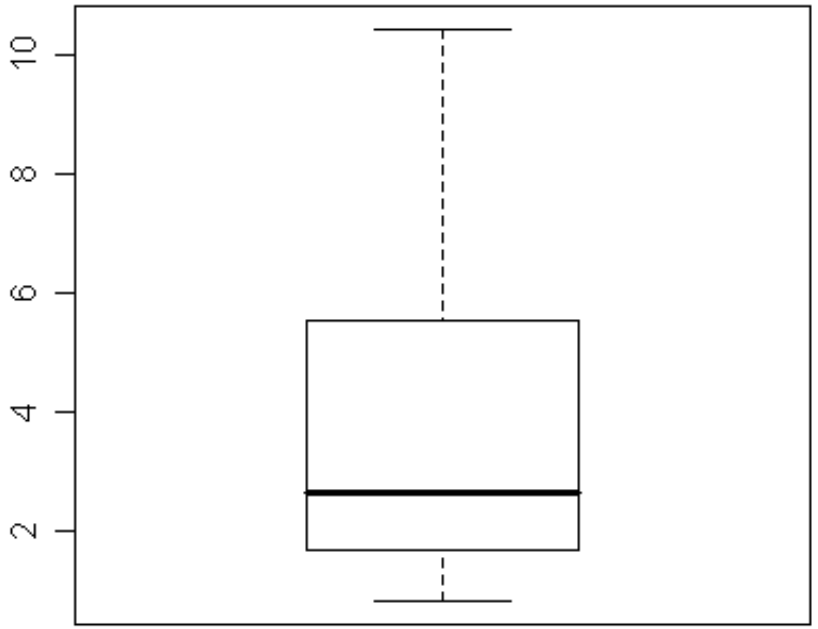
Demo Reading CSV File

```
> Data = read.table(file="d:/01.csv",header=TRUE, sep=",")
> Data
         Y       X1       X2
1  2.651680 13.808986 26.75896
2  1.875039 17.734523 37.89857
3  1.523964 19.891025 26.03624
4  2.984314 15.574261 30.21754
5 10.423087  9.293612 28.91459
6  0.840065  8.830160 30.38578
7  8.126936  9.615875 32.69579
> mean(Data$Y) # $ is used to identify which variable
[1] 4.060726
> mean(Data$X1)
[1] 13.53549
> mean(Data$X2)
[1] 30.41535
> sd(Data$Y)
[1] 3.691044
> summary(Data$Y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.8401  1.7000  2.6520  4.0610  5.5560 10.4200
> boxplot(Data$Y)
```

01.csv

Book1

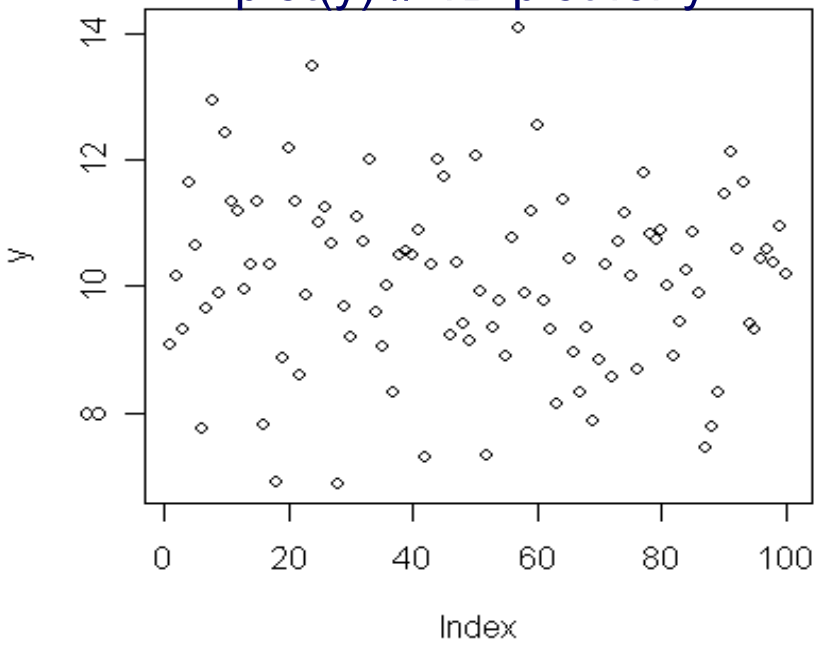
	A	B	C
1	Y	X1	X2
2	2.65168	13.80899	26.75896
3	1.875039	17.73452	37.89857
4	1.523964	19.89103	26.03624
5	2.984314	15.57426	30.21754
6	10.42309	9.293612	28.91459
7	0.840065	8.83016	30.38578
8	8.126936	9.615875	32.69579

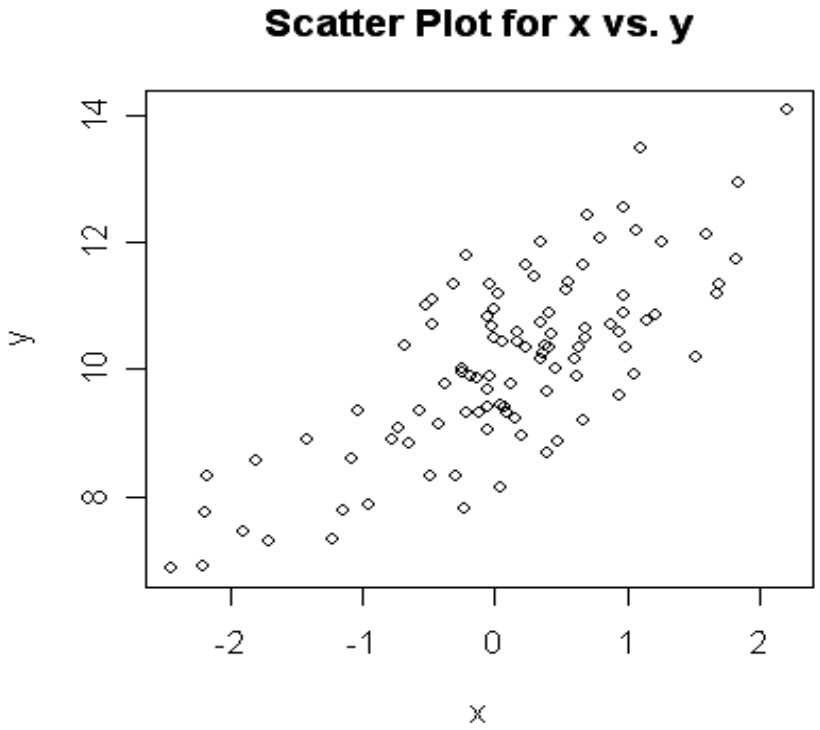


Scatter Plot in R

- Scatter plot:
  - x = rnorm(100) # Generated 100 N(0,1)
  - y = 10 + 1.2*x + rnorm(100) # Simulated y
  - plot(y) # 1D plot for y
  - windows() # Create a new graph device
  - plot(x,y) # 2D plot for x vs y
  - title (main="Scatter Plot for x vs. y") # Add title in plot

