Lecture 1
Introduction to R

GEOG 489 SPRING 2020

Functions

- Functions: name(parameters)

c() means "concatenate", i.e. it merges the numbers into a single *vector*. The parentheses, in this context, refer to a *function*.

The function name is "c" and the inputs are the things that are inside of the parentheses.

```
x <- c(1,2,4)

x[2:3]
```

Functions

- Functions: name(parameters)

c() means "concatenate", i.e. it merges the numbers into a single *vector*. The parentheses, in this context, refer to a *function*.

The function name is "c" and the inputs are the things that are inside of the parentheses.

```
q <- c(x,x,8)

mean(x)

sd(x)

hist(Nile)
```

Sample dataset: Nile (time series dataset)

Functions

- How to write functions?

1) What are the inputs?

2) How does the function manipulate the inputs?

3) What are the outputs that will be *returned*?

Example: We are going to define a function that does the following:

Input: a vector of values

Output: a count of the number of odd values in the vector

Functions

```
oddcount <- function(x)
{
    k <- 0 # assign 0 to k
    for (n in x) {
        if(n %% 2 == 1)
        {
            k <- k+1 # %% is the modulo operator
        }
    }
  return(k)
}
oddcount(x=c(1,3,5))
```

Functions

oddcount(x=c(1,3,5))

- In the function definition, x is the "formal argument" or "formal parameter" of function oddcount
- c(1,3,5) is the "actual argument" of the function.

R data types

1) Vector: an ordered set of elements that all share the same "mode" (data type).

For instance characters, integers, or floating point numbers.

```
x <- c(5,12,13)

length(x)

mode(x)   #data type
```

R data types

2) Matrix: A matrix is, technically, a vector that has two additional attributes: number or rows and number of columns.

```
mymatrix <- matrix(data=c(1,3,5,8),nrow=2,ncol=2)
#      [,1] [,2]
# [1,]    1    5
# [2,]    3    8
mymatrix2 <- rbind(c(1,5),c(3,8))
```

R data types

3) List: A list is a *vector* in which each element can be any type of data structure, so is the most flexible type of data structure.

We'll define a list as containing a single element numeric vector, a 3-element character vector, and a matrix:

```
mylist <- list(u=2,v=c("abc","def"),w=matrix(data=c(1,2,3,4),
nrow=2,ncol=2))
```

R data types

4) Data frame: A data frame is a list, but with some restrictions, namely, each element of the list must be 1) a vector and 2) the same length of the other elements.

The vectors, however, can be different modes (unlike a matrix). In other words, a data frame is the R equivalent of a spreadsheet.

```
d <- data.frame(kids=c("Jack","Jill"),ages=c(12,10))
     kids     ages
1    Jack      12
2    Jill       10
```

Getting help in R

1) If we know the function…

help(matrix)

?matrix

2) If we do not know what function to use…

e.g. to determine a multivariate normal distribution? We use help.search (and quotes):

help.search("multivariate normal")

??"multivariate normal"

Getting help in R

3) Get help for an entire package

help(package="MASS")

Other sources of help are:

http://www.r-project.org/ -> click "Manuals" on the left

http://www.r-project.org/ -> click "Search" on the left

http://www.rseek.org/ -> probably one of the best searchable help systems