Methods in Macroecology and Macroevolution

Natalie Cooper (natalie.cooper@nhm.ac.uk)

February 2017

Contents

Chapter 1

Methods in Macroecology and Macroevolution

Chapter 2

What you need to be able to do in R before you start

Most people taking this module have used R a lot already, but it is possible you're a bit rusty, or you've found this course on GitHub and have no R experience. This isn't a problem, I will try and summarise what you need to be able to do to get these practicals running below. However, I'm not going to write a help guide to R here, if you can't work out how to open it and get started etc. I strongly recommend the book Getting Started With R or there are lots of great tutorials online.

Throughout, R code will be in shaded boxes:

```
library(ape)
```

R output will be preceded by ## and important comments will be in quote blocks:

Note that many things in R can be done in multiple ways. You should choose the methods you feel most comfortable with, and do not panic if someone is doing the same analyses as you in a different way!

2.1 Installing R (and RStudio)

- Install R from [https://cran.r-project.org]
- You can install RStudio from [http://www.rstudio.com/products/rstudio/download/]. I'd recommend trying this out if you're a beginner as it has a nicer interface.

2.2 Setting the working directory

To use the practicals you need to download all the files for each practical into a folder somewhere on your computer (I usually put mine on the Desktop). We will then tell R to look in this folder for all data etc. by **setting the working directory** to that folder.

To set the working directory you'll need to know what the **path** of the folder is. The path is really easy to find in a Windows machine, just click on the address bar of the folder and the whole path will appear. For example on my Windows machine, the path is:

```
C:/Users/Natalie/Desktop/RAnalyses
```

It's a bit trickier to find the path on a Mac, so use Google if you need help. On my Mac the path is:

```
~/Desktop/RAnalyses
```

Note that the tilde ~ is a shorthand for /Users/Natalie.

We can then set the working directory to your folder using `setwd`:

```
setwd("~/Desktop/RAnalyses")
```

Alternatively if using RStudio you use the menus to do this. Go to Session > Set Working Directory > Choose Directory.

Setting the working directory tells R which folder to look for data in (and which folder you'd like it to write results to). It saves a bit of typing when reading files into R. Now I can read in a file called `mydata.csv` as follows:

```
mydata <- read.csv("mydata.csv")
```

rather than having to specify the folder too:

```
mydata <- read.csv("~/Desktop/RAnalyses/mydata.csv")
```

> Remember if you move the data files, or the folder itself, you'll need to set the working directory again.

2.3 Using a script

Next, open a text editor. R has an inbuilt editor that works pretty well, but NotePad and TextEdit are fine too. However, I **highly** recommend using something that will highlight code for you. My personal favorite is Sublime Text 2, because you can also use it for any other kind of text editing like LaTeX, html etc. RStudio's editor is also very nice.

You should type (or copy and paste) your code into the text editor, edit it until you think it'll work, and then either paste it into R's console window, or you can highlight the bit of code you want to run and press `ctrl` or `cmd` and `enter` or `R` (different computers seem to do this differently). This will automatically send it to the console.

Saving the script file lets you keep a record of the code you used, which can be a great time saver if you want to use it again, especially as you know this code will work!

You can cut and paste code from my handouts into your script. You don't need to retype everything!

If you want to add comments to the file (i.e., notes to remind yourself what the code is doing), put a hash/pound sign (#) in front of the comment.

```
# Comments are ignored by R but remind you what the code is doing.
# You need a # at the start of each line of a comment.
# Always make plenty of notes to help you remember what you did and why
```

2.4 Installing and loading extra packages in R

To run any specialised analysis in R, you need to download one or more additional packages from the basic R installation. For these problem sets you will need to install the following packages:

- `ape`
- `geiger`

- picante
- caper
- BAMMtools

To install the package ape:

```
install.packages("ape")
```

Pick the closest mirror to you if asked.