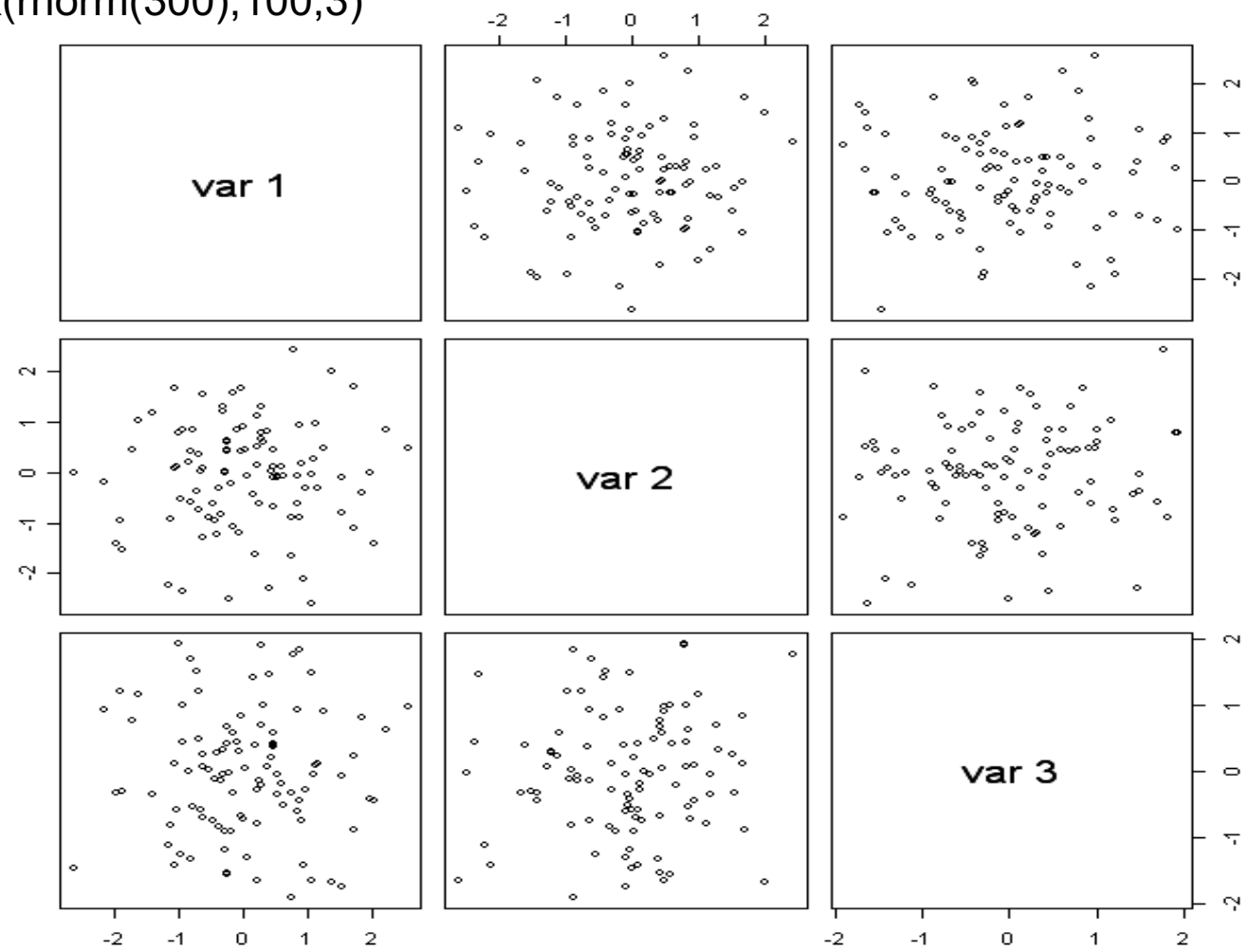
plot(y) # 1D plot for y





Scatterplot Matrices in R

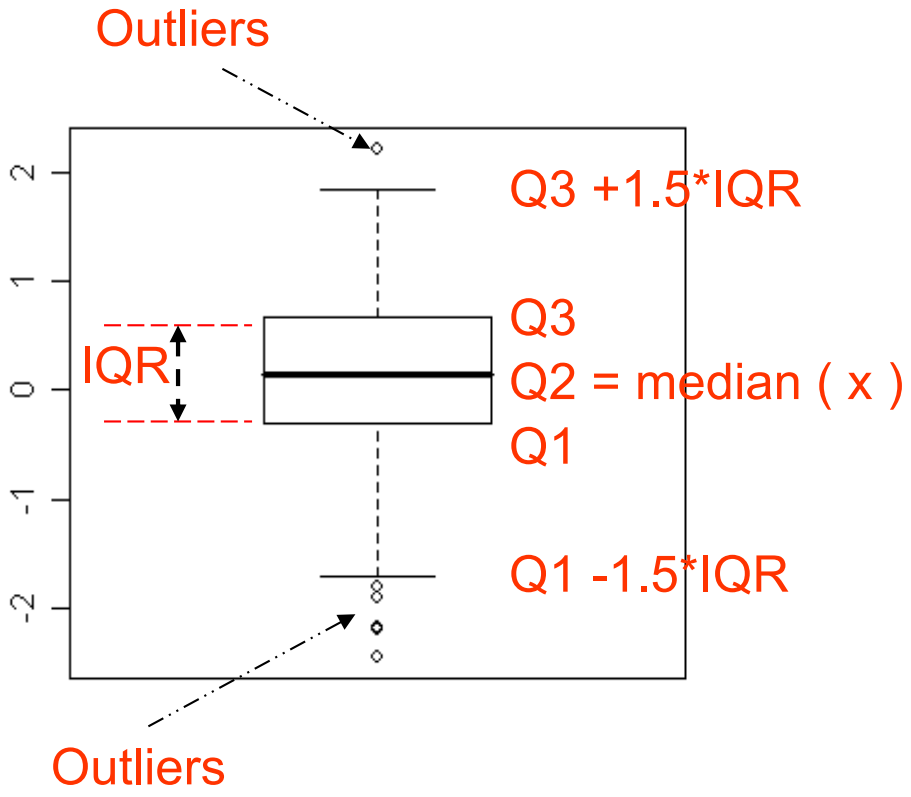
- A matrix of scatterplots is produced.
  - **Usage:**
    - pairs(x, ...)
  - Example:
    - X = matrix(rnorm(300),100,3)
    - pairs(X)

