Use R for data analysis and research

David Lillis[*]
New Zealand Qualifications Authority, PO Box 160, Wellington 6140

Introduction

The R statistics language and environment[1] for scientific and statistical computing and graphics is a genuine New Zealand success story. While at the University of Auckland during the early 1990s, Ross Ihaka and Robert Gentleman developed the initial version of R, primarily for use as a teaching tool. R is open source software which you can download from the CRAN (Comprehensive R Network) website (http://cran.r-project.org/), and combines a powerful programming language, a comprehensive range of statistical functions, and outstanding graphics.

R implements a dialect of the S language that was developed around 1975 at AT&T Bell Laboratories. Since mid-1997, further development of R has been overseen by a core team (currently 19 people), drawn from many different institutions worldwide. If you need a statistics environment that includes a programming language, R could be just what you are looking for. It's true that the learning curve is longer than for spreadsheet-based packages but, once you master the R programming syntax, you can develop your own very powerful analytic tools. Many software packages are available on the Internet for use with R, and very often the analytic tools you need can be downloaded at no cost.

R uses command line input that those with limited previous exposure to programming may find intimidating. However, the reward for mastery of R is access to a large and rapidly growing range of abilities that extends widely over many different areas of scientific, statistical, business and other applications. The CRAN task views web pages[2] give some indication of the range of abilities currently available.

The main issue for those new to R is the time required to master the syntax, but several nice Graphical User Interfaces (GUIs), such as John Fox's R Commander package[3,4], are available that make it much easier for the newcomer to develop proficiency in R than it used to be. In recent years several well known commercial packages, such as SPSS and SAS, have become very popular, both in New Zealand and around the world. However, for many statisticians and researchers, R is the environment of choice because of its powerful programming language; the wide range of abilities that are available from its contributed packages; the integration of those abilities with powerful and wide-ranging graphics; data input abilities (including the ability to input and process a wide range of specialist data formats such as Excel files and text files, various spatial data formats, Network Common Data Form formats for climate and mapping applications); and powerful data manipulation and analytic abilities.

For my own work as a statistician and data analyst, I have found the abilities discussed in this paper particularly relevant, though others will have different requirements. You may be looking for a tool for your own data analysis. If so, let's take a brief look at what R can do for you. First, let's look at some R syntax.

Some basic R syntax

R uses object-oriented programming so that your data can be created within R or else read in from .csv or other files as objects. For example, from the R command line or from within an R program, you can read in the data contained within a .csv file called mydata.csv, as follows:

```
A <- read.csv(mydata.csv, header =
              TRUE)
```

. . . where the argument **`header = True`** informs R that the data file has a header (i.e. a set of variable names) that occupies the first row. The object A is an array that contains all of the data of the original file. You can now examine the array using a diverse set of commands. For example, the functions **`mean(A)`** and **`sd(A)`** find the mean and standard deviation of each variable (or column), while the command **`summary(A)`** provides much diagnostic information on A. In addition, you can subset your data quite easily. The syntax **`A[3,7]`** picks out the element in row 3 and column 7, while **`A[14, ]`** selects the fourteenth row and **`A[,6]`** selects the sixth column, and so forth.

Now that you have read in your data, the syntax **`B <- 3*A + 7`** triples each element of A, adds 7 to each tripled element and stores the resulting array as the object B. Now you can save

[*] *Correspondence:david.lillis@nzqa.govt.nz*

David Lillis is a senior statistician with the New Zealand Qualifications Authority (NZQA). He holds a PhD from Curtin University in Western Australia. At NZQA he conducts a wide range of data analysis, including the analysis of NCEA and New Zealand Scholarship results. In particular, he writes software in the R language for Item Response Theory as one approach to ensuring the high quality of secondary examinations.

Dr Lillis is a past president of the New Zealand Association of Scientists.

this array as a .csv file called Outputfile.csv as follows:

```
write.csv(B, file=”Outputfile.csv”)
```

The **sort, rank** and **order** commands allow you to sort the columns of A as necessary (e.g. according to the elements in one particular column), and R provides various means of sub-setting arrays (e.g. the **subset** command) by selecting rows and columns according to your own particular criteria. R’s several control structures for creating loops have the reputation for being rather slow, but actually are pretty good[5].

