Review of R

- Download most recent versions of R and R-Studio, https://www.r-project.org/ select “CRAN” https://www.rstudio.com/
- A very readable but more detailed review of R “The Art of R Programming” by Normal Matloff.
- A pdf file of the book is posted on the class website.

R Studio Basics

- New File-> R Script or Text file
- Session->Set Working Directory->To Source File Location
- Session->Load Workspace..
- Session->Save Workspace As..
- Shortcuts for saving R script and running sections of code
- Environment window shows names of declared variables and their class.
- A collection of specialized programs are kept in packages. Click on the package name in the package tab or include in your script “library(“package name”)”

Input Data

- Organize data with rows corresponding to individual observations and columns to a type of data.
- First row should be column headers.
- Example:

```
selection  replicate      age    fecundity
aco        1              10     21
…..
```

- fec.data<- read.table(“fec.txt”,header=TRUE)
- Help on this or other R functions, type ?read.table in the console

Dataframes

- fec.data[,4] == fec.data$fecundity
- fec.data[20:40,] # rows 20-40 and all columns
- fec.data[,-2] # everything except column 2
- fec20.data<- fec.data[fec.data[,3]<20,] #all females younger than 20 days
- fec20.40.data<- fec.data[(fec.data[,3]<20)|(fec.data[,3]>40),] # all females younger than 20 or older than 40 days. “&” is the and operator, “==“ tests for equality, “!” negation operator -> x!==10 means x not equal to 10.
- Matrices of numbers have the same referencing nomenclature

Lists

- List are vectors of objects of different types
- a<- list(“Tom”,1500,”T”) or tag the components
  a<- list(name=“Tom”,salary=1500,union=”T”)
- a[[2]]=a$salary=a[[“salary”]]=1500
- b<- list(“Mary”,1000,”F”)
- a.b<- list(a,b)
- a.b[[2]][2] -> 1000

Common R functions

- apply(x,dimcode,f,f_args), dimcode=1 (rows), =2 (columns)
- x.means<- apply(x,2,mean)
- Apply to lists, lapply(list(1:10,23:90),median), output is a list.
- To get a vector or matrix output use, sapply(list(1:10,23:90),median).
- But sapply is very flexible. It can be used to do loops, my.output<- sapply(1:100,function(x) {lots of statements})

Writing R functions

-
  ```
  my.function<- function(x) {
  …R statements
  y<-…..
  return(y)
  }
  ```
- Call the function -> my.result<- my.function(7.5)
- Variables defined inside the function are local to the function
- The last line of a function can be, return(y) or just y
- A function can access variables declared outside the function
  ```
  > my.sq<- function(x) return(c*x^2)
  > c<- 2
  > my.sq(4)
  [1] 32
  ```
- To change the value of a variable outside the function use the "<<-" assignment
  ```
  > my.sq<- function(x) {c<<-6
  + return(c*x^2)}
  > c<- 2
  > my.sq(4)
  [1] 96
  > c
  [1] 6
  ```
- Review pages 42-52