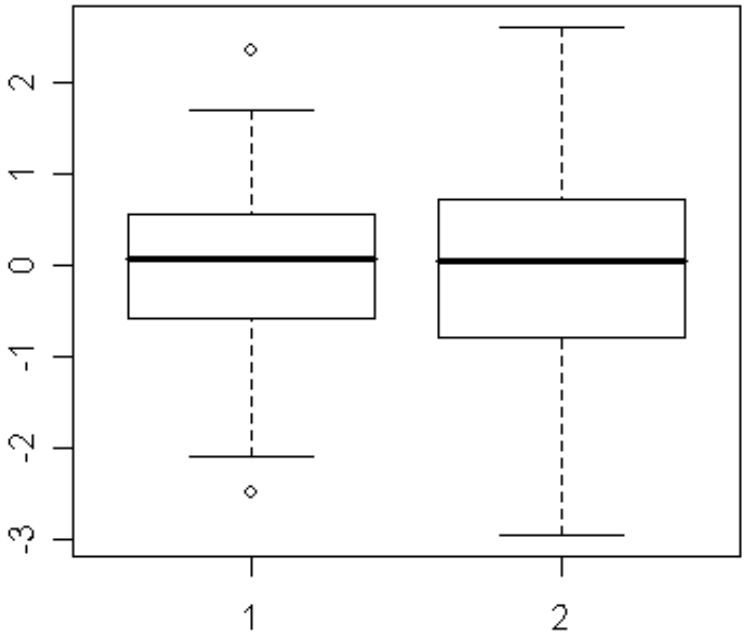


Box Plot in R

- Box plot:
  - Produce box-and-whisker plot of the given (grouped) values.
  - **Usage:** boxplot(x, ...)
  - Example1:
    - X=rnorm(100)
    - boxplot(x)

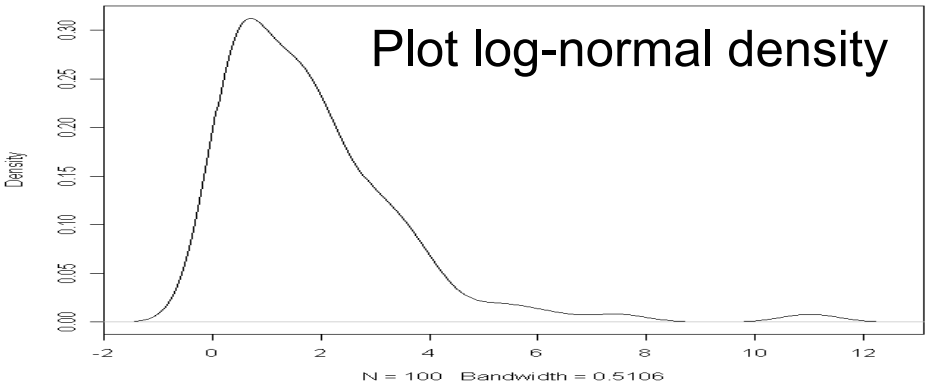
Example2:
x=rnorm(100); y=rnorm(100);
boxplot(x,y)

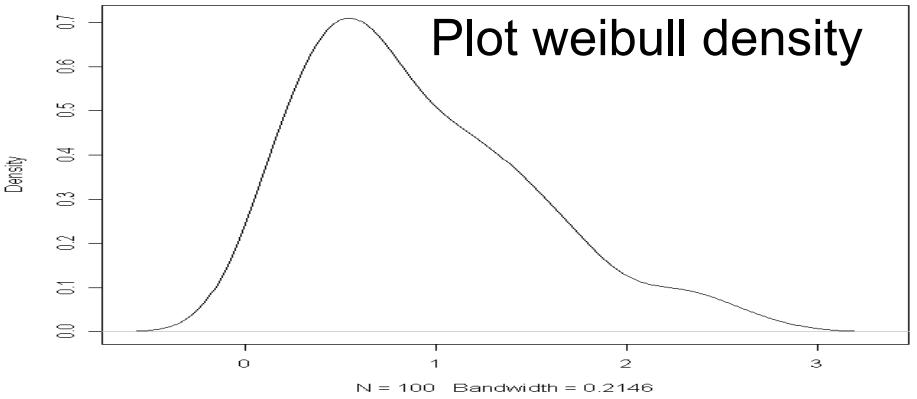


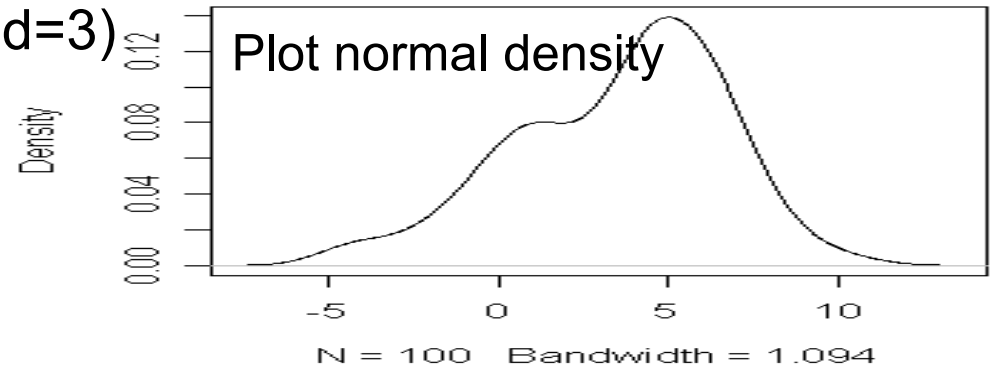


Kernel Density Plot

- Kernel Density:
  - density(x,…)
  - Kernel Density Plot:
    - plot(density(x,…))
  - Examples:
    - X= rlnorm(100,0,1)
    - plot(density(X))
    - Y= rweibull(100,1.5,1)
    - plot(density(Y))
    - x= rnorm(100, mean=3, sd=3)
    - plot(density(x))