R provides a comprehensive range of basic statistical functions relating to the commonly-used distributions (normal distribution, t distribution, Poisson, etc.), and many other distributions. It also provides a range of non-parametric tests that are appropriate when your data are not distributed normally (or according to any particular distribution). Linear, non-linear and logistic regressions are easy to perform, and finding the optimum model (i.e. by eliminating non-significant predictors and factor interactions) is particularly easy. Implementing General Linear Models, such as Analysis of Variance (ANOVA), Multivariate Analysis of Variance, and Analysis of Covariance, is also straightforward and, once you know the syntax, you may find that such tasks can be done just as quickly in R as in other packages. For example, the syntax:

```
model <- lm (y ~ x)
```

performs an ordinary least squares regression on two vectors of data, x (the independent variable) and y (the dependent variable), and stores the model as an object called ‘model’. The intercept and slope are stored as the objects **model$coefficient[1]** and **model$coefficient[2]**, so that you can create a predictive model as follows:

```
prediction <- model$coefficient[1] +
model$coefficient[2]*x
```

Non-linear regressions always pose a greater challenge than linear regressions. However, for many applications the nls function makes life relatively easy. Take a look at the rather noisy nonlinear data in the following plot (Figure 1). You will also see a smooth, two-parameter logistic curve:

$$Y = \{ 1 + exp \, [\, a \, (b - \theta) \,] \, \}^{-1}$$

that was fitted to this data using just a few lines of code through weighted least-squares estimation of the parameters a and b.

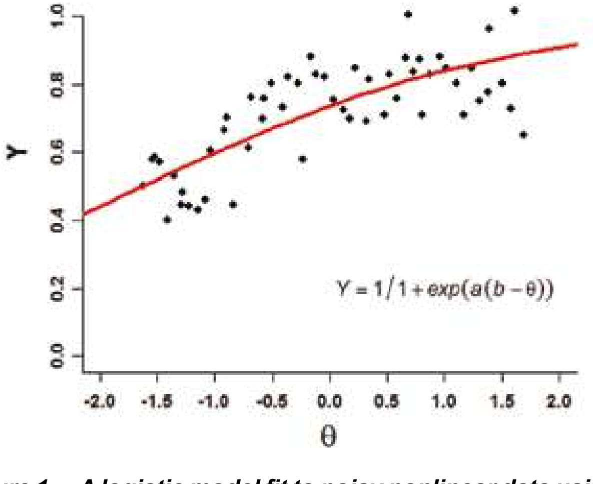

Figure 1. A logistic model fit to noisy nonlinear data using R’s nls function.

The nls function provides just one approach to fitting nonlinear models, and of course R provides the standard diagnostics that indicate the quality of the fit.

More on general linear models

R provides several powerful functions (e.g. the aov and manova functions) that allow you to perform standard inferential statistical modeling, including one- and two-way ANOVA, Multivariable Analysis of Variance, and Analysis of Covariance. The usual post-hoc tests for identifying factor levels that are significantly different from the other levels (e.g. Tukey and Sheffe tests) are available, and testing for interactions between factors is easy. P. M. E Altham, of the Centre for Mathematical Sciences at Cambridge University, has written a helpful paper on the above topics, entitled Introduction to Statistical Modeling in R[6].

Factor Analysis, and the related Principal Components Analysis, are well known dimension reduction techniques that enable you to explain your data in terms of smaller sets (linear combinations) of independent variables (or ‘factors’). Both methods are available in R, in addition to the usual graphical procedures (e.g. scree plots of principal component eigenvalues and plots of the first two principal components). Code for complex designs, including one- and two-way repeated measures, and four-way ANOVA (e.g. two repeated measures and two between-subjects), can be written relatively easily or downloaded from various websites (e.g. William Revelle’s Personality Project[7]). Other analytic methods include Cluster Analysis, Discriminant Analysis, and Multidimensional Scaling. Esoteric techniques such as Correspondence Analysis[8] can be utilised by downloading the necessary code from the Internet. Of course, R provides the standard methods for smoothing noisy data (e.g. locally weighted scatterplot smoothing and spline-based methods).

Helpful R websites

You can find some very useful tutorials and downloadable packages of R code for fields as diverse as biometry, epidemiology, spatial modeling, survey design and analysis, biostatistics and medical research, astrophysics, econometrics, financial and actuarial modeling, the social sciences, psychology and psychometrics. One of the best websites, providing comprehensive information on statistical analysis using R, is Rob Kabakoff’s Quick R website[9]. If you are interested in astrophysics, the Penn State Astrophyics School offers a nice website that includes both astrophysics-based examples and code[10]. Another outstanding website is that of Teresa Scott (a biostatistician at Vanderbilt University in Tennessee), which provides tutorials and executable code for use in biostatistics[11]. A short paper by Marco Martinez, R for Biologists[12], covers many of the basic statistical techniques commonly needed by postgraduate students in the biological sciences, and definitely is worth reading. Software packages are now available for spatial analysis using R[13]. If ecology is of interest, I can recommend a site by Ben Bolker called Ecological Models and Data in R[14], and excellent material on epidemiology can be found quite easily on the Internet. Finally, to help you get started on drawing graphs in R, you could try Frank McCown’s site - Producing Simple Graphs in R[15] - and a site by Mark Gardener called Using R for Statistical Analysis – An Introduction[16]. Following the references to this paper I give the URLs for several other helpful websites and tutorials. In the sections that follow, I also mention some specialist applications and then give examples of R's graphics capability (Figures 2 and 3), and how R was used in developing a model of the performances of international athletes (Figures 4–9).

