R programming

Lecture One

What is R

- It began with ‘S’ language. ‘S’ is a statistical tool developed back in the 1970s.
- R: initially written by Ross Ihaka and Robert Gentleman at Dep. of Statistics of U of Auckland, New Zealand during 1990s.
- R is a powerful, general purpose language and software environment for statistical computing and graphics.
- R is open source and free.

Data Analysis and Presentation

- The R distribution contains functionality for large number of statistical procedures.
  - linear and generalized linear models
  - nonlinear regression models
  - time series analysis
  - classical parametric and nonparametric tests
  - clustering
  - smoothing
- R also has a large set of functions which provide a flexible graphical environment for creating various kinds of data presentations.
- BioConductor for biological analysis

Obtaining R

- www.r-project.org
- MS windows
  - self extracting binary installation
  - R-3.2.2-win.exe
- Mac OS
  - R-3.2.2.pkg latest version
- Under Linux
  - Install R with one command line

```
> sudo apt-get install r-base
```

R tutorials

- http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf (A reading friendly text on R is "R for beginners")
- http://www.cyclismo.org/tutorial/R/
- http://tryr.codeschool.com/levels/1/challenges/1 (an interactive tutorial)
- http://cran.r-project.org/doc/manuals/R-intro.html
- http://math.illinoisstate.edu/dhkim/rstuff/rtutor.html (for beginners to plot figures)

Advanced references

http://cran.r-project.org/manuals.html

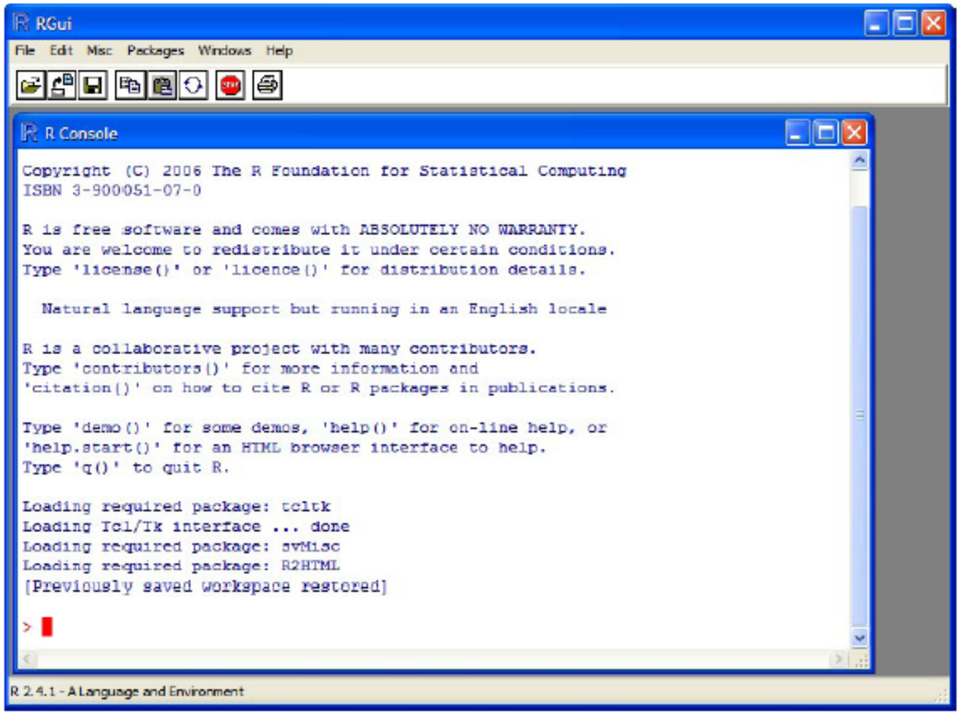
- There are several books for different topics
- It is suitable as a reference book
- One of the best text on R is “An Introduction to R”