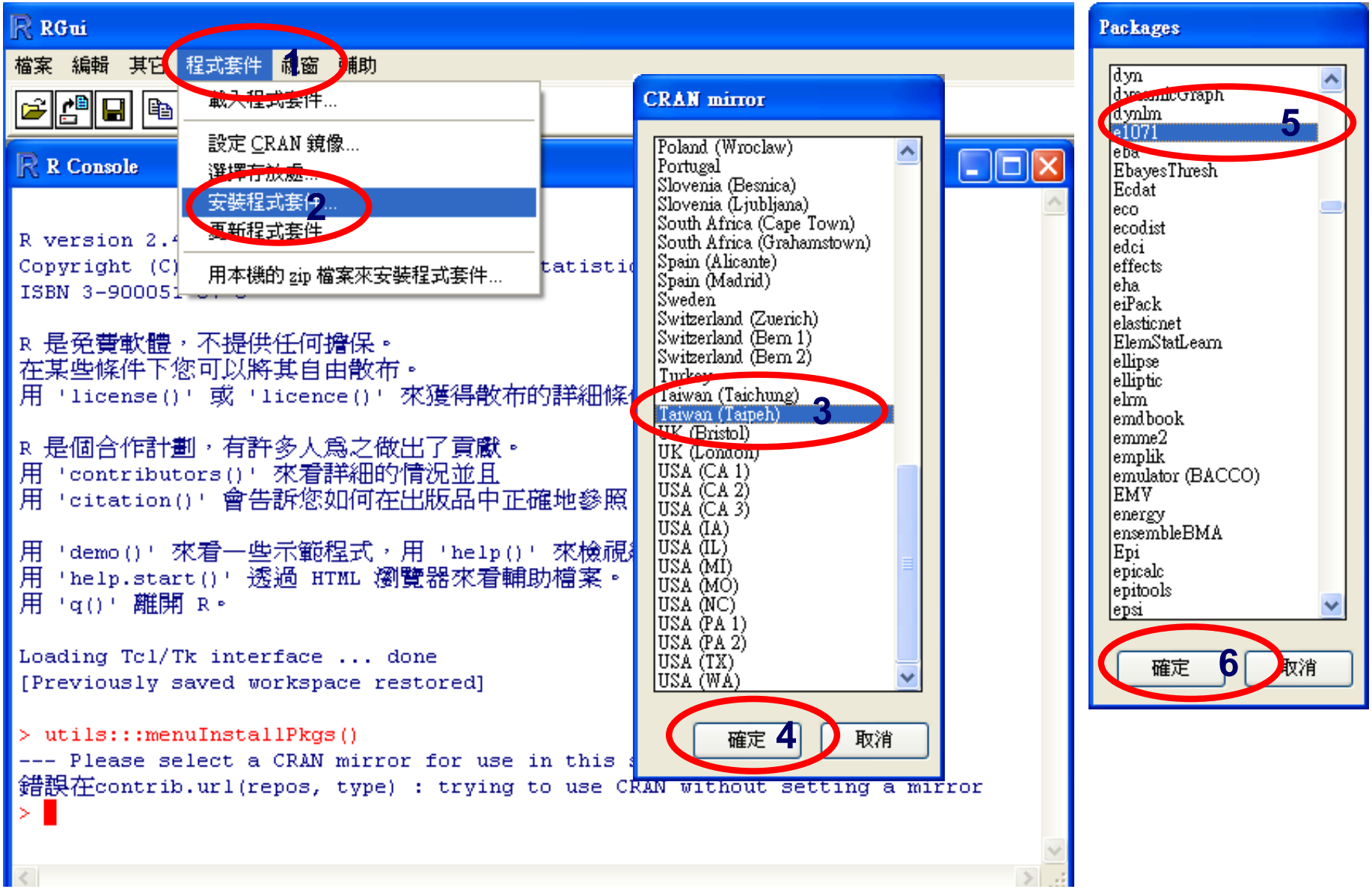




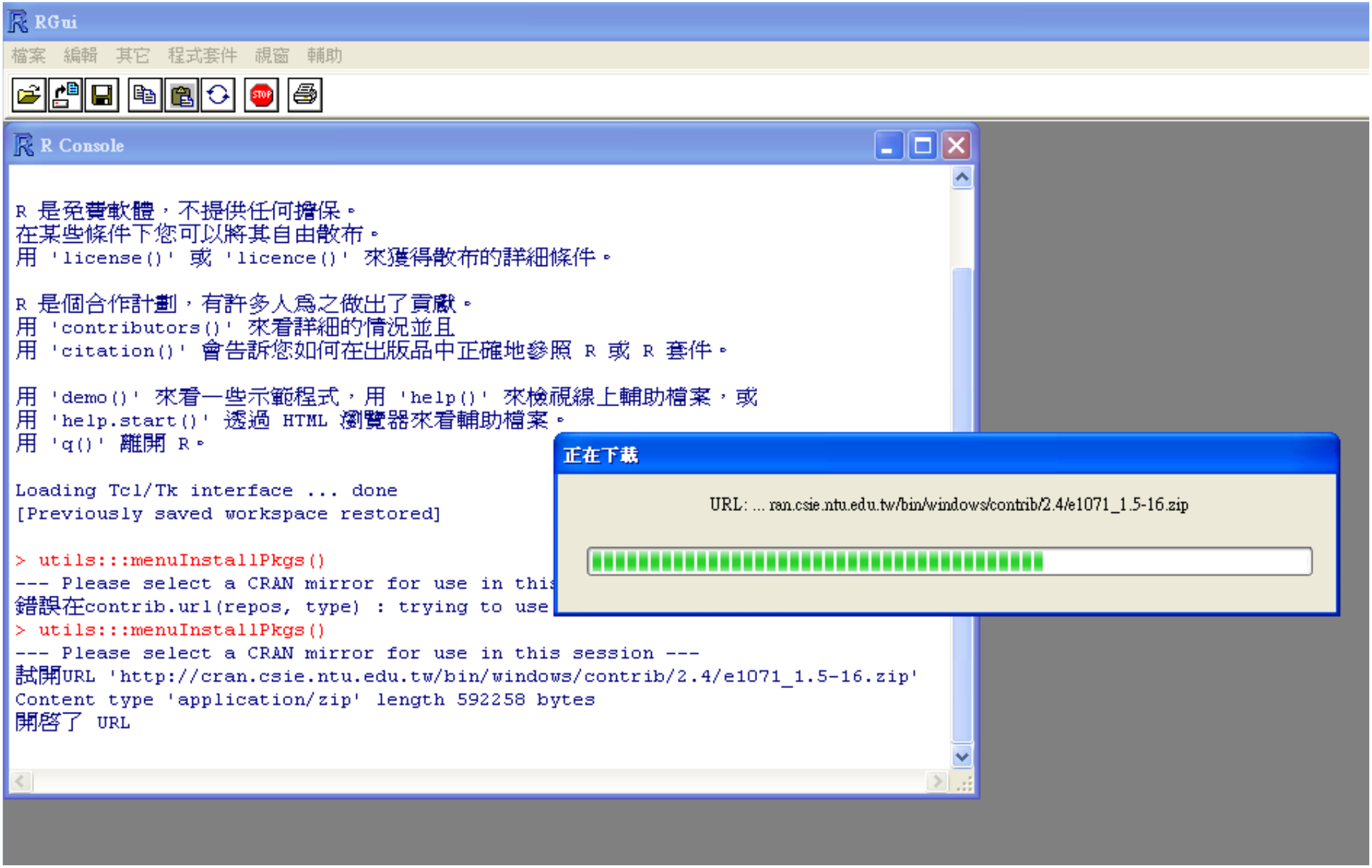


Probability Plots

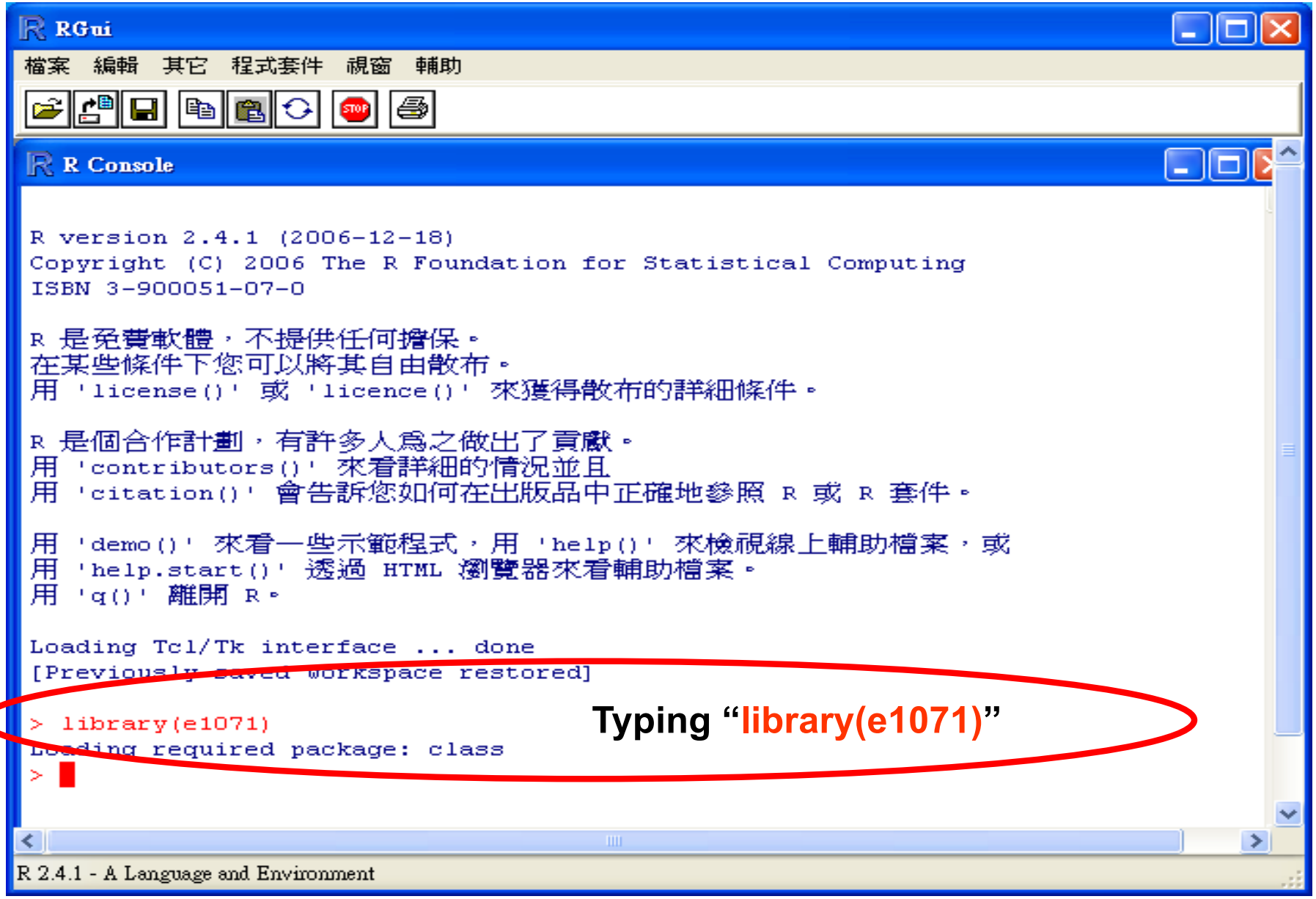
- Need installation R package : **e1071**



Installing

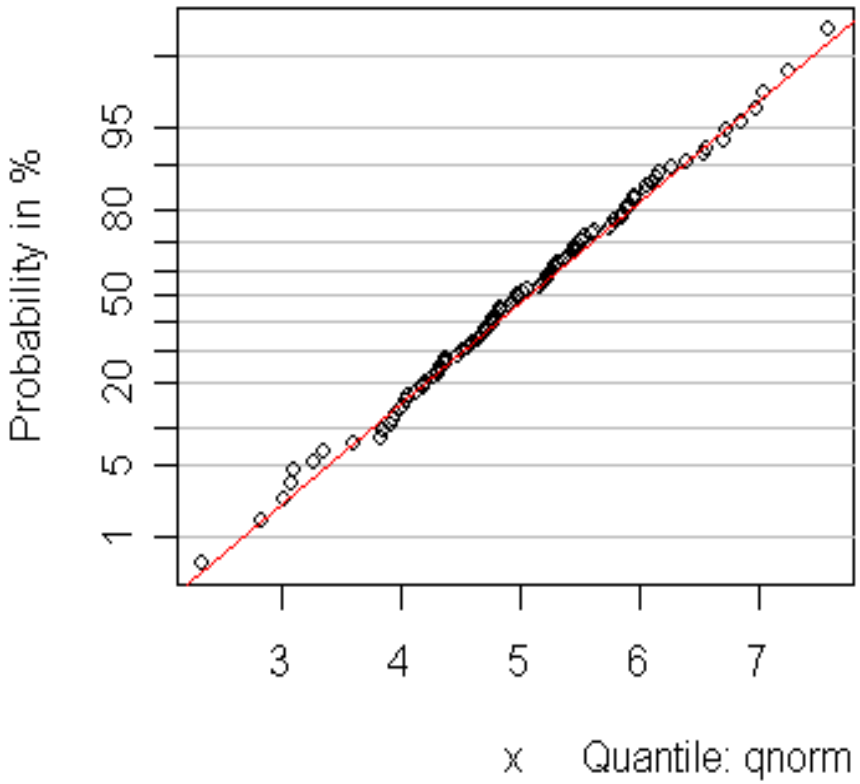

RGui
檔案 編輯 其它 程式套件 視窗 輔助
R Console
R 是免費軟體，不提供任何擔保。
在某些條件下您可以將其自由散布。
用 'license()' 或 'licence()' 來獲得散布的詳細條件。
R 是個合作計劃，有許多人爲之做出了貢獻。
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。
用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。
Loading Tcl/Tk interface ... done
[Previously saved workspace restored]
> utils:::menuInstallPkgs()
--- Please select a CRAN mirror for use in this
錯誤在contrib.url(repos, type) : trying to use
> utils:::menuInstallPkgs()
--- Please select a CRAN mirror for use in this session ---
試開URL 'http://cran.csie.ntu.edu.tw/bin/windows/contrib/2.4/e1071_1.5-16.zip'
Content type 'application/zip' length 592258 bytes
開啓了 URL
正在下載
URL: ... ran.csie.ntu.edu.tw/bin/windows/contrib/2.4/e1071_1.5-16.zip