Specialist applications

Here I will mention just a few of the more common applications.

Time Series Analysis

An excellent description of how to conduct time series analysis in R is that of the University of Pittsburgh, Time Series Analysis and its Applications: with R Examples[17].

Multi-Level Modeling

My own experience is that conducting multi-level modeling in R is considerably more difficult than it is using commercial packages such as MLWIN. Even so, Paul Bliese has produced a monograph that explains how to perform multi-level modeling using R[18].

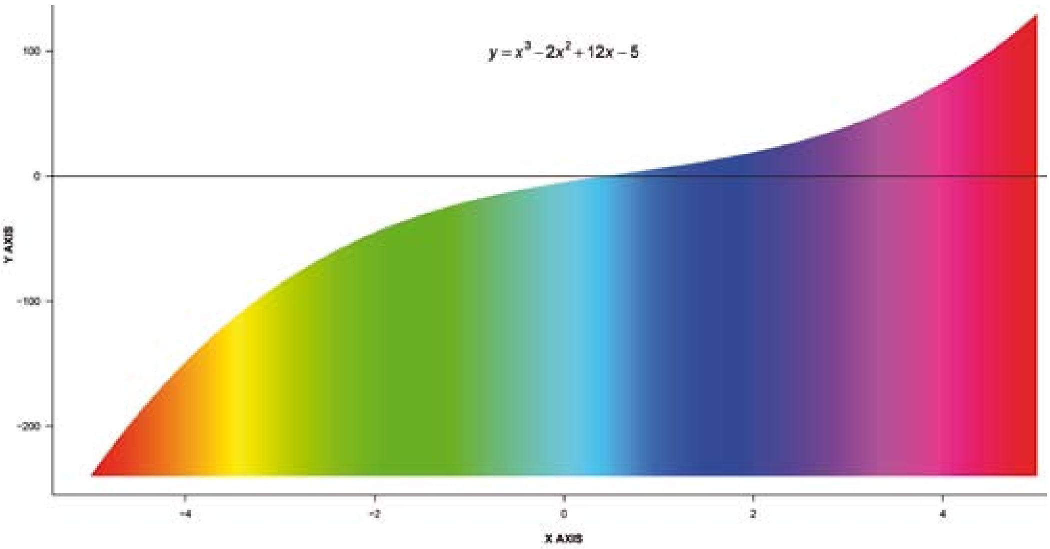

Figure 2. A nice example of R's graphics capability.

Monte Carlo Methods

A number of sources give excellent accounts of how to perform Monte Carlo simulations in R (i.e. drawing samples from multidimensional distributions and estimating expected values). A valuable text is Christian Robert's book, Introducing Monte Carlo Methods with R[19], and Murali Haran gives an interesting astrophysical example in the CAStR website[20].

Structural Equation Modeling

Structural Equation Modeling (SEM) is becoming increasingly popular in the social sciences and economics as a complement to other modeling techniques such as multiple regression, factor analysis, and analysis of covariance. Essentially, SEM is a kind of multiple regression that takes account of factor interactions, nonlinearities and measurement error. Useful references for conducting SEM in R include discussions by Revelle[7], Farnsworth[21] and Fox[22, 23].

Data Mining

R is not quite as effective as other commercial packages in dealing with very large datasets because usually the entire dataset must be read into memory. Even so, I have successfully performed analysis on datasets of around 250,000 rows and 200 columns using a 64 bit, 4 Gigabyte machine. A number of very useful resources are available for anyone undertaking data mining in R. For example, Luis Torgo has just published a book called Data Mining with R – learning with case studies[24], and presents a set of four case studies with accompanying datasets and code which the interested student can work through. Torgo's book provides the usual analytic and graphical techniques used every day by data miners, including specialized visualization techniques, dealing with missing values, developing prediction models, and methods for evaluating the performance of your models.

Also of interest to the data miner is the Rattle[25] (R Analytical Tool to Learn Easily) GUI. Rattle is a data mining facility for analysing very large data sets. It provides many useful statistical and graphical data summaries, presents mechanisms for developing a variety of models, and summarises the performance of your models.

Econometrics and Actuarial Science

For econometrics try Grant Farnsworth's Econometrics in R[21] and, for assistance with R for actuarial science, try An Introduction to R: examples for Actuaries, by De Silva[26].

