**Lab01 – Notes**

Functions

> funcname(input1,input2)

Where inputs (arguments) tell R *how* to run the function

To concatenate values

> c()

Saving and storing values in variables is achieved using an arrow or the equals sign:

> x <- [ c(2, 4, 6, 8)]

-or-

> y = [ c(2, 4, 6, 8)]

To remove *all objects* in the environment

> rm( list = ls())

Variables can be set *within* other variable definitions

> z = matrix( **data** = c(1, 2, 3, 4), **nrow** = 2, **ncol** = 2)

By default, R populates matrices *column-wise* (top to bottom, left to right). To instead populate a matrix by *row*, the ‘byrow’ modifier is used:

> matrix(c(1,2,3,4), 2,2, byrow=FALSE)

and vise-versa for column-wise population.

To generate a vector of random *normal* variables:

> x = rnorm()

Arguments:

*n* refers to the number of variables to generate

*mean* sets the average value of the distribution

*sd* sets the standard deviation of the distribution

e.g.

> y = x + rnorm(50, mean = 50, sd = .1)

To compute the correlation between two sets of numbers:

> cor(x,y)

NOTE: By default, rnorm creates standard normal random variables with a *mean* of 0 and a *standard deviation* of 1.

When it is desirable to have your code reproduce the *exact same set of random numbers*:

> set.seed()

Receives an arbitrary integer argument.

Additional functions:

Average:

mean()

Variance:

var()

Standard Deviation:

sd()

NOTE: var()^2 returns the standard deviation as well

Graphics

We will often want to save the output of an R plot. The command depends on the file *type* that we would like to create:

> pdf(“filename.pdf”)

> jpeg(“filename.jpeg)

The function dev.off() indicates to R that we are done creating the plot. Alternatively, we can simply copy and paste the plot window.

Contour plots: contour() produces a plot for representing three-dimensional data (*similar* to a topographical map). The function takes 3 arguments:

1. A vector of the *x* values (the first dimension)
2. A vector of the *y* values (the second dimension)
3. A *matrix* whose elements correspond to the *z* value (the third dimension) for each pair of (*x,y*) coordinates.

> y = x

> f = outer(x,y,function(x,y)cos(y)/(1+x^2))

> contour(x,y,f)

> contour(x,y,f,nlevels=45,add=T)

> fa=(f-t(f))/r

> contour(x,y,fa,nlevels=15)

The image() function works the same way as contour(), however, it produces a *color-coded* plot whose colors *depend on the z value.* AKA – “heatmap”.

Alternatively, persp() can be used to produce a three-dimensional plot. Arguments *theta* and *phi* control the angles at which the plot is viewed.

> image(x,y,fa)

> persp(x,y,fa)

> perp(x,y,fa,theta=30)

>persp(x,y,fa,theta=30,phi=20)

>persp(x,y,fa,theta=30,phi=70)

>persp(x,y,fa,theta=30,phi=40)

The summary() function produces a numerical summary of each variable in a particular data set.

For *qualitative* variables such as name, R will list the number of observations that fall in each category.