Load Package **e1071**



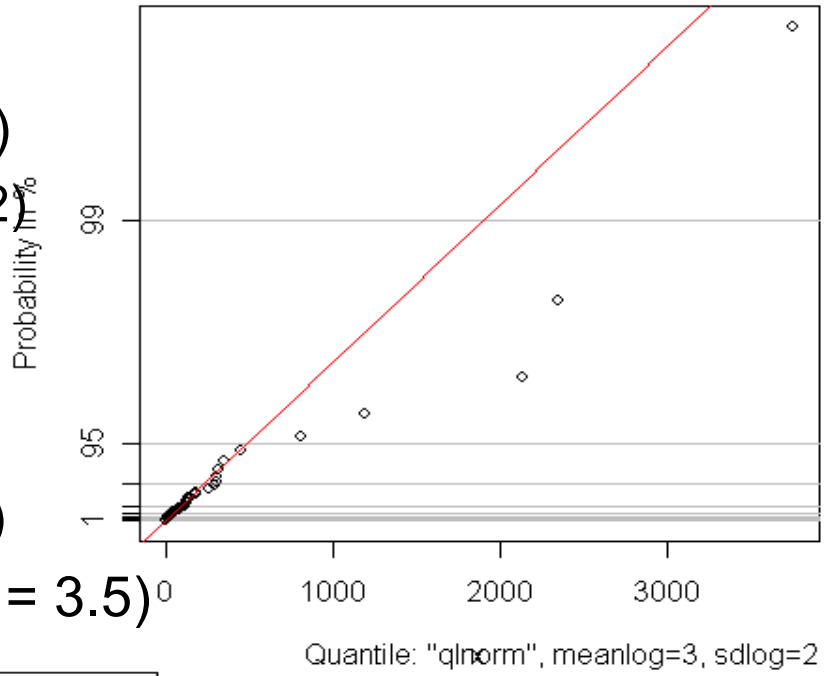
Using Probability Plot

- **Description**
  - Generates a probability plot for a specified theoretical distribution, i.e., basically a qqplot where the y-axis is labeled with probabilities instead of quantiles. The function is mainly intended for teaching the concept of quantile plots.
  - **Usage**
    - probplot(x,…)
    - Example1:
      - x <- rnorm(100, mean=5)
      - probplot(x)

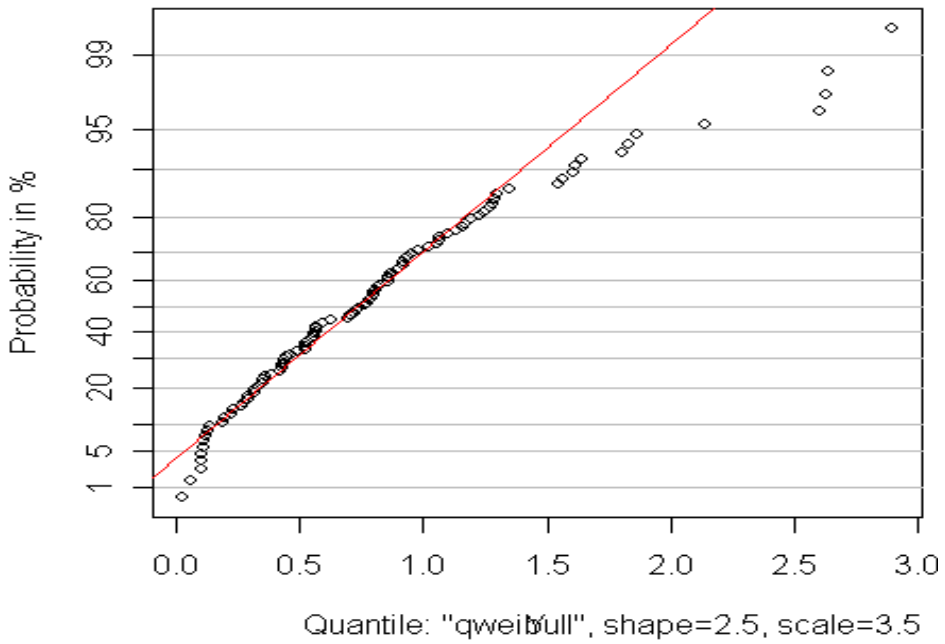


More Examples

- Example2:
  - x <- rlnorm(100,meanlog = 3, sdlog = 2)
  - probplot(x, "qlnorm", meanlog = 3, sdlog = 2)



- Example 3:
  - x=rweibull(100, shape=2.5, scale = 3.5)
  - probplot(x,"qweibull", shape=2.5, scale = 3.5)



Pareto Chart

- Load package: **library(qcc)**
  - Using command:

    pareto.chart(x, ylab = "Frequency", xlab, ylim, main, col = heat.colors(length(x)), ...)

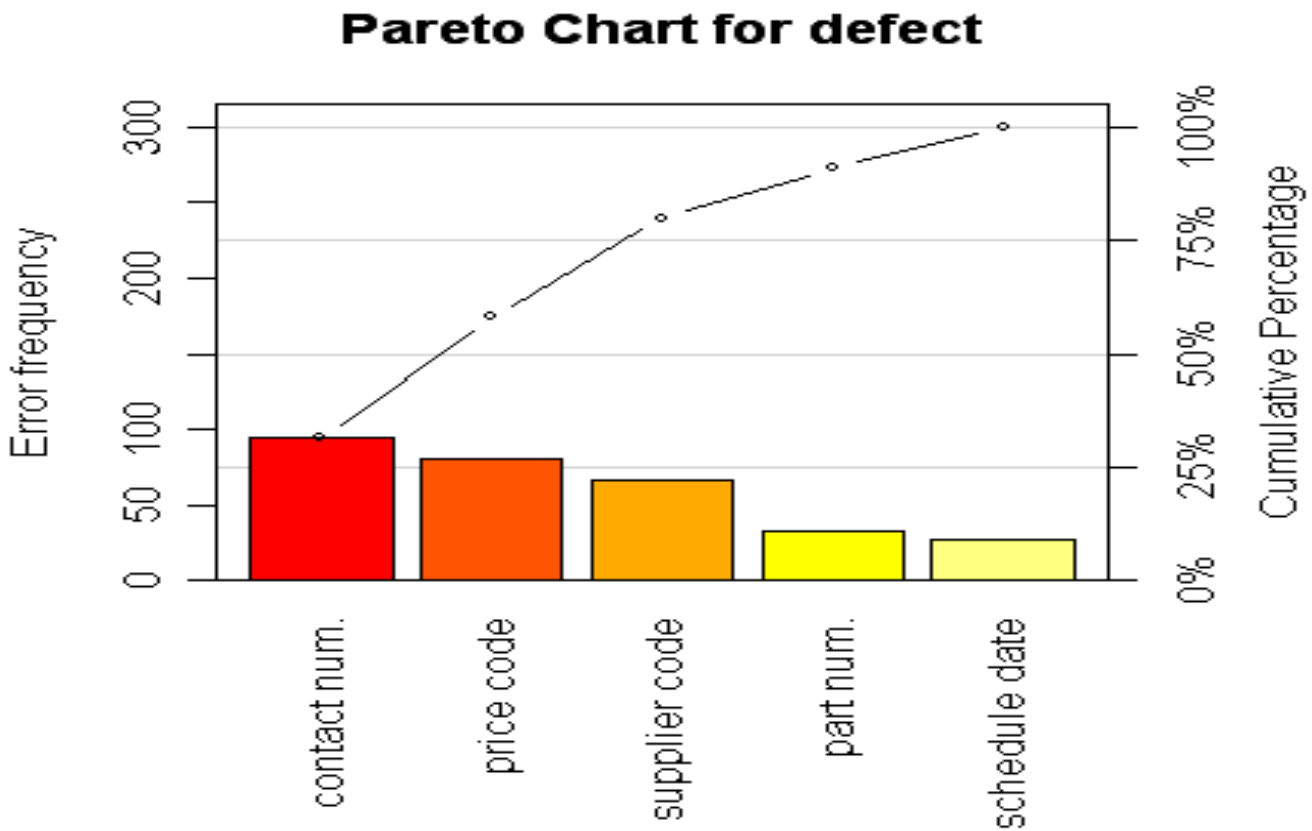
- Example:

```
defect <- c(80, 27, 66, 94, 33) # Frequency
names(defect) <- c("price code", "schedule date", "supplier code", "contact num.", "part num.") # names
pareto.chart(defect, ylab = "Error frequency")#plot
```