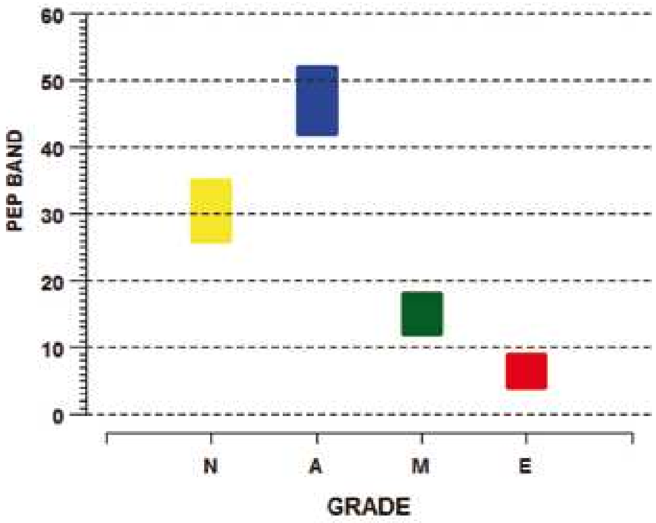

Figure 3. Estimated grade distributions for the Level 3 Physics Standard 90522.

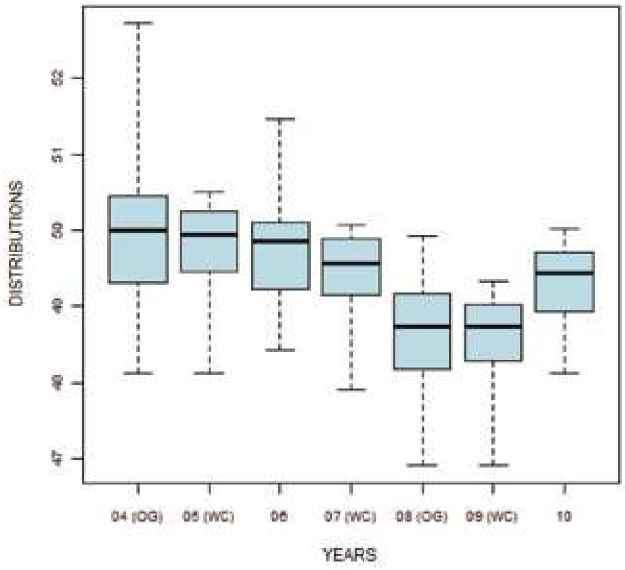

Figure 4. A boxplot of recorded times for the 100 m swim between 2004 and 2010.

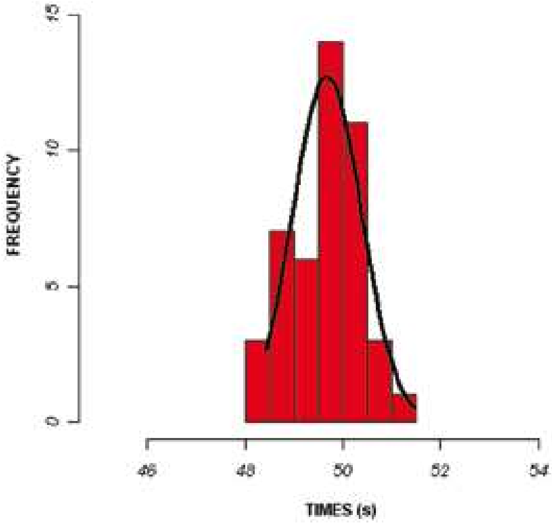

Figure 5. A histogram of recorded times for the 100 m swim for 1996.

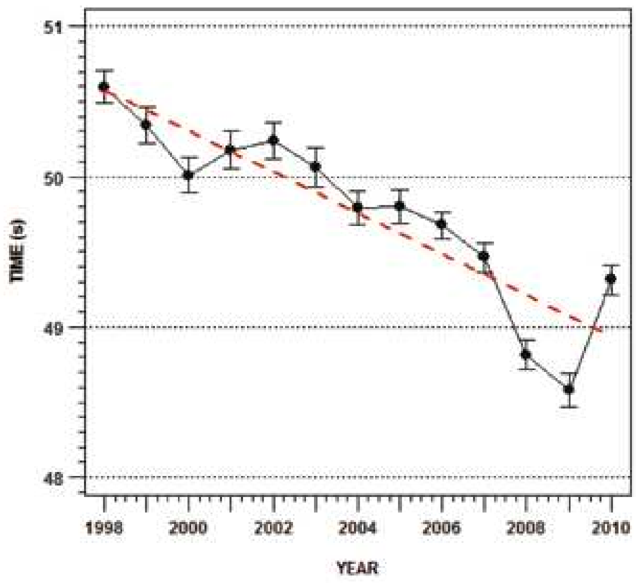

Figure 6. Mean recorded times for the 100 m swim with 95% confidence intervals, 1998–2010.