More Tutorials

- P. Kuhnert & B. Venables, An Introduction to R: Software for Statistical Modeling & Computing http://cran.r-project.org/doc/contrib/Kuhnert+Venables-R_Course_Notes.zip
- J.H. Maindonald, Using R for Data Analysis and Graphics , http://cran.r-project.org/doc/contrib/usingR.pdf
- B. Muenchen, R for SAS and SPSS Users , http://rforsasandspssusers.googlepages.com/RforSASSPSSusers.pdf
- W.J. Owen, The R Guide , http://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf
- D. Rossiter, Introduction to the R Project for Statistical Computing for Use at the ITC , http://cran.r-project.org/doc/contrib/Rossiter-RIntro-ITC.pdf
- W.N. Venebles & D. M. Smith, An Introduction to R, http://cran.r-project.org/doc/manuals/R-intro.pdf

How to start

- Under windows, double click on the R icon


RGui
File Edit Misc Packages Windows Help
R Console
R 2.4.1 - A Language and Environment

```
Copyright (C) 2006 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: tcltk
Loading Tcl/Tk interface ... done
Loading required package: svMisc
Loading required package: R2HTML
[Previously saved workspace restored]

>
```

Figure 1.1: The R system on Windows

- R is a script program. You type commands in the console window. Results are displayed there, and plots appear in associated graphics windows.
- Similar to “matlab”, “maple”, “mathematica”, “SAS”

Using R as a calculator

- > 3+2

[1] 5

- > sqrt(2)

[1] 1.414214

- > "hello, world”

[1] "hello, world”

- Note: R is a command line program. You type commands in the console window. Results are displayed there, and plots appear in associated graphics windows.

Variables

```
> a = 49
> sqrt(a)
[1] 7
```

numeric

```
>b = "The dog ate my homework“
```

character string

```
> c = (1+1==3)
> c
[1] FALSE
```

logical

- Variables are names for the space where you save your data. They can be used for further calculations.
- primitive (or: atomic) data types in R are:
  - numeric (integer, double, complex)
  - Character and character string.
  - Logical
- out of these, vectors, arrays, lists can be built

Variables

- Data or Results of calculations can be stored or assigned to variables (or objects) using the assignment operators:
  - An arrow (<-) formed by a smaller than character and a hyphen without a space!
  - The equal character (=).

```
> a = 49
> a <- 49
```

List all variables

- > objects()
- > rm(x)   #remove the variable “x”

Function

- Functions are "self contained" modules of code that accomplish a specific task. Functions usually "take in" data, process it, and "return" a result.
- R has lots of built-in functions.

```
> y=c(1,2,3,4,5,6,7)
> sum(y)
[1] 28
> sum(y)/length(y)
[1] 4
> mean(y)
[1] 4
> sd(y)
[1] 2.160247
```

** You can get help on functions by the help command. For example:

```
>help(sum) or >?sum
```

Function

-
  ```
  > y <- rnorm(10)
  > y
  [1] -1.07521929 -1.15549677 -1.88800876
  -0.89362362
  [5] 0.60838354 -2.11006124 0.41604637
  0.52506983
  [9] -0.06416302 -0.22610929
  ```

The “rnorm” function generates random variables from the normal distribution.

How to quit

- Simply type “q()” on a command line

```
> q()        #Image can be saved to .RData
```

- R always prints a prompt (usually a right angle bracket “>”) where you can type commands.
- In my slides, if a line starts with a “>”, that line is a command for R.
- If a line does not contain a complete command, then R prints a continuation prompt (usually +).

Getting help

- R has online help
- > help.start()   #Opens help browser
- > help(dist) #get help on function dist

  or
- >?dist
- > example(dist)
- to get help about something you don’t know the exact name:

  > help.search(“keyword”)

Package

- Every function in R is in a package, and packages come with documentation.

- To use a function, you need load its corresponding package first.

- To get help on the “stats” package, you would type help(package=stats)

  This will open a help window containing one-line descriptions of all functions in the package.

Load Package

- When R starts, it loads the packages “base”, “utils”, “graphics”, and “stats”.

- For other packages, you can load them by clicking “Packages” in R window, then “Load packages...”.

- Alternatively and mostly, we use the following command line.

```
>library(“utils”)   # the package name
```

Graphics

- R includes an extensive suite of graphics tools.