Pareto Chart Analysis

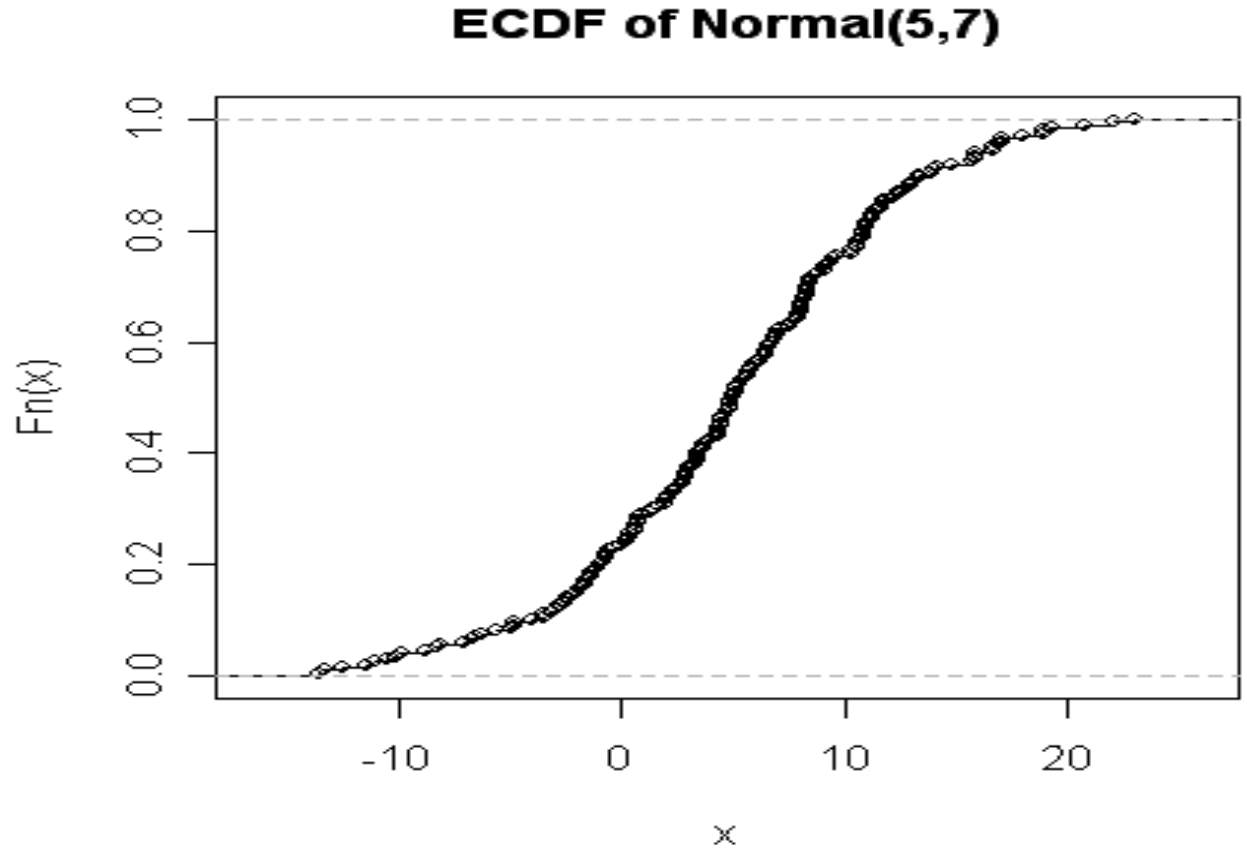
Pareto chart analysis for defect

	Frequency	Cum.Freq.	Percentage	Cum.Percent.
contact num.	94	94	31.33333	31.33333
price code	80	174	26.66667	58.00000
supplier code	66	240	22.00000	80.00000
part num.	33	273	11.00000	91.00000
schedule date	27	300	9.00000	100.00000