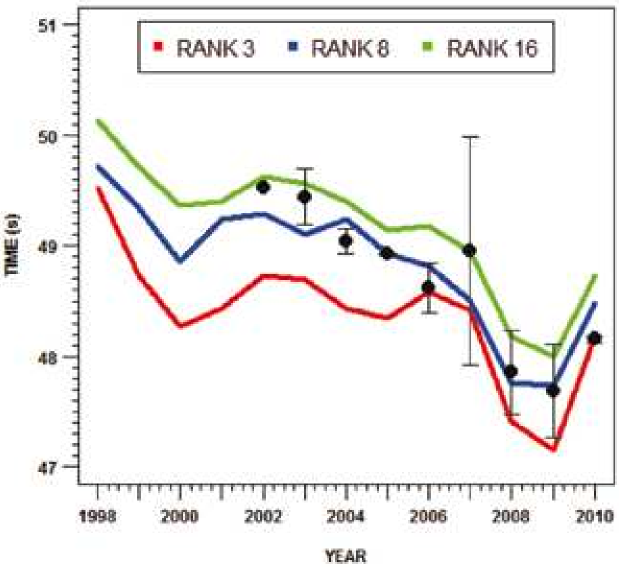

Figure 7. A time series of the mean annual performances of those swimmers who became medalists in 2010.

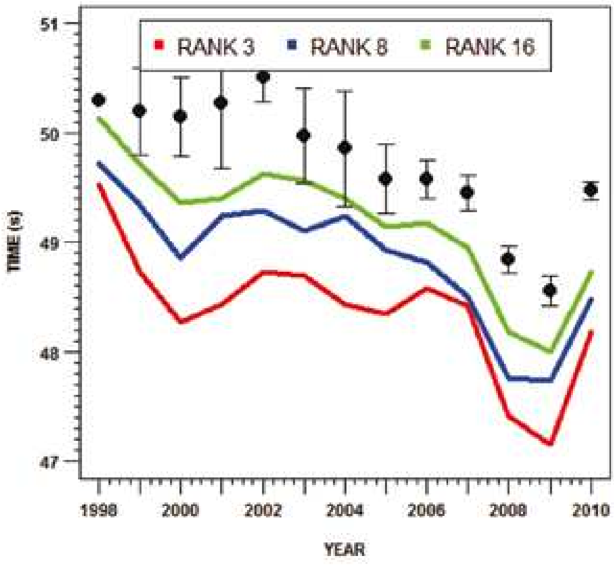

Figure 8. Mean annual performances of those swimmers ranked outside the top 16 in 2010.

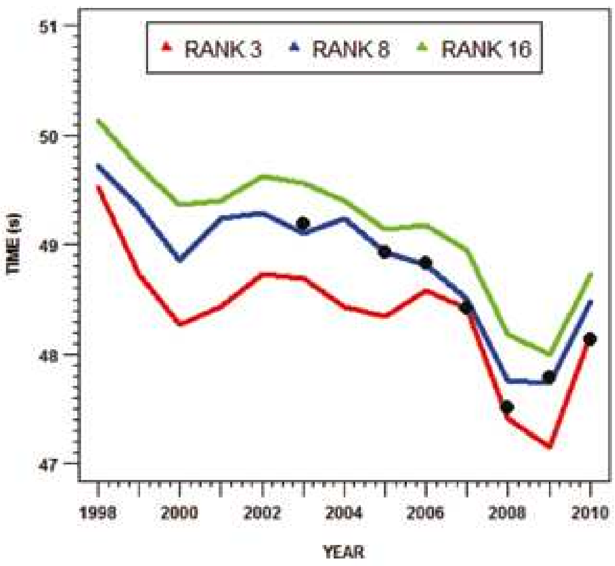

Figure 9. A time series of the mean annual performances of Michael Phelps.

Graphics

Quite simply, the quality and range of graphics available through R is superb and, in my view, superior to those of any other system I have encountered. Of course, you have to write the necessary code but, once you have mastered this skill, all things are possible! You can write your own code from scratch, but many websites provide helpful examples, complete with code, which you can download and modify to suit your own needs. R provides a comprehensive range of colours and hues (*see*: http://research.stowers-institute.org/efg/R/Color/Chart/ColorChart.pdf). Figure 2 gives an example of what you can achieve with R.

Of course, you may never need to produce a plot quite like this one, but you can see the wonderful graphics capability of R.

