Interacting with Data, Concepts and Models: Illustrations from the `rpanel` Package for `R`

Adrian W. Bowman

Abstract The rich statistical computing environments which are available for data analysis also provide very flexible tools for practical work in the teaching and learning of statistics. R is an outstanding example of this, with an enormous variety of data exploration and modelling tools which are easily accessible. However, there is also a need to support students in the understanding and intuition of key statistical concepts and models. The tools available within R to create interactive graphical controls can be used for this to very good effect. This is explored through the rpanel package, which aims to make the creation of this type of control as straightforward as possible. Examples of interacting with data (in windowing time series), concepts (random variation and correlation) and models (analysis of covariance and logistic regression) are described. The mechanism for coding this type of interactive display is also outlined.

Keywords Animation • Graphics • Interaction • R

1 Introduction

Computing technology has had a major influence on the teaching and learning of virtually all subjects within higher education and not least in Statistics. The ability to explore data graphically and fit a wide variety of statistical models has made real practical applications feasible to tackle in a lecture or laboratory setting and, where appropriate, has allowed the focus to be placed on issues of modelling and interpretation rather than on more technical aspects. In particular, the arrival of the open-source

A.W. Bowman (✉)
School of Mathematics and Statistics, The University of Glasgow,
Glasgow G12 8QQ, UK
e-mail: adrian.bowman@glasgow.ac.uk

H. MacGillivray et al. (eds.), *Topics from Australian Conferences on Teaching Statistics: OZCOTS 2008-2012*, Springer Proceedings in Mathematics & Statistics 81,
DOI 10.1007/978-1-4939-0603-1_1, © Springer Science+Business Media New York 2014

system R (R Development Core Team 2006) has provided data and modelling tools, from the elementary to the state of the art, which are easily accessible. In designing courses, the availability of suitable statistical computing tools is no longer a barrier.

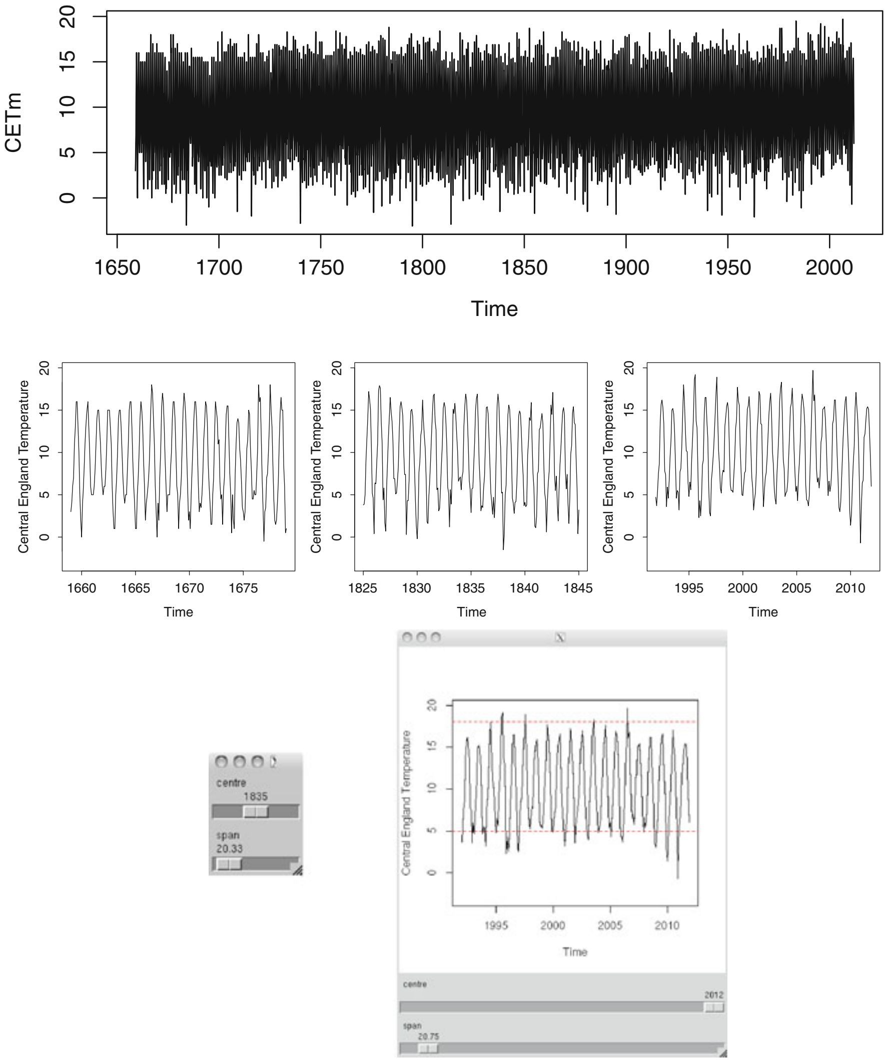
However, while the process of statistical modelling is very well supported computationally, this is much less true of the process of understanding the underlying concepts and methods. There have been many projects which have produced illustrative graphical software directed at the teaching and learning of concepts, but these have often been stand-alone tools written in languages which allow flexible, and often interactive, graphical tools to be created. The *STEPS* (Bowman and McColl 1999), *ActivStats* (Velleman 2004) and *CAST* (Stirling 2012) projects are all examples of this type of material, including multimedia resources, which all remain available.