Empirical CDF

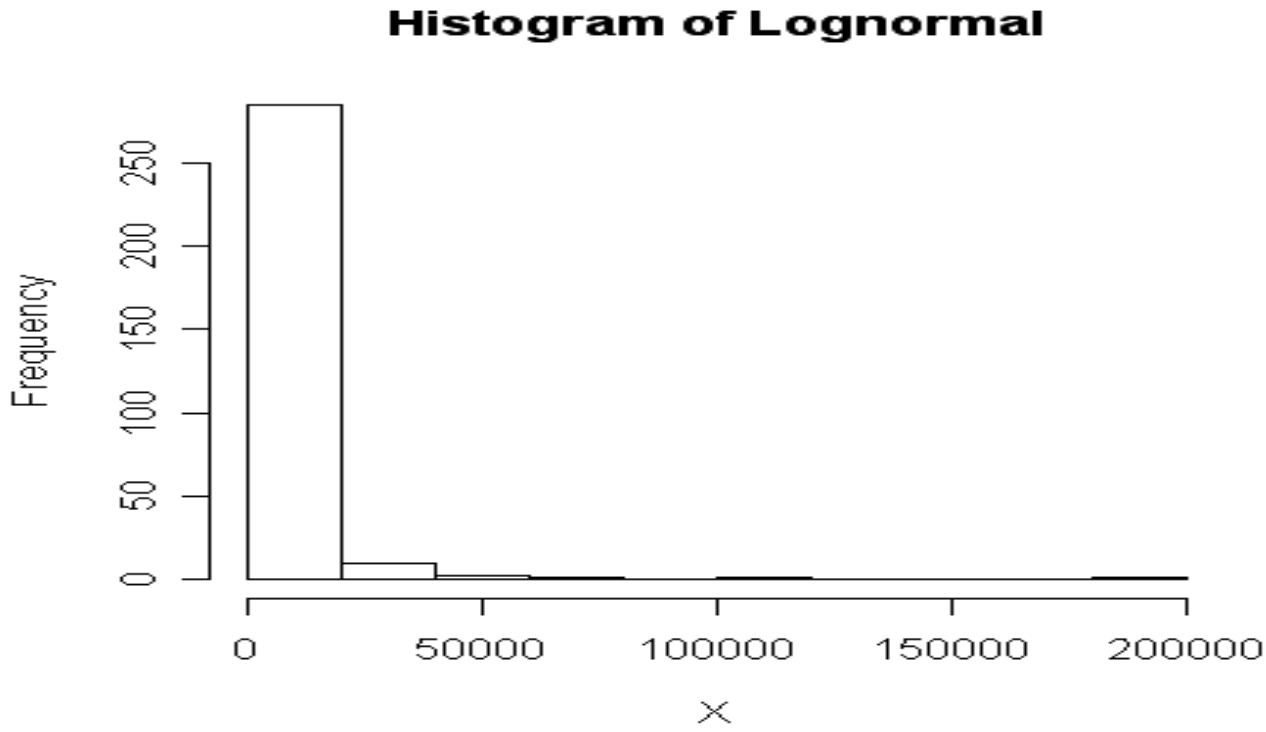
- Empirical Cumulative Distribution Function
- Usage: ecdf(x); plot(ecdf(x),…)
- Example:
  - X= rnorm(200,mean=5,sd=7)
  - plot(ecdf(X),main= “ECDF of Normal(5,7)")



Histogram (1)

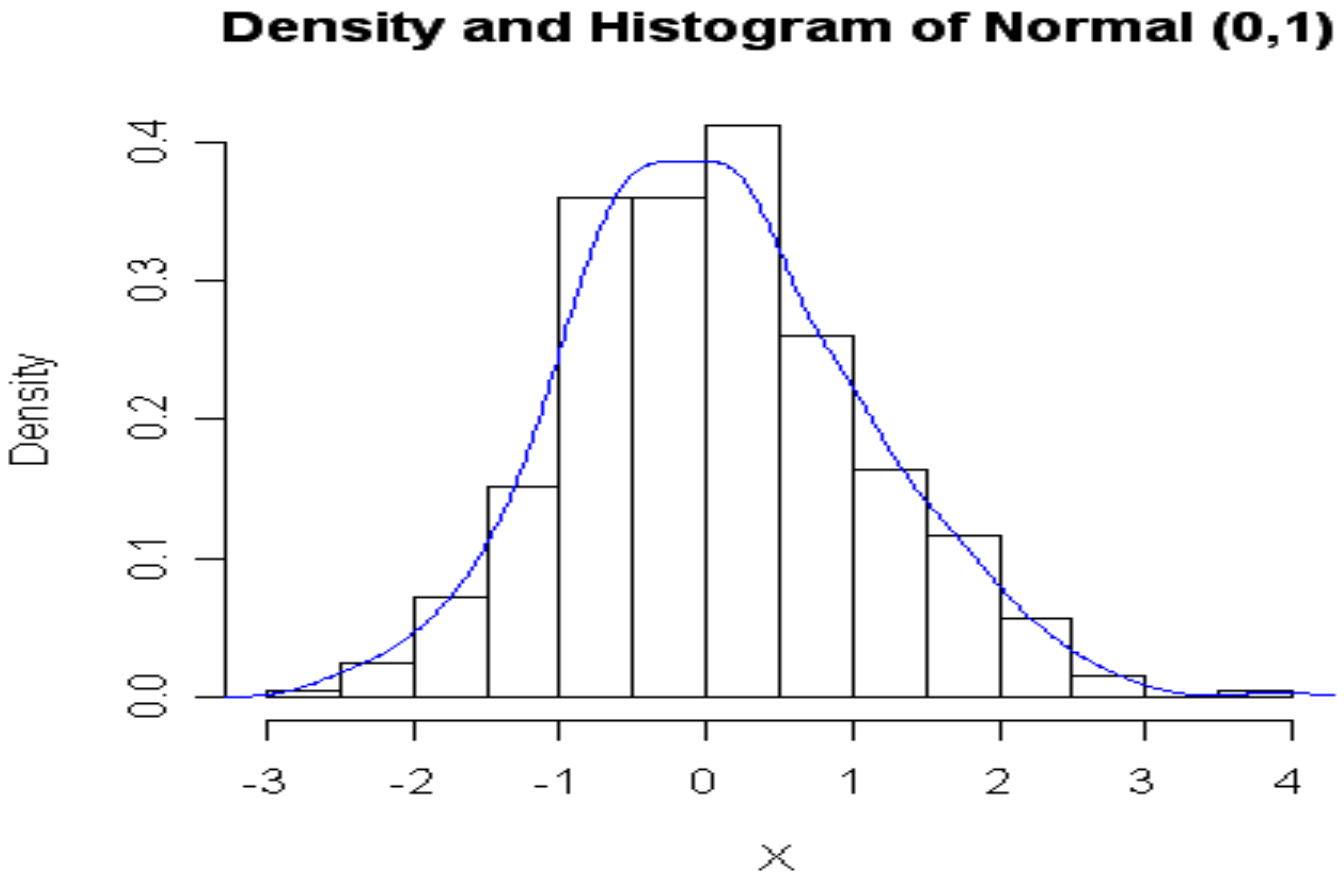
- The generic function ***hist*** computes a histogram of the given data values.

- Usage: **hist(x, probability = !freq, ...)**
- Example:
  - X=rlnorm(300,lmean=5, logsd=3)
  - hist(X) # Using default setting



Histogram (2)

- Example: Density and Histogram
  - X=rnorm(500)
  - hist(X, probability=TRUE,main="Density and Histogram of Normal (0,1)")
  - lines(density(X),col="blue")



Useful Refs.

- http://colinfay.me/intro-to-r/ordered-and-unordered-factors.html
- https://ucdavis-bioinformatics-training.github.io/2019_August_UCD_mRNAseq_Workshop/intro2R/Intro2R
- https://intro2r.com/install-r.html#linux-users