

Introduction to R

Downloading

For your home/office computer, you can get binaries (ready-to-run) for R by going to <http://www.r-project.org/> clicking on CRAN and picking one of the mirror websites to download from, then click on **Windows**, **base**, and follow the instructions. In the TCC labs, R is already installed.

Getting help

If you know the name of the topic, you can type

```
?topic
```

in the R console. There's also browsable help available by typing in

```
help.start()
```

Intro

Both R and Matlab are based on similar programming principles. Also, they are both case-sensitive.

A distinct feature of R (partly, Matlab, too) is that it's built out of *packages*. The basic packages we'll need are **basic**, **graphics** and **stats**. A lot of other packages are user-contributed (buyer beware!).

To load a package, type in `require(package_name)`.

The R program can be typed in line by line (use the up arrow to recall previously entered lines), but I usually type it in some editor (e.g. Notepad) and then paste the contents into the R console.

Other useful things:

- list all the variables: `ls()`
- change the working directory: *File* → *Change dir*.
- executing a script: if a program is kept in a file, you can run it by using `source('file_name')` (although it will not print anything unless specifically requested by `print` command). I find it useful when I re-use several functions, so I don't have to input large text every time.
- Saving graphics output: when in graph window, *File* → *Save As*. This is useful for saving a picture for writing up your homework or a project. Postscript and PDF formats are useful when typesetting in T_EX.

Comparative table

R	Matlab	comments
<pre>x <- 2 # X if you want printing { ... ; }</pre>	<pre>x = 2 % X; if you don't want printing , ... end</pre>	<p>assignment operator commentary</p> <p>group of operators</p>
<pre>plot(x,y) plot(x,y, type="l") contour(X,Y,Z)</pre>	<p>plotting</p> <pre>scatter(x,y) plot(x,y) contour(X,Y,Z)</pre>	<p>scatterplot line plot contour plots</p>
<pre>c(1,2,3) rbind(c(1,2,3),c(4,5,6)) A <- matrix(0, n1, n2) dim(A) x+y, x*y, x^2, sin(x) ... t(x) A %% B A[1:5,2:4] A[1:5,] solve(A,b) 1:n seq(a,b,by = step)</pre>	<p>matrix/vector operations</p> <pre>[1 2 3] [1 2 3; 4 5 6] A = zeros(n1,n2) size(A) x+y, x.*y, x.^2, sin(x) x' A*B A(1:5,2:4) A(1:5,:) b\A 1:n a:step:b</pre>	<p>concatenation assigning row by row assigning null dimensions element-wise operations on vec. transposition vector/matrix multiplication subsetting a matrix take rows 1 through 5 solving a linear system sequence of integers (same) sequence with increment <i>step</i></p>
<pre>X <- read.table("U: /Data.dat") write.table(X,....)</pre>	<p>interface</p> <pre>load Data.dat; (reads into a matrix with same name) save Data.dat X -ascii</pre>	<p>loading data</p> <p>saving data</p>
<pre>abs, sin, exp, log, ... rnorm(n,mean,var) runif(n,a,b) mean(x), sd(x), var(x) cor(x,y), cov(x,y)</pre>	<p>some functions</p> <pre>abs, sin, exp, log, ... normrnd (MU,SIGMA) unifrnd(A,B) mean(x), std(x), var(x) corrcoef(x,y), cov(x,y)</pre>	<p>Normal random variable(s) Uniform random variable(s) descriptive stats cov., corr.</p>
<pre>if (q==1) x <- 5 else x <- 2 for (i in 1:n){ ... } f <- function(x){ ... out}</pre>	<p>control flow</p> <pre>if (q==1), ... end; else ... end; for i = 1:n, ... end function out = f(x)</pre>	<p>if ... then</p> <p>loops function assignment</p>