

UNIVERSITY OF WOLLONGONG
School of Mathematics and Applied Statistics
STAT902. Advanced Data Analysis

ASSIGNMENT 3

Due: 5:00pm Monday 29th March, 2010.

Local students: please put under lecturer's door.

Remote students: please post by this date.

The first part of this assignment involves performing an optional computing lab on R. You do not have to hand anything in for this part. The 'hand-in' part of the assignment follows the lab. Please be aware that, since R runs on three different computing platforms (Linux, MacOS X and Windows) it is difficult to give system-dependent instructions that cover all cases.

Introductory R Lab

WARNING: This lab may still have a few errors. If anything doesn't work then please let me know by e-mail. Any errors found will be conveyed by e-mail. Be aware that R (like any computing language) is very precise. If you mis-type one of the commands or e.g. leave out a bracket then there is a good chance the command will fail. So you need to check carefully that you follow the instructions in the lab exactly.

1. Log into a computer that has R installed and start an R session. (The process for doing this differs for each operating system and installation. In Linux it is usually: type R at the command line. In Windows it is usually: click on the R icon.)

2. Some Basic Operations

- (a) Type `x <- 4` and hit Enter. From now on I will just put the line to type (hitting Enter is assumed).
- (b) `y <- 7`
- (c) `sin(x*y)*sqrt(log(3*x+y))`
- (d) You have just computed $\sin(4 \times 7)\sqrt{\log(3 \times 4 + 7)}$ and should get the answer: 0.4648572
- (e) `A <- cbind(c(6, -2), c(3, 11))`
- (f) A
- (g) The previous command should print the 2×2 matrix A to the screen with columns (6,-2) and (3,11).
- (h) `b <- c(4, -13)`
- (i) `A%*%b`
- (j) You have just done the matrix multiplication

$$\begin{bmatrix} 6 & 3 \\ -2 & 11 \end{bmatrix} \begin{bmatrix} 4 \\ -13 \end{bmatrix} = \begin{bmatrix} -15 \\ -151 \end{bmatrix}$$

- (k) `x <- seq(0, 1, length=201)`
- (l) `y <- sin(2*pi*x^3)`
- (m) `plot(x, y, type="l")`

Note that this command involves the letter lower-case L ("l") and not the number one ("1").

- (n) You have just plotted the function $y = \sin(2\pi x^3)$ over the range $0 \leq x \leq 1$.
- (o) Some fancier graphics:

```
data(volcano)
z <- 2 * volcano
x <- 10 * (1:nrow(z))
y <- 10 * (1:ncol(z))
par(bg = "slategray")
persp(x, y, z, theta=135, phi=30, col="green3",
      scale=FALSE, ltheta=-120, shade=0.75,
      border=NA, box=FALSE)
```

- (p) Type `help(persp)`. This gives documentation on the function `persp()` that you just used for doing a perspective plot.
- (q) Type `q()` and hit Enter to quit R.

3. Fitting a Linear Model

- (a) Open a web browser and go to the web-page:
`www.uow.edu.au/~mwand/web902/codedata.html`
- (b) In the web browser, click on the link for `mitsub.txt`. This should open a file containing data on the price and age of 41 Mitsubishi cars. Save this as an **ordinary text file** named `mitsub.txt` in a directory on the computer you are currently using. In Linux this is straightforward and the best directory is the one in which you open R. In Windows saving a text file named `mitsub.txt` may not be straightforward — it seems to depend on the current settings. You can try saving it as a text file directly in a directory such as the Desktop. If that fails then open a text file on the Desktop and cut and paste the data into the file. Save the file to have the name `mitsub.txt`. Note that for some configurations of Windows the file may actually be named `mitsub.txt.txt` (i.e. Windows sneakily puts a `.txt` extension on the file name that you chose). So you need to be aware of this possibility when it comes to reading in the file a few steps later. Before proceeding make sure the file that you have saved looks exactly like what is on the web-site. In particular `age`, `price` should be in the **first line** of the file – even though certain web browsers may give the impression that the first line is blank.
- (c) In the web browser, open the file called `mitsub.Rs`. This is an R script for regression analysis of the Mitsubishi data set.
- (d) Start an R session.
- (e) If using R in Windows you need to change the directory (i.e. folder) so that it corresponds to the one where you saved `mitsub.txt`. Suppose this is the Desktop. In the File menu scroll to `Change dir...` and then change to the directory where you saved `mitsub.txt`.
- (f) Ignoring, for now, the parts of the R script `mitsub.Rs` involving `wait()`, cut and paste each line uncommented line (i.e. without the hash symbol at the front)

into the R session. This will give you a step-by-step appreciation of what the script does. Note that if `mitsub.txt` really is called `mitsub.txt.txt` (due to Windows sneakily adding the `.txt` and not necessarily telling you about it) then the line `mitsub <- read.table("mitsub.txt", header=T, sep=", ")` should be replaced by

```
mitsub <- read.table("mitsub.txt.txt", header=T, sep=", ")
```

- (g) Save the file `mitsub.Rs` as a text file in the same directory where you saved `mitsub.txt`. Note the comments in (b) above about file saving and naming in Windows.
- (h) Type `source("mitsub.Rs")`. This should run the whole script again; illustrating the use or re-use of a saved script of R instructions.
- (i) Type `q()` to exit R.

