Shock and Awe by Statistical Software – Why R?

Owen L Petchey, Andrew P Beckerman, Dylan Z Childs

Answering ecological and evolutionary questions requires a diverse set of skills and tools. These include managing, visualising, and analysing data from surveys, experiments, and theoretical models. Managing data efficiently builds a solid foundation for visualisations and analyses, as well as for sharing, collaboration and meta-analysis. Visualising data, by plotting graphs or maps, is essential to explore and communicate (in peer review publications for example). Analysing the patterns in the visualisations provides measures of confidence or uncertainty in the answers to questions. Combining management, visualisation and analysis as efficiently, reliably, accurately, and enjoyably as possible will go a long way towards making for a productive, successful, and rewarding career.

The three of us have taken similar, rather tortuous paths towards a solution that satisfies all of these requirements (figure 1). We each started by using a separate software program to do separate tasks: Excel for data management, Sigmaplot for visualisation, Systat or SAS for statistical testing, and a range of programming languages for simulation modelling. Greater experience with SAS led us to use it also for data management. We then migrated to use R for all our management, visualisation, analysis and reporting tasks. In hindsight, we wish we knew then (at the start of our PhDs, or even before) what we know now. R is the best integrated solution to these issues. In this article we will describe the one perceived disadvantage and many advantages associated with R, and hope this helps you decide whether to take the plunge. If you already use R, feel free to read on with a comfortable smug feeling.

The disadvantage

There are very few pull down menus. For anyone used to software with menus and buttons, learning to use R will mean stepping out of their comfort zone. It will involve moments of frustration, exasperation, and shock; but eventually these are replaced with enlightenment, wonder, and awe. One colleague wisely decided to learn R on an old computer so it could be physically abused without too much cost.

Most of these negative emotions result from one thing – having to type every instruction/command/request. The general absence of pull down menus removes the dominant method that we use for interacting with computers. A consequence is that it can take time and effort to learn or transition to R, just because we have to learn an entirely new way to interact with a computer. In our experience, however, the investment made is paid back many times over.

The advantages

There are very few pull down menus. The vast majority of instructions, from importing, to plotting, to analysing, are carried out by typing commands. These instructions can be typed directly into R, like a calculator. However, its always preferable to keep all the instructions in a separate text file, called a script, from which instructions or chunks of instructions can be 'copy and pasted' (or often sent automatically) to R. Keeping instructions in a separate text file has several advantages.

First, your data and the text file are the only two things you need to save and manage for a project's analysis. Repeating

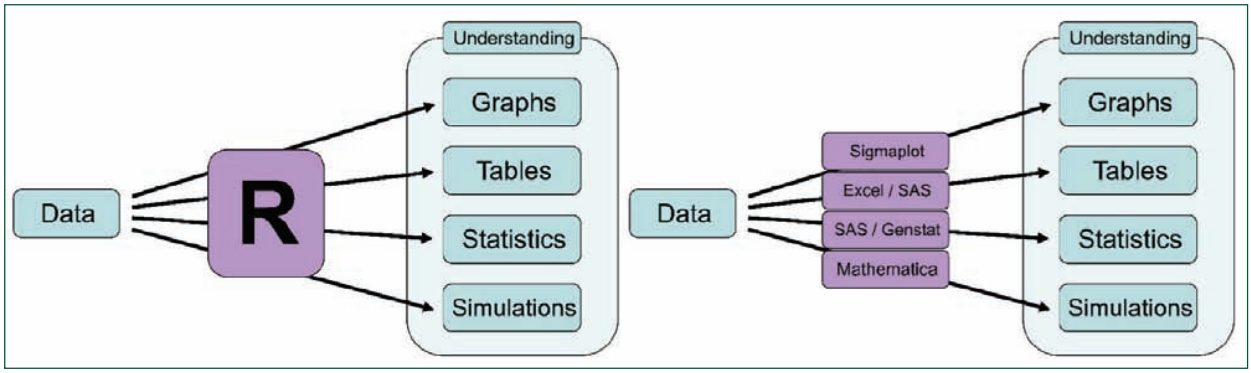


Figure 1 Legend *Our tortuous path to enlightenment. Research and analysis requires a workflow, from experimental design and gathering data through to production of figures, analysis and ultimately a document/report/publication. Initially, we used a variety of tools to achieve the workflow linking data to publications (a – Excel, SAS / Systat / Genstat / GLIM / Mathematica / C). We have found that R can be a single tool facilitating the management of your data, visualisation, and analysis. Importantly, R allows you to produce a single set of instructions for visualisation and analysis that are permanent, annotated, repeatable record of your efforts, and fit for sharing with anyone around the world, on any type of computer.*