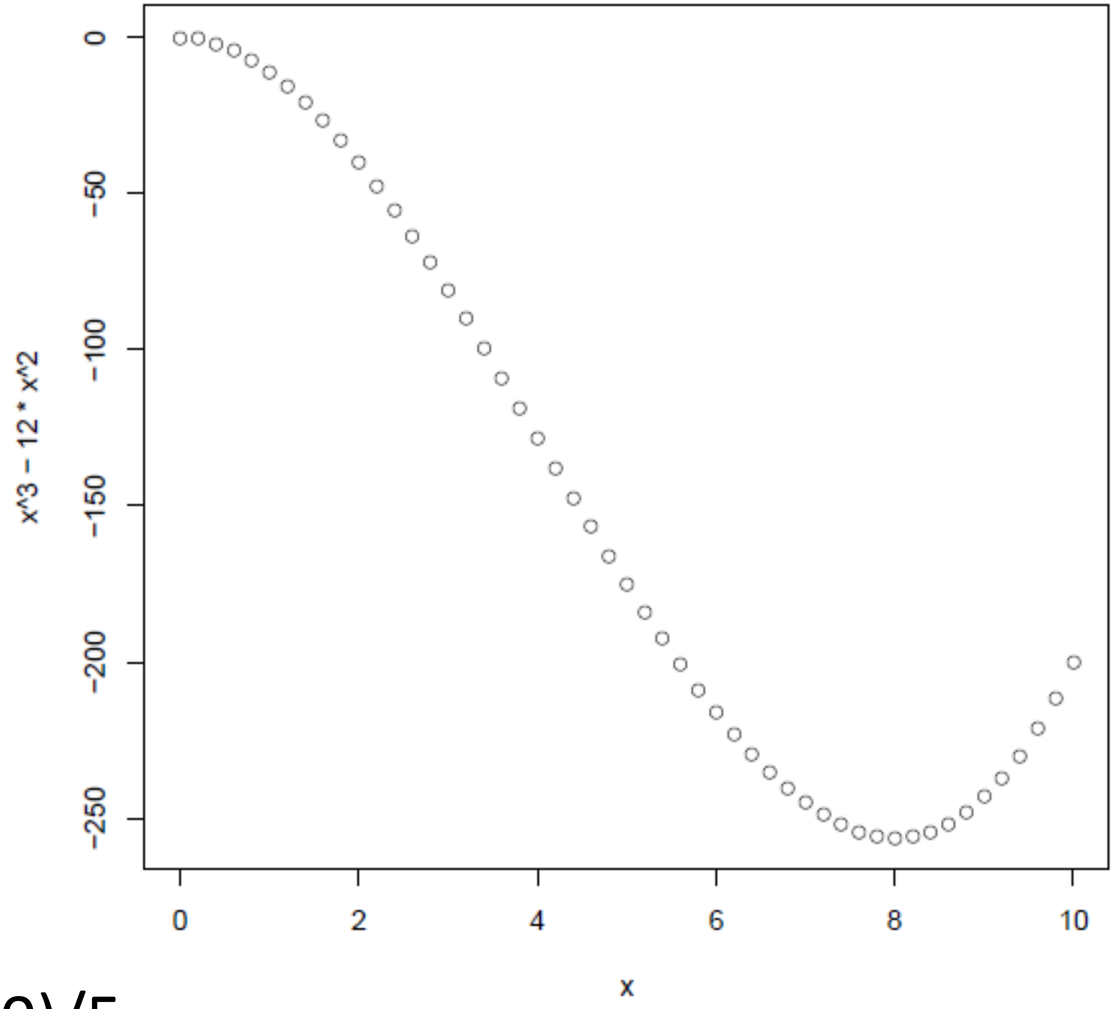
- Three steps for producing useful graphs in R;