4. Creating a Function

Most of R involves calls to functions. Thousands of functions already exist in R. Examples are `sqrt()` for square-root, `plot()` for simple two-dimensional plots and `solve()` for matrix inversion. However, it is also useful to be able to create your own function.

- (a) Type

```
CtoF <- function(x)
  return(1.8*x+32)
```

You have just created a function called `CtoF()` for conversion from degrees Celsius to degrees Fahrenheit. To get a feeling for `CtoF()` type

- i. `CtoF(24)`
- ii. `CtoF(40)`
- iii. `CtoF(0)`
- iv. `CtoF(-40)`

- (b) Functions are not always mathematical. The following one takes the name of a chemical element as a character string and returns its abbreviation.

Type

```
elementAbbrev <- function(fullName)
{
  if (fullName=="copper") return("Cu")
  if (fullName=="iron") return("Fe")
  if (fullName=="lead") return("Pb")
  if (fullName=="gold") return("Au")
  if (fullName=="silver") return("Ag")
  if (!any(fullName==c("copper","iron","lead","gold","silver")))
    stop("argument not supported by this function")
}
```

To get a feeling for `CtoF()` type

- i. `elementAbbrev("lead")`
- ii. `elementAbbrev("copper")`
- iii. `elementAbbrev("chicken")`

5. Creating a List

Lists are a very useful data structure supported by R. Enter the following commands to produce a list of attributes about some Australian animals:

```
AusAnimals <- list(kangaroo=list(family="marsupial", legs=2, cover="fur"),
                  emu=list(family="bird", legs=2, cover="plumes"),
                  wombat=list(family="marsupial", legs=4, cover="fur"),
                  echidna=list(family="monotreme", legs=4, cover="spikes"))
```

List components are accessed using the \$ sign. To get a feeling for this type

- (a) `AusAnimals$kangaroo`
- (b) `AusAnimals$wombat$legs`

The 'Hand-in' Part of the Assignment

Your submission should include all output generated from the following questions.

1. What is the R command for evaluation of

$$108\sqrt{e^{\sin(34)} - 3\ln(78)}?$$

What is this number to 3 decimal places?

[2 marks]

2. Write a R function named `state.info()` which takes as input the first digit of the typical post code of a state of Australia (e.g. this digit is 2 for New South Wales) and outputs the full state name and its capital city as a single character string (e.g. "New South Wales, Sydney").

[4 marks]

3. Modify `state.info()` to obtain a new function called `state.info.list()`. The difference is that the output is now a list with two components named `$state` and `$capital`.

[3 marks]

4. Write R code to create a list named `my.info` with the following components

- (a) `$name` - a character string with your full name.
- (b) `$universities` - an array of character strings containing each of the universities where you have either worked or studied.
- (c) `$other.info` - another list with the components: `$study` - a character string with your major area of study. `$birth.month` - the month number corresponding to your birthday. `$post.code` - the post code of your home address.

[4 marks]

5. Let M be the matrix:

$$M = \begin{bmatrix} 5 & 85 & 5 & 1 \\ 6 & 26 & 86 & 3 \\ 4 & 24 & 94 & 9 \\ 1 & 11 & 41 & 3 \end{bmatrix}$$

Write R code for creation of M using:

- (a) the `cbind()` function (look it up using the on-line documentation)
- (b) the `rbind()` function
- (c) the `matrix()` function

[4 marks]

6. Write R code for drawing plots of the functions

$$f_j(x) = \sin(j\pi x), \quad j = 1, 2, 3, 4, 5, 6$$

over the unit interval $0 \leq x \leq 1$ on a single graph. The `lines()` function is appropriate for this. However, each curve should have a different line thickness (use e.g. `lwd=3`) **and** a different colour. (use e.g. `col="pink"`).

[4 marks]

7. This question involves the **cheese** data set on the course web-site (`cheese.txt`). Write an R script for that fits the linear regression model

$$\text{taste} = a + b_1 \text{acetic} + b_2 \text{lactic} + b_3 \text{H2S} + \text{error},$$

prints a summary table of the fit and draws a histogram of the residuals. (Hint: `residuals(fit)` is a fast way of getting residuals from an `lm()` fit object).

[6 marks]