

Getting started in R

1. What is R?

First of all, read 'What is R?' on the R webpage: <http://www.r-project.org> Then, get R!

Install R: <http://cran.stat.sfu.ca/>

Next, install RStudio: <http://rstudio.org/>

RStudio is the program you will run 'R' through. You must have R to run RStudio. Open RStudio and let's take a look..... RStudio provides a nice GUI (graphical user interface) to use R in, which there are four windows:

upper left: where your R scripting files will be located

lower left: the R console (i.e. where the R command line is

upper right: where your R history and workspace are

lower right: where you can open files, view plots, install R packages and get R help

R is the console - this is where the data is kept, where you run analyses, where you create plots. The script is a text file that contains the code for everything you do - save this regularly and annotate with #.

We send code from the script to the console with command + enter (control + enter on Windows).

2. So, what can I do in R?

R is a fantastic open-source program for manipulating data, statistical analysis, and making figures.

At its most simple, R is a calculator.

```
1 + 10  
mean(1,5,5, 8)
```

We can store functions, data, numbers, and words using the assign command.

```
a <- 10  
a  
animal <- "parrotfish"  
animal
```

As a first little example, let's write a simple function and make a plot. First, open a new R script file, and comment it with your name, date, and purpose.

'#' is not interpreted as code - we use # for commenting our scripts

Comment everything! - you will need to revisit your scripts, and comments help to explain what your code actually means

```
#Started April 18th, by JKB
#Purpose: To start learning R

a<-seq(1,10,1)    # create an object 'a' which is a sequence from 1 to 10
b<-seq(10,500,50) # create an object 'b' which is a sequence from 10 to 500,
by 50 each time
plot(a,b, main="My first plot in R") ## plot a against b
```

To get help in R, type '?' plus the function you want help with in the command line. For example:

```
?seq
```

This will open a help file on 'seq' in the lower right hand window of R Studio. This help file tells us that the numbers inside the brackets after 'seq' are the 'arguments' to the function sequence. For the function sequence, the 1st number is the number the sequence starts at, the 2nd number is the number the sequence ends at, and the 3rd number is the amount the sequence increments by. So in our example above a is a sequence that runs from 1 to 10 by 1.

3. Functions in R

But what if you want to run all three lines at once? That's where a function comes in. We usually use functions when we want to run a series of commands in one line - you might do this to simplify your script, or just to save time typing code.

Functions use curly brackets {} to contain commands. In the command line type:

```
my.first.function <- function(){
  a<-seq(1,10,1)
  b<-seq(10,500,50)
  plot(a,b, main="My first plot in R")
}

my.first.function()
```

Functions can get much much more complicated. In fact, everything you do in R is just a function that someone else has written - you don't actually need to look at the function code, just use the help files to

figure out how to use each function.

4. Setting Your Working Directory

Data management (keeping track of code, figures, data) is **essential** for any researcher. If you can get good at this early on you'll save future you a huge amount of time and effort. Read 'BestPractices_DataManagement' for tips on setting your folder structure up correctly.

First off, it is good practice to have a separate working directory for each of your major projects. For these tutorials you might want to create a folder called something like 'LearnR'.

In the command line, type:

```
getwd( )
```

This will tell you what directory (also known as a folder) R has opened in, and this is where your R workspace will be located.

You want your R workspace to be located in the R folder for the project you are currently working on. We can then set the working directory using the 'setwd' function (where setwd stands for 'set working directory'):

```
setwd("Users/James/Dropbox/LearnR")
```

Note that you need to include the full path name

5. Reading in data in R

We want to look at the Isle Royale dataset in R (IsleRoyale_Data.csv). This long-term dataset on wolf-moose population sizes was collected over 5 decades - take a look at how this data is used at <http://www.isleroyalewolf.org/>.

We work with csv file formats in R - you can create these for your own data using excel (File > Save As > .csv). Most online data is provided in csv format.

```
isle_dat<-read.csv(IsleRoyale_Data.csv)
```

To look at a dataset we can just type its name

```
isle_dat
```

But that doesn't work for bigger datasets - instead we usually use head() or tail() to look at the top and bottom of a dataset.

```
head(isle_dat)
tail(isle_dat)
```

6. Now what? There are a number of resources available for learning R. We have posted a few of these resources on the course website. If you run into problems, try to resolve them yourself, then consult your classmates, your TA, or the professor. Feel free to contact us via email or ask questions after lecture/lab.

Here are a few resources on R that I've found useful.

Books:

- The R Book (Crawley)
- Ecological Models and Data in R (Bolker)
- R Cookbook (Teetor)
- Statistics: An Introduction Using R (Crawley)
- Introductory Statistics with R (Dalgaard)

Websites:

- <http://www.ats.ucla.edu/stat/r/>
- <http://www.statmethods.net/> - QuickR website, useful tips
- <http://www.stackoverflow.com/> - every problem you'll ever have is probably answered here
- <https://www.zoology.ubc.ca/~schluter/R/> - UBC Quantitative Methods in Ecology and Evolution course page for R tips

Tutorials:

- <http://cran.r-project.org/doc/manuals/R-intro.html>
- <https://www.codeschool.com/courses/try-r>
- <https://www.coursera.org/> - often has courses that use R

7. Additional things to explore

The best way to learn R is to teach yourself by trying new things and spending time getting lots and lots of error messages (seriously). If you have time, here are some other tasks to explore.

- EcoData Retriever is a site with clean, usable ecological data run by Benjamin Morris and Ethan White. Grab some interesting data and try to make some plots (<http://www.ecodataretriever.org/>)
- Find an R textbook in the Baum lab (but don't take it home!) and try some exercises
- Write a function that creates a new data frame (this might help...
<http://nicercode.github.io/guides/functions/>)