Use R for repetitive analyses

One of the main advantages of R is that you can create programs that deal efficiently with situations in which a particular analysis is to be implemented on many similar datasets, requiring heavy computation and the preparation of plots, charts and tables. R is in a class by itself for such applications.

For example, every year the New Zealand Qualifications Authority conducts various analyses on the considerable volumes of data that emerge from the NCEA and New Zealand Scholarship examinations and internal assessments. These analyses include direct processing of national results distributions for hundreds of standards, analyses of the performance of the assessments themselves using Item Response Theory (IRT), conventional analytic approaches that include Principal Components Analysis for exploring the dimensionality of the assessments, and various reliability and correlation procedures. Involving many hundreds of process steps, our IRT analyses alone would be prohibitively time-consuming if undertaken using spreadsheet-based packages. To analyse the quality of the assessment items we use the graded response model of Samejima[27] and a two-parameter logistic model, viz:

$$P_k = \{ 1 + exp [a (b_k - \theta)] \}^{-1}$$

. . .where k indexes the assessment grades Achieved (A), Merit (M) and Excellence (E), θ is the calculated ability (which you can also think of as a measure of performance) and is measured in standard deviations from the mean ability of the candidature, P_k is the probability of achieving a particular grade for a candidate of ability θ, a is the fitted item discrimination (a measure of the extent to which the item discriminates between candidates of different abilities), and b_k is the estimated difficulty of gaining either an A, M or E grade for the item. Our approach uses a maximum likelihood method and an in-house search algorithm to estimate θ for each candidate on the basis of his or her performance across all items of the assessment, taking account of the discrimination and the three difficulty parameters of each item. Our software comprises about 3000 lines of R code, allowing us to perform IRT analysis on hundreds of standards and create for each a range of complex diagnostic plots and tables in just a couple of weeks.

As another example, every year, prior to the annual external examinations, we use statistical models of the abilities of national candidatures in each subject to estimate the likely national grade distributions (percentages of all results at Not Achieved (N), Achieved, Merit or Excellence level) for all externally assessed standards, and we publish plots of these distributions on NZQA's public website. These estimated distributions are called Profiles of Expected Performance (PEPs). Figure 3 gives the plot for standard 90522 (a Level 3 Physics standard that covers elementary atomic and nuclear physics), calculated prior to the 2010 examination round.

Creating some 300 plots like this one would be a time-consuming business using Excel, SPSS or some other spread-sheet-based system, but our in-house R code creates all of them as pdf files in less than 4 seconds, ready for publication on the Internet. The program reads in a csv file of predicted national grade distributions (one row for each standard) and calls a function that creates and labels each axis, creates the title, draws the horizontal dotted lines that you can see in the plot, and finally draws the colored rectangles so that the upper and lower bounds of each rectangle match the upper and lower bounds of the distribution.

Develop your own models: an example

Writing R code can be demanding for those with limited experience. However, the final product can be immensely powerful. As an example, I have written R software to analyse the performance of domestic and international athletes as a tool for predicting the likely future success of individual athletes and particular groups of athletes. The software developed so far involves about 1500 lines of R code that produces various diagnostic plots and tables. In Figure 4 we see a boxplot of times recorded by some 450 of the world's top 100 m swimmers between 2004 and 2010. In 2004 (Athens) and 2008 (Beijing) we had the Olympic Games (denoted by OG), while in 2005 (Montreal) and 2009 (several venues) we had the World Aquatics Championships (denoted by WC).

About 10 lines of code were required to produce the boxplot, which suggests consistent improvements in performance between 2004 and 2009. R provides a wide range of options for drawing boxplots like this one, including notched boxplots (which provide evidence of equality or otherwise of medians) and violin plots[28]. Figure 5 gives a histogram of times with a fitted normal curve for all recorded performances from the same group of international swimmers in 2006.

A function involving about 20 lines of code produces a histogram with a fitted normal curve automatically for each year of the dataset. Drawing a histogram for every year involves a loop over each year and processing the appropriate records before moving to successive years.

So - what else can we do? Figure 6 gives a time series of mean performances for Olympic Games and World Championship 100 m finalists between 1998 and 2010. The error bars denote 95% confidence intervals (or about 1.96 times the standard error of the mean).

The extent of the improvement in overall performance between 1998 and 2009 is quite striking, and a partial explanation for the abrupt reduction in performance evident in 2010 was FINA's (swimming's international governing body) decision to ban certain high-technology swimsuits at pinnacle events such as Olympic Games and World Championships. It takes about 30 lines of code to perform the necessary calculations (including a function developed in-house to plot the 95% confidence intervals) and create the plot automatically as just one part of the analysis.