  1. Creating the basic plot

  2. Enhancing the plot with labels, legends, colors, etc.

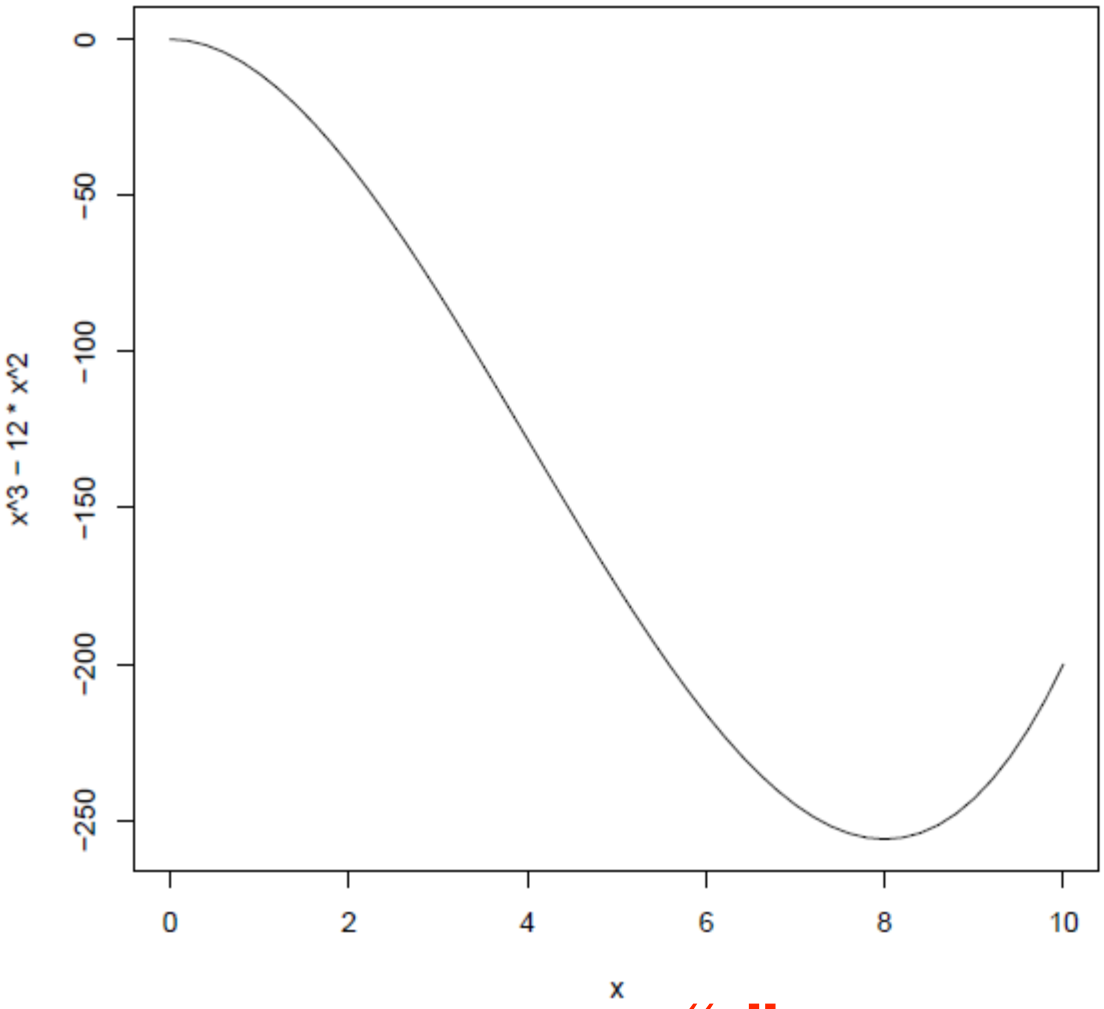
  3. Exporting the plot

Basic plot



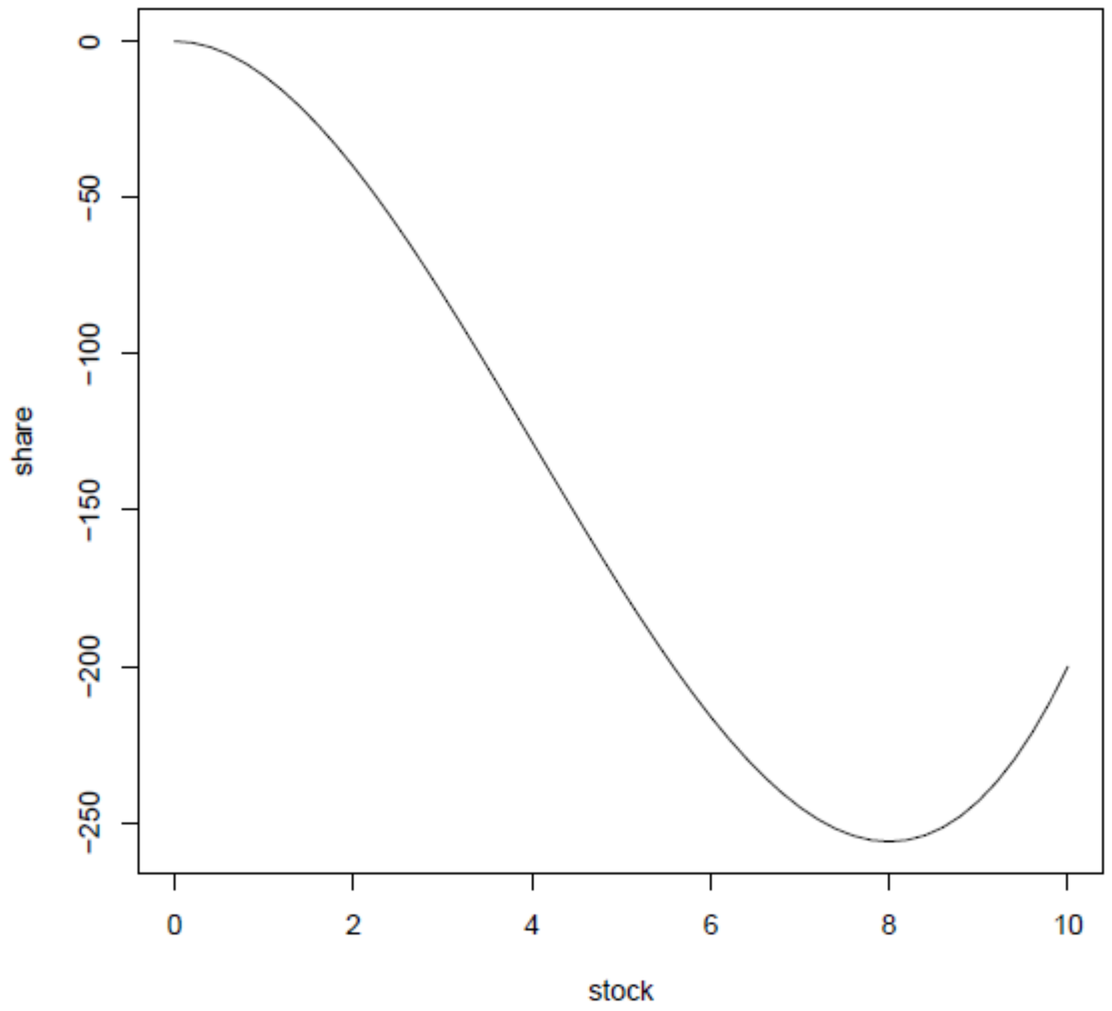
```
> x<- (0:50)/5
> Plot (x, x^3-12*x^2)
```

Changing the points to curves



> Plot (x, x^3-12*x^2, type="l")

Labeling the axes



> Plot (x, x^3-12*x^2, type="l", xlab="stock", ylab ="share")

Save your plot

- On the “File” menu of R GUI, choose “Save As ->”, which gives you several choices of file format (“pdf”, “png” or “postscript”).

Complex data structure

- Data structures
  - Vector
  - Array
  - Matrix
  - List
  - Factor

Vector

- vector: an ordered collection of data of the same type.