You've *installed* the packages but they don't automatically get loaded into your R session. Instead you need to tell R to load them **every time** you start a new R session and want to use functions from these packages. To load the package ape into your current R session:

```
library(ape)
```

You can think of install.packages like installing an app from the App Store on your smart phone - *you only do this once* - and library as being like pushing the app button on your phone - *you do this every time you want to use the app.*

2.5 Loading and viewing your data in R

R can read files in lots of formats, including comma-delimited and tab-delimited files. Excel (and many other applications) can output files in this format (it's an option in the Save As dialog box under the File menu). Mostly I will give you .csv files during these practicals. As an example, here is how you would read in the tab-delimited text file called Primatedata.csv which we are going to use in the PGLS practical. Load these data as follows, assuming you have set your working directory (see step 2 above).

```
primatedata <- read.csv("Primatedata.csv")
```

read.csv reads in comma delimited files.

This is a good point to note that unless you **tell** R you want to do something, it won't do it automatically. So here if you successfully entered the data, R won't give you any indication that it worked. Instead you need to specifically ask R to look at the data.

We can look at the data by typing:

```
str(primatedata)
```

```
## 'data.frame':    77 obs. of  9 variables:
##  $ Order          : Factor w/ 1 level "Primates": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Family         : Factor w/ 15 levels "Aotidae","Atelidae",..: 2 2 2 14 3 3 3 4 4 4 ...
##  $ Binomial       : Factor w/ 77 levels "Alouatta palliata",..: 5 6 7 8 9 10 11 15 16 17 ...
##  $ AdultBodyMass_g: num  6692 7582 8697 958 558 ...
##  $ GestationLen_d : num  138 226 228 164 154 ...
##  $ HomeRange_km2  : num  2.28 0.73 1.36 0.02 0.32 0.02 0.00212 0.51 0.16 0.24 ...
##  $ MaxLongevity_m : num  336 328 454 304 215 ...
##  $ SocialGroupSize: num  14.5 42 20 2.95 6.85 ...
##  $ SocialStatus   : int  2 2 2 2 2 2 2 2 2 2 ...
```

Always look at your data before beginning any analysis to check it read in correctly.

str shows the structure of the data frame (this can be a really useful command when you have a big data file). It also tells you what kind of variables R thinks you have (characters, integers, numeric, factors etc.). Some R functions need the data to be certain kinds of variables so it's useful to check this.

As you can see, the data contains the following variables: Order, Family, Binomial, AdultBodyMass_g, GestationLen_d, HomeRange_km2, MaxLongevity_m, and SocialGroupSize.

```
head(primatedata)
```

```
##      Order      Family           Binomial AdultBodyMass_g GestationLen_d
## 1 Primates    Atelidae   Ateles belzebuth         6692.42         138.20
## 2 Primates    Atelidae   Ateles geoffroyi         7582.40         226.37
## 3 Primates    Atelidae    Ateles paniscus         8697.25         228.18
## 4 Primates Pitheciidae  Callicebus moloch          958.13         164.00
## 5 Primates     Cebidae  Callimico goeldii          558.00         153.99
## 6 Primates     Cebidae Callithrix jacchus          290.21         144.00
##   HomeRange_km2 MaxLongevity_m SocialGroupSize SocialStatus
## 1          2.28          336.0           14.50            2
## 2          0.73          327.6           42.00            2
## 3          1.36          453.6           20.00            2
## 4          0.02          303.6            2.95            2
## 5          0.32          214.8            6.85            2
## 6          0.02          201.6            8.55            2
```

This gives you the first few rows of data along with the column headings.

```
names(primatedata)
```

```
## [1] "Order"           "Family"          "Binomial"        "AdultBodyMass_g"
## [5] "GestationLen_d"  "HomeRange_km2"   "MaxLongevity_m"  "SocialGroupSize"
## [9] "SocialStatus"
```

This gives you the names of the columns.

```
primatedata
```

now do square root of 10

```
sqrt(10)
```

```
## [1] 3.162278
```

This will print out all of the data!

This should be everything you need to know to get the practicals that follow working. Let me know if you have any problems (natalie.cooper@nhm.ac.uk).

Chapter 3

Diversity Indices in R

Chapter 4

Visualising phylogenies in R

Chapter 5

Phylogenetic Generalised Least Squares (PGLS) in R

Chapter 6

Macroevolutionary models in R: Part 1 - continuous traits

Chapter 7

Macroevolutionary models in R: Part 2 - discrete traits

Chapter 8

Geometric Morphometrics in R

Chapter 9

BAMM: Bayesian Analysis of Macroevolutionary Mixtures

Chapter 10

Critical thinking about methods and analyses

Chapter 11

Practice Questions