Figure 7 gives a time series of the mean annual performances of those swimmers who eventually became 100 m medalists in 2010. The solid red line denotes the third-ranked performance in each year. The blue line denotes the eighth-ranked performance, while the green line denotes the sixteenth-ranked performance in each year. We refer to these lines and the spaces between them as 'rank-based zones'.

My time series of recorded performances for the 2010 medalists begins in 2002, though at present I have only one record for some years and hence no confidence intervals for those years. The wide confidence intervals in some years arise because few records pertaining to the medalists are available for those years. Naturally, the medalists performed outstandingly in 2010, but my models show that they also performed more strongly than both finalists who did not win medals and non-finalists prior to that year. Equivalent plots for finalists and non-finalists assist in creating predictive models for identifying possible future medalists. Figure 8 shows the equivalent plot for those swimmers who were ranked outside the top 16 in 2010.

It is clear that few, if any, of those swimmers who were ranked outside the top 16 in 2010 were ever likely to emerge as medalists. The software also produces a plot for each individual athlete. For example, Figure 9 gives the plot for the US swimmer, Michael Phelps.

Phelps has won a total of 16 Olympic medals, including six gold medals at Athens in 2004 and eight at Beijing in 2008. At present, my dataset holds only one record for Phelps for each year, so that there are no confidence intervals on this plot. We see that Phelps' performances improved consistently between 2003 and 2008, but fell off slightly afterwards, perhaps partly due to FINA's ban on high-technology swimsuits.

The software produces a similar plot for every swimmer, each named and labeled automatically. In addition to the plots shown in this paper, the software produces a range of diagnostic tables and other outputs that could prove useful for managing the training regimes of competition athletes. In all, the software creates some 500 plots and tables in about 10 seconds. It deals very effectively with missing data because R provides functions that are designed specifically for this purpose, and it can be adapted to suit the different datasets that emerge from many kinds of sport. Extensions to this work could include the development of mechanisms for characterising the performances of different groups of athletes (for example, do medalists and finalists display lower variability in performances than others and do they show greater levels of improvement prior to pinnacle events than others?), analytic methods for predicting the future performances of athletes, and the likely benchmarks for success in top level competitions such as the Olympic Games. R can handle it!

Books, tutorials and other resources for the beginner

Currently, a huge range of books on R, or that demonstrate the use of R in various contexts, is available[29].

Many of them are excellent, but I found two to be particularly helpful when I was new to R. The first is Statistics: An Introduction using R[30], by M.J. Crawley, of the Department of Biological Sciences at Imperial College. This outstanding book is supported by a website that provides much valuable additional material, exercises and executable code. Another very helpful text is that by Professor John Maindonald of the Australian National University – Data Analysis and Graphics using R[31].

Many short descriptions and tutorials on R are downloadable from the Internet. I have also prepared a short tutorial, designed to get around the kinds of problems that beset those new to R. Please contact me by e-mail if you wish to receive this tutorial and accompanying datasets. Also very helpful for beginners is a wide range of blogs and e-mail lists (such as R-downunder[32]) in which experts can provide assistance with programming problems.

Conclusion

For many scientists and data analysts, mastery of R could be an investment for the future, particularly for those who are beginning their careers. The technology for handling scientific computation is advancing very quickly, and is a major impetus for scientific advance. Some level of mastery of R (or an equivalent such as Matlab or Python) has become, for many applications, essential for taking advantage of these developments. Spatial analysis, where R provides an integrated framework access to abilities that are spread across many different computer programs, is a good example.

A few years ago I would not have recommended R as a statistics environment for generalist data analysts or postgraduate students, except those working directly in areas involving statistical modeling. However, the appearance of GUIs, such as R Commander and the new iNZight GUI[33] (designed by Chris Wild and Dineika Chandrananda at Auckland University for use in schools), makes it easier for non-specialists to learn and use R effectively. I am most happy to provide advice to anyone contemplating learning to use this outstanding statistical and research tool.

References

1. R Foundation for Statistical Computing *R: A language and environment for statistical computing.* R Foundation for Statistical Computing,Vienna. ISBN 3-900051-07-0. http://www.R-project.org.
2. CRAN Task Views: http://cran.ms.unimelb.edu.au/web/views/
3. Fox, John. *Getting Started with the R Commander.* http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/Getting-Started-with-the-Rcmdr.pdf
4. Fox, John. 2005. The R Commander: A Basic-Statistics Graphical User Interface to R. *Journal of Statistical Software 14(9)*. http://www.jstatsoft.org/
5. See: http://manuals.bioinformatics.ucr.edu/home/programming-in-r
6. Altham, P.M.E. *Introduction to Statistical Modeling in R.* www.statslab.cam.ac.uk/~pat/redwsheets.pdf
7. Revelle, William. *Using R for Psychological Research.* http://www.personality-project.org/r/
8. Greenacre, M.J. 1984. *Theory and Applications of Correspondence Analysis*. Academic Press, London.
9. Kabakoff, Robert. *Quick-R.* http://www.statmethods.net/index.html
10. Penn State Astrostatistics School. http://www.iiap.res.in/astrostat/RTutorials.html