Many users of R remain unaware that systems for providing GUI (graphical user interface) tools have been available within R for some time. These do not provide the full range of facilities provided by multimedia authoring systems but they do provide a very useful set of tools for adding a degree of interactivity to R operations. While R has, for very good reasons, been constructed with the philosophy of command-driven control, there are some specific operations where GUI control is very helpful and the teaching and learning context provides a large number of examples. Systems which can provide this include `iPlots` (Urbanek and Theus 2003) based on Java and `RGtk2` (Lang and Lawrence 2006) based on the GTK tools. Verzani (2007) provides the `gWidgets` interface which provides access to several different GUI systems in R.

This paper focusses on the `rpanel` package (Bowman et al. 2006, 2007) for R which has two aims. The first is to provide access to GUI tools in as simple and direct a manner as possible. The `rtcltk` package (Dalgaard 2001) is used because of its native presence in R for many years. `rpanel` carries out the management of communications behind the single function calls required to add individual controls. The second aim is to provide higher level functions which use these lower level tools to create useful interactive operations, with a particular emphasis on teaching and learning. An illustration of this is given in Sect. 2, focussing on a simple example involving data exploration and plotting. Sections 3 and 4 extend this into interaction with concepts and models. Some final discussion is provided in Sect. 5.

2 Interacting with Data

Time series data offer a simple, easily understood structure which has many applications and creates interest in the setting of teaching by raising many interesting questions. In the context of climate change, the Central England Temperature data provide a remarkably long documentation of monthly temperature (°C), from 1659 to the present day. The data are available from the Hadley Centre (www.metoffice.gov.uk/hadobs/hadcet/) but are also available in the `multitaper` package (Rahim and Burr 2012) in R.



centre
1835
span
20.33

centre
2012
span
20.75

Fig. 1 The *top plot* shows the Central England Temperature from 1659 to 2011. The three plots in the *second row* show the effects of setting the centre of this window to 1659, 1835 and 2011, respectively, with the span set to 20 years in all three cases. The *bottom row* shows the two slider controls in a separate panel and then integrated with the plot into a single display

The top panel of Fig. 1 plots the entire time series but, with such a substantial amount of data, it is very difficult to assess anything other than the broadest features. It is a simple matter to plot a subset of the data which corresponds to any particular time window. However, repetition of this is rather cumbersome. A very attractive alternative is to use two sliders, one for the centre of the time window and the other for its span, as shown in the bottom left panel of Fig. 1. The three graphs in the middle row show spans of 20 years of data centred on different locations. It is hard to communicate the effect of this in static, printed plots but the advantage of animation and direct interaction is considerable, allowing easy inspection across the entire range of the time series, with simple graphical identification of the seasonal effect and cases of unusually high or low summer and winter temperatures. The physical process of using the slider substantially enhances the psychological effect of flexible interaction with the data. The bottom right panel shows the further convenient step of integrating the sliders and the plot into a single window, using the `tkrplot` package (Tierney 2005).

