Introduction to the R program

Gilles Guillot

Department of Applied Mathematics and Computer Science
Technical University of Denmark
Lyngby, Denmark
gigu@dtu.dk
www2.imm.dtu.dk/~gigu/

Some useful documentation for the R program

- The R primer by Christopher G. Green 46 pp. http://students.washington.edu/cggreen/uwstat/rprimer/
- R for Beginners by Emmanuel Paradis 76 pp. http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
- R reference card by Tom Short 4 pages http://cran.r-project.org/doc/contrib/Short-refcard.pdf
- MATLAB/R Reference, D. Hiebeler, 2001: http://cran.r-project.org/doc/contrib/Hiebeler-matlabR.pdf

How to use these slides

- Open R studio
- Read slides one by one
- For each slide, type the R commands in the R script window of R studio then execute it to see what happens
- You can also try to modify the commands to see what happens
- Refer to The R primer by Christopher G. Green or your TA when you need more detail about a specific point

Getting started

R can be used as pocket calculator to

- do basic operation
- create variables
- visualize them

Always type your R commands in a file that can be saved at the end of a session, using either

- R built-in editor (Windows or Mac)
- RStudio http://www.rstudio.com
- Emacs and ESS http://ess.r-project.org (more convenient for code development involving different languages)

Mode and class of an object

Mode

An R object can have mode `"logical"`, `"character"`, `"integer"`, `"double"`, `"complex"`.

`mode(x)` returns the mode of `x`

Class

An R object can have class `"vector"`, `"matrix"`, `"data.frame"`, `"list"`, `"function"`

`class(x)` returns the class of `x`

Vectors

Creating vectors

```
x = -3:7 # consecutive integers
y = seq(from=-pi,pi,by=0.1)
z = c(x,y) # Concatenates x and y
```

Operating on vectors

```
x = x+1 # add up 1 to each entry of x
x = 2*x # multiply by 2 each entry of x
y = x^2 # squares each entry of x
x[3] # 3-rd entry of x
x[2:5] # entry with index 2 to 5
x[c(F,T,F,F,F,F,F,F,F,F,F)]
x[-2] # all but second entry
is.vector(x) # checks the class of x
length(x) # length of x
x>0 # logical
```

Matrices

Creating matrices

```
x = matrix(nrow=3,nc=4,data=1:12)
z = rbind(x,y)
```

Manipulating matrices

```
x = x+1 # add up 1 to each entry of x ...
x[2,3] # one entry
x[2,] # second row
x[,3] # third column
x[2:3,1:2] # a sub-matrix
colnames(x) = c("math","physics","biology","chemistry")
rownames(x)
rownames(x) = c("Peter","John","william")
dim(x) # dimension of x
```

Data frames

Definition: data frame

A rectangular object whose row or columns may be of different mode, e.g. combining columns of character strings, of logicals and of numerics.

Creating a data frame

```
y = read.table(file=
       "http://www2.imm.dtu.dk/courses/02443/slides2013/data/data.txt",
       header=TRUE)
is.data.frame(y) # checking
```

Operating on data frames

Pretty much like matrices but they are some restrictions due to the "composite" nature of data frames

Lists

Definition: list

An R object made of several other R objects.

Creating a list

```
x = c("apple","ananas",cherry")
y = seq(-pi,pi,.1)
z = read.table(file=
       "http://www2.imm.dtu.dk/courses/02443/slides2013/data/data.txt",
       header=TRUE)
mylist = list(fruits=x,angles=y,data=z) # creates a list called mylist
mylist$[[1]] # first component of the list
mylist$fruits # same via the name
```

Basic operations

Basic operations on numeric objects

Operations on numeric objects: `+`, `-`, `*`, `/`, `%`
Matrix product: `A %*% B`
Operations on logical objects: `&`, `|`, `!`

Conditionnal execution

if() else

```
x = -999
if(x < 0)
  { y = 1 }else
  { y = -1}
```

for()

```
n = 10 ; x = rep(NA,n)
for(i in 1:n) { x[i] =  i^2 } # correct

x = (1:n)^2 # correct and much faster than line above
```

while()

```
x = -5
while(x<0) { x = x+2 }
x
```

R functions library

All common math. and stat. functions are available.
Remember that they are vectorized:

```
x = 1:10
y = exp(x)
x = matrix(nrow=10,col=10,data=1:100)
y = log(x)
```

Writing R functions

An example

```
myfunc = function(x,y)
   {
    x = cos(x)
    y = sin(y)
    z = x*y
    return(z)
   }

x = 1:10
y = -10:-1
myfunc(x,y)
x # x is not modified
z # z does not exist
  # (use <<- instead of = for global variable)
```

Visualizing data

The `plot()` function

```
x = seq(-pi,pi,.1)
y = sin(x)
z = cos(x)
plot(x,y) # simplest plot
plot(x,y,type='l',col=2,lty=2,lwd=1.5,  # nicer with
     xlab='angles',ylab='Measurement')  # optional arguments
lines(x,z,col=3,lty=1,lwd=4)

par(mfrow=c(2,3)) # splot plotting window
plot(x,y,type='l')
plot(x,z,type='l')
```

To go beyond

The R on-line help

```
help.start() # open the help in a web browser

help(hist) #  open help on R function hist
```

As always, Google is your best friend...

Exercise 1

1. Open RStudio
2. Type some of the example commands e.g. of slide on matrices in the top-left window
3. Save these commands as a file with a .R extension
4. Install the R package knitr by typing:
   `install.packages('knitr')`
5. Close and re-open RStudio, open the R file previously created and execute the commands it contains
   - line-by-line
   - globally
   - check that commands can be recalled with Up-Arrow button
   - check the command completion with the tab button
6. Check the content of the memory
   - with `ls()`
   - by clicking on the various object in the top-right window
7. Remove all the objects with `remove(list=ls())`
8. Save your session as an html notebook [top-right button in top-left window]
9. Close RStudio and check that you have saved your session as an R file and an html file. Check how this html file looks like under your web browser [should look good with chrome]

Exercise 2

1. Create a vector `x` containing 10 integers
2. Store its first 5 entries in a vector named `y`
3. Create a vector `z` containing each second entry of `x` [hint: use `seq(from=...,to=...,by=...)`]
4. Create `x` as
   `x = matrix(nr=4,nc=5,data=1:20)`
   and add 10 to the first line then 20 to the third column.
5. Define `y` as the transpose of `x` [see R function `t()` in on-line help]
6. Define `z` as the matrix product of `x` and `y`
7. Read the help of read.table of [`? read.table`]
8. Create a data frame from the file `data_ex2.txt`[1] making use of the arguments `file, header, , row.names` and `na.strings` (assuming -999 stands for a missing value)
9. Correct the name of the third column
10. Compute the means by subject then by student. [See `? mean`]
11. Create a vector of `logicals` of `length` 4 containing `TRUE` for Physics grades larger than 10.
12. Create a 4x3 matrix of `logicals` containing `TRUE` for grades larger than 10.

[1] On `www2.imm.dtu.dk/courses/02443/slides2014/data/`

Exercise 3

1. Use a `for()` loop to create a vector containing the 100 first numbers of the Fibonacci sequence defined there
2. Write a function returning the real roots (if any) of a second order polynomial

Exercise 4

The file

`http://www2.imm.dtu.dk/courses/02443/projects/data/Eurostoxx50.txt`

contains stock prices of ABN Amro Holding, Aegon, Ahold Kon. and Air Liquide.

1. Load them as a data.frame
2. Plot the variation of the prices as a function of time on the same plotting window with different colors and line types.
3. Add axes labels and title.
4. Add a legend. [See `? legend`].