

22S:164:Lab1

August 30th, 2007

1. Brief Introduction to R

- (1) **R** is a command line statistical package, which means that the user types a statement requesting a computation or a graph, and it is executed immediately. There are a lot of packages of functions available. In this course, we will use a specific one called `alr3`, which includes datasets from the book as well as a number of functions that implement techniques in the textbook.
- (2) How to download and install **R** on your home computer
 - Go to <http://cran.us.r-project.org/> and download the **R** source on your computer.
 - Install **R** on your computer.
 - Go to <http://www.stat.umn.edu/alr/R.html> and follow the instruction to install the `alr3` package.
- (3) How to run **R** Program on Linux
 - Log on to the Linux system using your Linux ID (not your Hawk ID) and password.
 - Bring up a terminal window by clicking the black screen icon on the panel in the top of the screen.
 - Type `R` in the terminal window and click ENTER. You'll see a prompt that looks like this: `>`, which means you are running **R**.
 - To quit **R**, type the command `q()` and hit ENTER.

Note: Everything in Linux and in **R** is case-sensitive. That means that `spot` is not the same as `SPOT` or `Spot`. Please be careful when you type commands.

2. R basics

- (1) Access a data file from the package `alr3`
 - Type the command `>library(alr3)`, which opens the package for your use.
 - To read a particular file, say `forbes.txt`, type `>data(forbes)`. This will create a data frame with the name `forbes`.
 - Then type `>forbes`, the data will be listed.
 - To access a particular variable in `forbes`, say `Pressure`, just simply type `>forbes$Pressure`.
 - Now, type `>attach(forbes)`, then `>Pressure`. Compare this with the previous command; you will see the same results. The function `attach` allows reference to the variables in `forbes` without specifying the data frame.
 - To detach the file, simply type `>detach(forbes)`. Try to type `>Pressure` again, see what will happen now.
- (2) Read a data file of other type
 - To read a `.txt` data file, use the command

```
> newname <- read.table("filename", header=TRUE),
```

 where the complete path to the file is required. The argument `header=TRUE` indicates that the first line of the file has variable names. Check `>help(read.table)` for more argument options.
 - To read the data directly from the internet, type

```
> d <-read.table(url("http://www.stat.umn.edu/alr/data/htwt.txt"),header=TRUE)
```

A new data frame named `d` is created to **R**'s search path.
 - **R** can't work with Excel files directly, and you must save the Excel file as a `.csv` file. Then read the `.csv` file with the command

```
>data <-read.csv("filename',header=TRUE),
```

 where once again the complete path to the file is required.

(3) R help facility

- Enter the command `>help.start()` to bring up an HTML-based help system.
- If you want to search for help for a specific command, say `rnorm`, type `>help (rnorm)` or `>?rnorm`. This will bring up the same help documentation. To quit the help, type `q`.

(4) Basic Statistical commands

- Type `>rand <-rnorm(100, 0,3)`. This generates 100 random numbers normally distributed with mean 0 and standard deviation 3. To see these numbers, just type the variable name, `rand`. Also, you can do some calculation based on this variable. Try `>rand/3+2`.
- Try `>mean(rand)`, `median(rand)`, `sd(rand)`, and `summary(rand)`.
- There are several functions for looking up critical values and significant levels for standard distributions like normal, t, F and chisquare distribution. Try `>qt(0.3,5)`, `>pf(3,2,3)`. Type `?qt`, `?pf` to learn more commands regarding the standard distributions.

(5) Scatterplots

- Type the following commands to get the two-dimensional scatterplots of file heights
`> data(heights)`, `> attach(heights)`, `> plot(Mheight,Dheight)`.
Compare it with `> plot(Dheight,Mheight)`. Notice that the vertical axes are different. See more argument options in help document.
- To save the graph, we need to use another library: `s164`. Type `>library(s164)`, then use one of the following commands to save the graph as pdf, or png, or jpg file: `>save.pdf("filename")`.
- Also, you can select the points you want to appear by using following commands, for example, try following commands
`> sel <-(57.5 < Mheight)& (Mheight<=58.5)| (62.5 < Mheight)& (Mheight<=63.5)`
`> plot(Mheight[sel],Dheight[sel])`
- R allows you to draw several graphs in one window. To do this, we will use the `par` function, which sets global parameters for graphical windows. Try
`>par(mfrow=c(1,2))`, `> plot(Mheight,Dheight)`, `> plot(Mheight[sel],Dheight[sel])`.
You will see two graphs in one row and two columns. Compare it with
`>par(mfrow=c(2,1))`, `> plot(Mheight,Dheight)`, `> plot(Mheight[sel],Dheight[sel])`. This will hold two graphs in two rows and one column. Close the graph window if you want to quit.

(6) Linear regression

- The `lm` function can be used to carry out simple linear regression. Try the `forbes` example, type `>data(wblake)`, `>attach(wblake)`, `> m0 <- lm(Length~Age)`. To see the intercept and slope of the regression model, just type `>m0`.
- The regression line can be drawn by using the `abline` function. Graph the scatterplot by `>plot(Age,Length)`, then add the regression line by `>abline(m0)`.
- The `lines` command adds lines to an existing plot. Type
`> lines(tapply(Length,Age,mean),lty=2)`. The `tapply` function applies the function `mean` to the variable `Length` separately for each value of `Age`. `lty2` refers to the dashed line.
- Try the residual plot by `>plot(Age, residuals(m0))`. A horizontal dashed line is added by specifying intercept `a` and slope `b` equal to zero. The command is `>abline(a=0,b=0,lty=2)`.