- x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
  # assignment operator (`<-')

- 1/x
- sort(x)
- max(x)   # Returns the largest value in vector

Adding a *name* to the vector

- food <- c(5, 10, 1, 20)

```
[1]  5 10  1 20
```

- names(food) <- c("orange", "banana", "apple", "peach")

```
orange banana  apple  peach
     5     10      1     20
```

- lunch <- food[c("apple", "orange")]

```
apple orange
    1      5
```

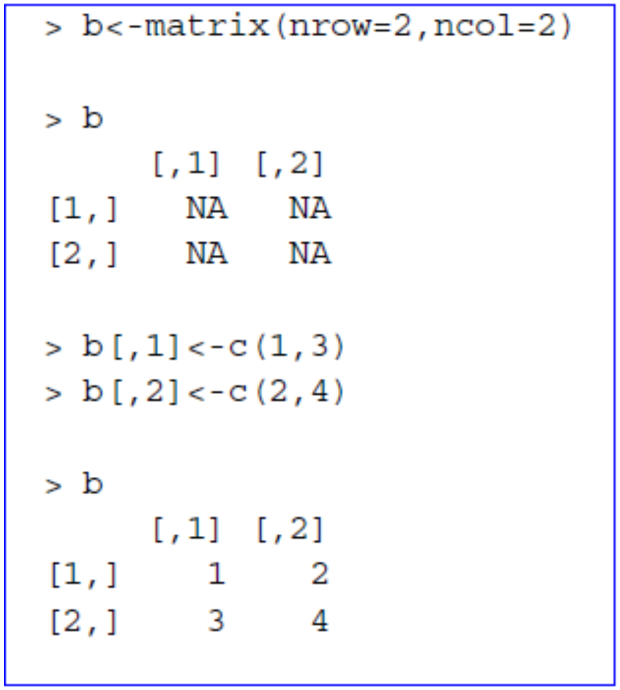
Array

- Creating an array:

  Z <- array(*data_vector*, *dim_vector*)
- x <- array(1:40, dim=c(5,4)) # Generate a 5 by 4 array.
- i <- array(c(1:3,3:1), dim=c(3,2))
- Array indexing
  x[i]
  x[i] <- 0

Arrays have to define with fixed size it will not grow dynamically , vector size can be increased dynamically and vectors are synchronized.

Matrix



```
> b<-matrix(nrow=2,ncol=2)

> b
     [,1] [,2]
[1,]   NA   NA
[2,]   NA   NA

> b[,1]<-c(1,3)
> b[,2]<-c(2,4)

> b
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

- Let A and B be two matrices:

  >A * B # element by element multiplication

  >A %*% B # matrix multiplication

Branching

```
if (logical expression) {
  statements
} else {
  alternative statements
}
```

else branch is optional

Loops

- When the same or similar tasks need to be performed multiple times; for all elements of a list; for all columns of an array;

```
for(i in 1:10) {
   print(i*i)
}

i=1
while(i<=10) {
   print(i*i)
   i=i+sqrt(i)
}
```

Reading data from files

- The read.table() function
  - To read an entire data frame directly, the external file will normally have a special form.
  - The first line of the file should have a name for each variable in the data frame.
  - Each additional line of the file has its first item a row label and the values for each variable.

```
     Price   Floor   Area   Rooms   Age   Cent.heat
01   52.00   111.0    830     5     6.2      no
02   54.75   128.0    710     5     7.5      no
03   57.50   101.0   1000     5     4.2      no
04   57.50   131.0    690     6     8.8      no
05   59.75    93.0    900     5     1.9      yes
...
```

Reading data from files

- HousePrice <- read.table("houses.data", header=TRUE)

```
 Price   Floor   Area  Rooms  Age  Cent.heat
52.00   111.0    830     5    6.2      no
54.75   128.0    710     5    7.5      no
57.50   101.0   1000     5    4.2      no
57.50   131.0    690     6    8.8      no
59.75    93.0    900     5    1.9     yes
   …
```

From A Comma Delimited Text File

```
# first row contains variable names, comma is separator
# assign the variable id to row names

> mydata <- read.table("c:/mydata.csv", header=TRUE,
sep=",", row.names="id")
> x = read.delim(“filename.txt”)
> x=read.csv(“filename.txt”)
```

Save data into a file

```
> write.table(x, file=“x.txt”, sep=“\t”)
```

Note: For other specific file type, we need use specific packages and functions to read and save data files.