**Outline**

1. Basic introduction to R

2. Making data: vector, matrix, list, data frames

3. Loading data into R

4. Generate random data

5. Graphics

**Introduction**

R is a cross-platform language and environment for programming, statistical computing, graphics, and a host of other things. R is also free – it is distributed under the GNU General Public License that covers many open source projects.

There are two components to R that you should download, install, and have opened at least once before the first lab session.

* The R Environment: The “base” of R available from the link:  
  <https://cran.revolutionanalytics.com>
* Rstudio: A Graphical User Interface to facilitate coding in R from the link:  
  <https://www.rstudio.com/products/rstudio/#Desktop>

Install the R environment first, then Rstudio. You MUST have Rstudio up and running before the first lab. If you have problems contact your TA.

Rstudio: This is a Graphical User Interface for R

Console: Where you type the commands (CASE SENSITIVE)

Environment: Identifies the objects (Objects: “items” in the workspace – once declared, they can be called or manipulated) that are available in the current workspace

History: All of the commands you have issued

Save history: Easy way to record what you have done

Plots: Graphics show up here

Export: Save a graphic as a .png

Help: self explanatory. Also, type ?<command name> to get help

**#1. Making data#**

#Data structure: vector, character strings, matrix, list, data frames.#

**#Vector#**

x <- c(1,3,2,5) # Assignment operator: <- (like an arrow)#

length (x)

mode(x)

y <- “abc” #only one element in the vector#

length(y)

mode(y)

z <- c(“abc”, “29 88”) #two elements in the vector#

length(z)

mode(z)

ls() #The ls() function allows us to look at a list of all of the objects, such as data and functions, that we have saved so far.#

rm(x,y)

#The rm() function can be used to delete any that we don’t want.#

ls()

**#Matrix#**

#matrix: the function rbind (for row bind) can build a matrix from two vector#

?rbind

m<-rbind(c(1,4), c(2,2))

n<-cbind(c(1,4), c(2,2))

x=matrix(data=c(1,2,3,4), nrow=2, ncol=2)

x

w<-matrix(seq(1,12),4,3)

w

y=matrix(seq(1,16),4,4, byrow=TRUE)

y

#If byrow is FALSE (the default), the matrix is filled by columns, otherwise the matrix is filled by rows#

#matrix selection#

y[1,2]

# The first number after the **open-bracket symbol [** always refers to the row, and the second number always refers to the column#

y[1, ] #row 1#

y[, 2] # column 2#

y[3:4,2:3] #row 3 and 4; column 2 and 3#

y[, 2:3]

y[-c(1,3),-c(1,3,4)] #we don’t want row 1 to 3; we don’t want column 1,3,4#

dim(y)

**#Data Frame#**

#data frame: a typical data set contains data of different modes#

d<-data.frame(list(kids=c(“Jack”, “Rose”), ages=c(12,10)))

d$ages

**#List#**

#list is also a container for values (like vector), but its contents can be items of different data types#

x<- list(u=2, v=”abc”)

x$u

# Vector with numerics from 1 up to 10

my\_vector <- 1:10

# Matrix with numerics from 1 up to 9

my\_matrix <- matrix(1:9, ncol = 3)

# The first element of the data frame d

my\_df <- d[1,]

# Construct list with these different elements:

my\_list <- list(my\_vector,my\_matrix,my\_df)

**#2. Loading data into R#**

getwd ()

setwd ("the pathway of your folder")

Auto = read.table("Auto.txt")

names(Auto)

fix(Auto) # can be used to view the data in a spreadsheet like window#

dim(Auto)

class(Auto)

summary(Auto)

Auto=read.table("Auto.txt",header=T,na.strings="?")

#the option na.strings tells R that any time it sees a particular character or set of characters (such as a question mark), it should be treated as a missing element of the data matrix#

dim(Auto)

class(Auto)

summary(Auto)

Auto=na.omit(Auto)

#to use the na.omit() function to simply remove these rows#

plot(cylinders, mpg)

plot(Auto$cylinders, Auto$mpg) # Auto$cylinders is to get a list#

#attributes: parts of an object: like dataframe$column#

attach (Auto)

plot(cylinders, mpg)

cylinders = as.factor(cylinders)

#The as.factor() function converts quantitative variables into qualitative variables.#

**#3. Generate random data, graphics#**

x<- runif (50) #random uniform on 0, 1#

y<- rnorm (50) #random normal variable# #By default, rnorm() creates standard normal random variables with a mean of 0 and a standard deviation of 1#

mean(y)

var(y)

y<- rnorm (50, mean=10, sd=2)

z= y+rnorm(50,mean=50,sd=0.1)

#if we want to reproduce our random results, we need to set seeds before generating random variable#

set.seed(3)

ser.seed(123456)

y=rnorm(100)

**#Graphics#**

x=rnorm(100)

y=rnorm(100)

plot(x,y)

# plot(x,y) produces a scatterplot of the numbers in x versus the numbers in y.#

plot(x,y, xlab="this is the x-axis",ylab="this is the y-axis", main="Plot of X vs Y")

par(mfrow=c(2,1))

plot(x,y)

hist(y)

#The pairs() function creates a scatterplot matrix i.e. a scatterplot for every scatterplot pair of variables for any given data set. We can also produce scatterplots matrix for just a subset of the variables#

pairs(Auto)

pairs(∼ mpg + displacement + horsepower + weight + acceleration , Auto)

**Resource:**

A wealth of online resources exist to help you learn R (in addition to the built in help features).

For help conducting statistical analyses, IDRE is a stellar resource: <http://www.ats.ucla.edu/stat/r/>

The “Data Analysis Examples” are particularly helpful if you are familiar with/translating between other statistical languages like Stata: <http://www.ats.ucla.edu/stat/dae/>

UT has a deal with Lynda.com that allows access to their tutorials for free, for example:

A beginner’s introduction to R:

<http://www.lynda.com/R-tutorials/Up-Running-R/120612-2.html>

A tutorial with a focus on statistics:

<http://www.lynda.com/R-tutorials/R-Statistics-Essential-Training/142447-2.html>

And a tutorial on using R as a coding language and coding across multiple languages:

<http://www.lynda.com/R-tutorials/Code-Clinic-R/372541-2.html>

There are also resources for getting a little further “under the hood” of R by exploring the nature of data structures, routines, and objects in a pleasantly tongue-in-cheek manner:

<http://www.burns-stat.com/pages/Tutor/R_inferno.pdf>

Video tutorials here:

<http://jeromyanglim.blogspot.com/search/label/R>

and on Youtube.