Stat 202a, Statistical Computing. Prof. Rick Paik Schoenberg.

1. Modes and lists.
2. Finding the statistical mode in R.
3. Working directories and libraries.
4. Input and output in R.

1. Modes and Lists

Two common problems discussed by Teetor p48 are aList[i] and aList[[i]], and & and &&.

```
aList = list(w = rep(0,4), x = 1:10, y = rep(3,12), z = 1:5)
aList[[2]] ## the second item in the list
aList[2] ## a list containing the second item, and usually not what you want.
x = aList[[2]]
y = aList[2]
x+1
y+1
mode(x)
mode(y)

x = 3; mode(x) ## "numeric".
x = "a"; mode(x) ## "character".
```

2. Finding the statistical mode, in R.

mode(x) gives you the type of variable x is, not a vector's most commonly appearing element. How do find the mode, in the statistical sense?

We can find the mode using table(). table(x) gives a list of the sorted elements in x, along with their counts.

```
  x = c(17,4.3,2.1,4.3,1)
  table(x)
  y = which.max(table(x)) ## Outputs the index in sorted list of x, not the mode.
  y = table(x)
  names(y)
  z = as.numeric(names(y)) ## equivalent to z = sort(unique(x))
  z[which.max(y)]

  mode2 = function(x){ ## finds the mode, but if there's a tie, defaults to the smallest mode.
  y = table(x)
  z = as.numeric(names(y))
  z[which.max(y)]
  }

 x = c(17,17,4,4,2)
 mode2(x)
```

```
  mode3 = function(x){ ## finds the mode(s)
  y = table(x)
  z = as.numeric(names(y))
  y1 = as.numeric(y)
  w = (y1 == max(y1))
  z[w]
  }

## equivalently, we could replace w = (y1 == max(y1)) with
## w = which(y1 == max(y1))

  mode3(x)
  x = c(rep(1,7), rep(10,7), rep(-1,8))
  mode3(x)
  x1 = x[x != -1]
  mode3(x1)
```

3. Working directories and libraries.

Setting the working directory is really important to read files or write to files and know where they go. See Teetor p51.

p54 search() lists all packages currently loaded [not just installed but loaded in current session].

```
  search()
  library(MASS)
  search()
  detach(package:MASS)
  search()
```

Loading in datasets is just like loading packages, see Teetor p53.

```
  head(pressure) ## or just pressure or pressure[1,]
  data() ## lists all the preloaded datasets.
```

instead of data(Cars93, package="MASS") you could just do

```
  library(MASS)
  data(Cars93)
```

p58, library() lists all installed (but not necessarily loaded) packages.
install.packages() is useful to install a new one.
p63, source() is sometimes useful though it can have problems,
especially if your comments go over lines, as in the example with

```
  ## this will be the total of all
the elements in my dataset.
```

4. Input and Output in R.

sink() is useful to output to a file, see Teetor p75. It is especially useful when outputting within a loop.
Similar to cat("x", file = "x.txt")
Note that you have to do sink("x.txt"), then your commands, and then sink() to return to normal output.

read.fwf() p 77 is nice. A similar function is strsplit().
You can load a whole page of numbers, text, etc. using
read.delim() and then choose the column you want using substr().
read.table() is extremely useful, p79.

scan() p87 is great, and once you get the data in R, you can
usually manipulate the data quite easily into a table or
matrix anyway.

For instance,

```
sink("y.txt")
z = runif(102)
cat(z)
sink()
y = scan("y.txt")
x = matrix(y, ncol=3, byrow=T)
```