11. Scott, Theresa. http://biostat.mc.vanderbilt.edu/wiki/Main/TheresaScott

12. Martinez, Marco. *R for Biologists*. http://cran.r-project.org/doc/contrib/Martinez-RforBiologistv1.1.pdf

13. Bivand, R.S.; Pebesma, J.; Gomez-Rubio, V. *Applied Spatial Data Analysis with R*. UseR! Series, Springer, ISBN: 978-0-387-78170-9 http://gisandscience.com/2010/03/01/applied-spatial-data-analysis-with-r/

14. Bolker, Ben. *Ecological Models and Data in R*. http://www.math.mcmaster.ca/~bolker/emdbook/

15. McCown, Frank. *Producing Simple Graphs with R*. http://www.harding.edu/fmccown/R/#barcharts

16. Gardener, Mark. *Using R for Statistical Analysis – An Introduction*. http://www.gardenersown.co.uk/Education/Lectures/R/index.htm

17. University of Pittsburgh. *Time Series Analysis and its Applications: with R Examples*. http://www.stat.pitt.edu/stoffer/tsa2/index.html

18. Bliese, Paul. *Multilevel Modeling in R*. http://cran.r-project.org/doc/contrib/Bliese_Multilevel.pdf

19. Robert, Christian P. 2010. *Introducing Monte Carlo Methods with R*. Series: Use R, Casella, George. 284 p. ISBN: 978-1-4419-1575-7.

20. Murali, Haran CAStR website. http://www.stat.psu.edu/~mharan/MCMCtut/MCMC.html

21. Farnsworth, Grant V. 2008. *Econometrics in R*. http://cran.r-project.org/doc/contrib/Farnsworth-EconometricsInR.pdf

22. Fox, John. 2006. Structural Equation Modeling With the sem Package in R. *Structural Equation Modeling 13(3)*: 465–486. Lawrence Erlbaum Associates, Inc.

23. Fox, John. *Structural Equation Models, Appendix to. An R and S-PLUS Companion to Applied Regression*. http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-sems.pdf

24. Torgo, Luis. *Data Mining with R*. http://www.liaad.up.pt/~ltorgo/DataMiningWithR/

25. Rattle GUI http://rattle.togaware.com/

26. De Silva, N. 2006. *An Introduction to R: examples for Actuaries*. http://toolkit.pbworks.com/f/R%20Examples%20for%20Actuaries%20v0.1-1.pdf

27. Samejima, Fumika. 1969. Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement 34*: 100–114.

28. http://www.statmethods.net/graphs/boxplot.html

29. http://www.R-project.org/doc/bib/R-books.html.

30. Crawley, M.J. 2005. *Statistics: An Introduction using R*. John Wiley & Sons Ltd. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470022973,subjectCd-ST05.html ISBN 0-470-02297-3. http://www3.imperial.ac.uk/naturalsciences/research/statisticsusingr

31. Maindonald, J.H. 2010. *Data Analysis and Graphics Using R: an Example-Based Approach*. 3rd edn. Cambridge University Press. ISBN 798-0-521-762-93-9. http://maths.anu.edu.au/~johnm/r-book/daagur3.html

32. R-downunder. http://www.stat.auckland.ac.nz/mailman/listinfo/r-downunder

33. http://www.stat.auckland.ac.nz/~wild/iNZight/

Other recommended material available on the web

Aragon, Tamas; Enanoria, W.T. *Applied Epidemiology using R*. http://www.medepi.com/

Harte, David. *An Introduction to the R Language*. Statistics Research Associates Ltd. www.statsresearch.co.nz

Lumley, Thomas. *Survey Analysis in R*. http://faculty.washington.edu/tlumley/survey/

Muenchen, Bob. *R for SAS and SPSS Users*.
http://RforSASandSPSSusers.com

Nenadi, C.; Zucchini, Walter. *Statistical Analysis with R – a quick start.* http://www.statoek.wiso.uni-goettingen.de/mitarbeiter/ogi/pub/r_workshop.pdf

Paradis, Emannuel. *R for Beginners*. Institut des Sciences de l'Evolution, Universite Montpellier II, F-34095 Montpellier cedex 05, France. E-mail: paradis@isem.univ-montp2.fr

Verzani, John. *SimpleR - Using R for Introductory Statistics*. http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf