**Introduction to R - Basics**

Appendix A of my “Analysis of Categorical Data with R” book contains much of the same content as in my introduction to R notes.

R can be downloaded from <https://cran.r-project.org>. For Windows users, the direct link to the newest version is [http://cran.r-project.org/bin/windows/base](http://cran.r-project.org/bin/windows/base/). Select the “Download R \*.\*.\* for Windows” link. You can simply execute the file on your computer to install (all the installation defaults are o.k. to use). Both a 32-bit and 64-bit version of R will be installed.

Basics of R

The R Console window is where commands are typed.



The Console can be used like a calculator. Below are some examples:

> 2+2

[1] 4

> qchisq(0.95,1)

[1] 3.841459

> pnorm(1.96)

[1] 0.9750021

> (2-3)/6

[1] -0.1666667

> 2^2

[1] 4

> sin(pi/2)

[1] 1

> log(1)

[1] 0

Results from these calculations can be stored in an *object*. The <- is used to make the assignment and is read as “gets”.

> save <- 2+2

> save

[1] 4

The objects are stored in R’s *workspace*. When you close R you will be asked if you would like to save or delete them. If you are familiar with SAS, this is kind of like the SAS WORK library, but R gives you the choice to save them.

To see a listing of the objects, you can do either of the following:

> ls()

[1] "save"

> objects()

[1] "save"

To delete an object, use rm() and insert the object name in the parentheses.

Functions

R performs calculations using functions. For example, the pnorm() and the log()commands used earlier are functions. Writing your own function is fairly simple. For example, suppose you want to write a function to calculate the standard deviation. Below is an example where 5 observations are saved to an object using the *concatenate* or *combine* function c(). A function called sd2() is written to find the standard deviation simply by using the square root of the variance. The sd2 object is now stored in the R database.

> x <- c(1,2,3,4,5)

> sd2 <- function(numbers) {

 sqrt(var(numbers))

 }

> sd2(x)

[1] 1.581139

Note that there already is a function in R to calculate the standard deviation, and this function is sd().

When a function has multiple lines of code in it, the last line corresponds to the returned value. For example,

> x <- c(1,2,3,4,5)

> sd2 <- function(numbers) {

 cat("Print the data \n", numbers, "\n")

 sqrt(var(numbers))

 }

> save <- sd2(x)

Print the data
1 2 3 4 5

> save

[1] 1.581139

Note that the cat() function is used to print text and the \n character tells R to go to a new line.

Help

To see a listing of all R functions which are “built in”, open the Help by selecting HELP > HTML HELP from the main R menu bar.



Under REFERENCE, select the link called PACKAGES. All built in R functions are stored in a package.



We have been using functions from the base and stats packages. By selecting stats, you can scroll down to find help on the pnorm() function. Note the full syntax for pnorm() is

pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p

 = FALSE)

The q value corresponds to the 1.96 that was entered earlier. So

> pnorm(1.96)

[1] 0.9750021

> pnorm(q = 1.96)

[1] 0.9750021

> pnorm(q = 1.96, mean = 0, sd = 1)

[1] 0.9750021

all produce the same results. The other entries in the function have default values set. For example, R assumes you want to work with the standard normal distribution by assigning mean = 0 and sd = 1 (standard deviation).

The q, mean, sd, … items within the parentheses of pnorm are referred to as *arguments*. Values like 1.96 are referred to as *argument values*.

If you know the exact name of the function, simply type help(function name) at the R Console command prompt to open its help. For example,

> help(pnorm)

results in



Working with vectors

Many R functions are set up to return multiple results when used with *vectors*. For example,

> pnorm(q = c(-1.96, 1.96))

[1] 0.02499790 0.97500210

The pnorm() function finds the probability a standard normal random variable is less than -1.96 and the probability a standard normal random variable is less than 1.96.

Below are some simple examples showing how R works with vectors.

> x <- c(3.68, -3.63, 0.80, 3.03, -9.86, -8.66,
 -2.38, 8.94, 0.52, 1.25)

> y <- c(0.55, 1.65, 0.98, -0.07, -0.01, -0.31,

 -0.34, -1.38, -1.32, 0.53)

> x

 [1] 3.68 -3.63 0.80 3.03 -9.86 -8.66 -2.38 8.94

 0.52 1.25

> y

 [1] 0.55 1.65 0.98 -0.07 -0.01 -0.31 -0.34 -1.38

 -1.32 0.53

> x + y

 [1] 4.23 -1.98 1.78 2.96 -9.87 -8.97 -2.72 7.56

 -0.80 1.78

> x \* y

 [1] 2.0240 -5.9895 0.7840 -0.2121 0.0986

 2.6846 0.8092 -12.3372 -0.6864 0.6625

> mean(x)

[1] -0.631

> x - mean(x)

 [1] 4.311 -2.999 1.431 3.661 -9.229 -8.029

 -1.749 9.571 1.151 1.881

> x \* 2

 [1] 7.36 -7.26 1.60 6.06 -19.72 -17.32

 -4.76 17.88 1.04 2.50

Elementwise addition and multiplication is performed by using + and \*, respectively. Also, when a single value is involved in the mathematical operation, this operation is applied to each element of the vector.

The reason for R to perform calculations in this manner is to make the code easier to write than it would be otherwise. For example, other languages may require x and y to be added in the following way

> x[1]

[1] 3.68

> y[1]

[1] 0.55

> x[1] + y[1]

[1] 4.23

> x[2] + y[2]

[1] -1.98

> x[8] + y[8]

[1] 7.56

Packages

If you want to use functions that are in other packages, you may need to install and then load the package into R. For example, a commonly used package is car. While in the R console, select PACKAGES > INSTALL PACKAGE(S) from the main menu.



A number of locations around the world will come up. Choose one close to you (I usually choose USA (IA), which is at Iowa State U.). Next, the list of packages will appear. Select the car package and select OK.

The package will now be installed onto your computer. This only needs to be done once per computer. To load the package into your current R session, type library(package = car) at the R Console prompt. This needs to be done only once in an R session. If you close R and reopen, you will need to use the library() function again.

These packages are installed from the Comprehensive R Archive Network (CRAN).

Characters

Object names can include periods and underscores. For example, “mod.fit” could be a name of an object and it is often said as “mod dot fit”.

R IS CASE SENSITIVE!