an analysis and the associated visualisation is a simple matter of resending the instructions to R. Second, comments and notes can (and should!) be inserted into the script file. For example, you may want to tell the reader (you or colleagues) why the analysis was performed one way, and why the at first glance a sensible method of analysis turned out to be not so sensible. A running commentary through a script can save hours. Third, after three or four months, perhaps at the request of a peer reviewer, changes to graphs or statistics can be made or the analysis run on a revised dataset, without having to repeat the entire analysis step by step using menus. You simply change the instructions for drawing the graph (e.g., add a different legend), and re-run the script – fast, efficient and somewhat revelatory the first time you do it! Finally, collaboration is straightforward when using R, as co-workers can simply share script files, rather than trying to explain and then re-implement an analysis many times over.

A huge amount of freely accessible help and support is available. Within R, every command (function) has a standard help document that is opened using, for example, '?t.test'. One of the most useful sections in these documents is the Examples, which can be copied and pasted into your script file and modified as required. An extensive online community exists. This community, moderated by authors of R, authors of packages and academic and professional statisticians, contains many tutorials, a constantly updated FAQ, and online searchable archives of common questions and problems. The cross-platform, open source nature of R ensures that there is always a group of R geeks just over your shoulder, usually more than willing to help. There are also subject/task specific groupings of packages ('Task Views' on the R web page), covering major sets of analysis tools, such as Multivariate, Spatial, Environmetrics and Bayesian, to name a few. Finally, the growing importance of R is evidenced by the rapidly expanding suite of books describing its use in settings ranging from morphometric analysis to the analysis of time series using wavelets.

R is open source and freely available. Don't take our word for it... go to http://cran.r-project.org and download it. You, your colleagues, your students, your teachers, your friends, your family, your pets will not have to pay a penny, cent, or dong for R. This freedom to download and distribute R is protected by international law. The most significant consequence is that we can be sure that we will, for the rest of our academic careers, be able to use R. Anyone we teach it to will be able to use it, wherever they go. The continuity this creates is priceless, in our opinion. The direct financial savings for institutions, departments, individuals, funding agencies, governments, and ultimately tax-payers are obvious. Furthermore, the open source nature of R ensures that at least the core functionality of R has experienced and continues to experience intense quality control. You can have trust in it.

Visualisation capabilities are unparalleled. Standard 'base' functions for visualisation in R are more than capable of producing figures suitable for publication, ranging from simple box and whisker plots (see Box 1), to highly annotated scatterplots, boxplots, barplots, histograms and custom figures. Take a look at the R graph gallery (http://addictedtor.free.fr/graphiques/) for some interesting examples. Adding fitted/prediction lines, error bars, text, labels, second axes, or additional data is straightforward. With built in libraries (e.g. lattice, ggplot2), R makes it easy to produce multi-panelled figures, facilitating deeper insight into complex experimental designs and observational studies.

Figures can be copied and pasted into other applications, such as MS Word and Powerpoint, and they can also be saved in many standard formats that are accepted by the leading journals around the world. One of the most useful formats is the PDF, and R also supports PNG, jpeg, bitmaps and postscript file formats. Because R is script based, it is also possible to 'automate' the production of figures, controlling pagination, layout and numbering. There are even facilities for making animations from a sequence of figures.

R does statistics. Perhaps most significantly, you can be sure that the vast majority of statistical methods you will ever need are or will be (when they are invented) available in R. It is unlikely that you will need to use a statistical routine that is not included in R or one of its packages for most ecological analyses. Of course, if you do, you can write it in R's language or in C and Fortran code linked to R! Contrary to a commonly held belief, R is actually a rather intuitive programme. Box 1 shows just how easy it can be, via an example of a t-test.

For the novice statistician, R's base distribution contains all of the standard methods you are likely to require, including t – and chi – square tests, regression, multiple regression, ANOVA and ANCOVA (see Box 1). Standard but more advanced methods, including survival models, mixed effects models, multivariate methods, and time series analysis are extensively catered for. As noted above, one of the most useful outputs of the R community are online "task views", web sites hosting collections of packages for specific subjects. Furthermore, many new methods, at the cutting edge of statistical development are developed first in R. Recent examples include, for example, POMP (inference for partially-

observed Markov processes), coxme (general mixed-effects Cox models), lme4 (linear and non-linear mixed effects models), MCMCglmm (Bayesian approach to mixed effect models including animal models).

R is not just for statistics. It is a programming language and environment. It is broadly comparable to MATLAB and you really can do almost anything you want in R, such as construct an individual based simulation or analyse a matrix projection model. Importantly, you can be sure that the interface between data, visualisation, analysis and output, is a seamless one. R kicks Rse.

Disclaimer: OP, DZC and APB are in the Department of Animal and Plant Sciences at the University of Sheffield. They teach statistics in R there. OP and APB also help people learn R through their R4All Courses (http://www.r4all.group.shef.ac.uk/).

Acknowledgements go to the developers of and contributors to R. R Development Core Team (2009). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Box 1 – A publication ready plot and t-test in three lines of code (instructions).

How easy is R? Here are the three lines of code (instructions) needed to carry out a t-test. Lets imagine we have some data on the diameter of some beech trees (response variable) in two sites, A and B (independent variable). We've collected the data and it is now stored in a comma delimited file, that has informative column names, such as the *.csv format available in Excel.

The first step is to import the data into R. We use a command called "read.csv". We call the data set "beech.diameters" and use the "<-" symbol to assign the imported data to this name. The second step is to plot the data (we always plot our data before analysing it). This lets us see the medians and distributions of the two sets of diameters. We use a command called, surprisingly, "plot". Plot requires that you know the response variable names, the independent variable and in which dataset these variables reside. Finally, we do the analysis. You guessed it – the command for a t-test is "t.test".

Here is what you would type into your script file and send/copy to R.

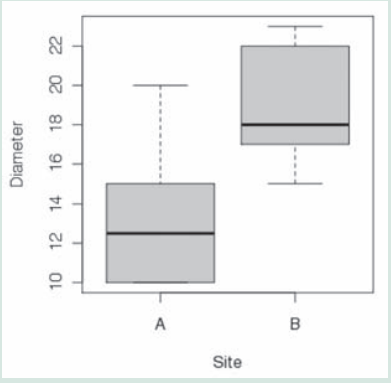
```
beech.diameters <- read.csv("t.test.csv")
plot(Diameter~Site, beech.diameters, col="grey")
t.test(Diameter~Site, beech.diameters)
```

The output from R is very informative. The first line, importing the data, produces no output (no news is good news). "plot" produces an informative box and whiskers plot on continuous response variables versus categorical independent variables.

The "t.test" command produces the name of the test (Welch Two Sample t-test), details about the data (Diameter by Site), the test statistics, degrees of freedom and p-values, the non-null hypothesis, the 95% confidence interval around the difference in means between the two sites, and the Site means.

```
        Welch Two Sample t-test

data: Diameter by Site
t = -4.2226, df = 17.754, p-value = 0.0005259
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.688615 -2.911385
sample estimates:
mean in group A mean in group B
           13.1            18.9
```



Script and data are available at www.r4all.group.shef.ac.uk

Also of interest:

Special interest group for use of R in Ecology (R-sig-ecology)

https://stat.ethz.ch/mailman/listinfo/r-sig-ecology)

The R-sig-ecology special interest group is a discussion list that provides a lively source of information for ecologists using R in their daily data analysis. The more than 1000 registered users of the list range from the novice to authors of code packages. A variety of detailed questions are answered daily. Unlike other R lists, the users are predominantly ecologists, so that members of the list are better able to understand the particular quirks and issues inherent in exploring ecological data. Overall, this list and its archives are a wonderful resource for ecologists interested in using R for data analysis.

Celebrating Ecology: Runner up in the Ecology in Action section was Hannah Grist for her photo of researchers conducting a tern colony census at Monomoy, Massachusetts. Aka, The Birds...