As R is in such widespread use, it is worthwhile indicating the mechanism by which this type of plot can be constructed, to encourage those with some knowledge of R coding that the addition of interactive controls is a very straightforward step. The starting point is code to plot the data `y` (a time series object) in a time window defined by its `centre` and `span`.

```
w  <- centre + c(-0.5, 0.5) * span
w  <- w + max(0, tsp(y)[1] - w[1])
      - max(0, w[2] - tsp(y)[2])
yw <- window(y, w[1], w[2])
plot(yw, ylim = range(y))
```

The second line of this code simply adjusts the window near the ends of the range of the time series to ensure that the window is always of length `span`. The code segment as a whole can easily be made into an *action* function, as the code below shows. The list object `panel` is the mechanism used by `rpanel` for communication and each action function should return the panel object. The lines of code at the end of the section below simply create a control panel window and add the two sliders with nominated start and end values. As each slider is moved, the action function is called with the new setting and the repeated redrawing which this invokes creates the animation.

```
subset.draw <- function(panel) {
 with(panel, {
   w  <- centre + c(-0.5, 0.5) * span
   w  <- w + max(0, tsp(y)[1] - w[1])
         - max(0, w[2] - tsp(y)[2])
   yw <- window(y, w[1], w[2])
   plot(yw, ylim = range(y))
   abline(h = mark, col = "red", lty = 2)
})
   panel
}
panel <- rp.control(y = y)
rp.slider(panel, centre, tsp(y)[1], tsp(y)[2],
subset.draw)
rp.slider(panel, span, 2 / tsp(y)[3],
diff(tsp(y)[1:2]),
                subset.draw, initval = 20)
```

This simple mechanism allows a wide variety of controls to be added easily to code which may have been created for a limitless variety of graphical and other tasks. A further small step is to encapsulate this kind of code in new functions which offer further flexibility, such as specifying axis labels or adding horizontal lines for reference values. Bowman et al. (2006, 2007) describe the `rpanel` tools in detail and give a wide variety of examples, while Bowman et al. (2010) discuss spatial examples in particular.

3 Interacting with Concepts

The concept of random variation is fundamental to an understanding of probability and statistics, but this can be a difficult concept to grasp when it is met for the first time. A classic mistake is to over-interpret the detailed shape of histograms, scatterplots and other data displays, attributing meaningful structure to features which are simply manifestations of random variation. A simple device to counteract this is to use repeated simulations of data. A simple example is to simulate several groups of data from the same population and observe the apparent differences which can arise when the data are plotted. The upper panels of Fig. 2 use an `rpanel` control to do this, with radiobuttons to select the sample size, a button to create a new

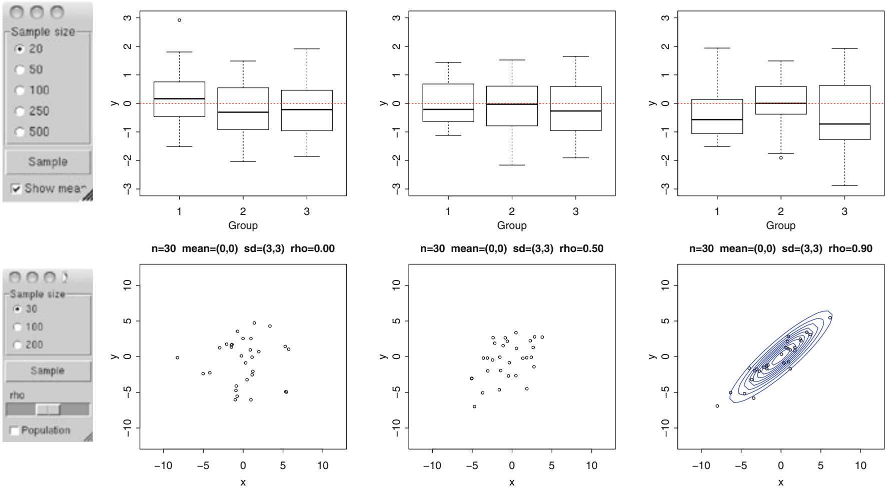


Fig. 2 In the *top row*, the *left hand panel* shows radiobutton controls for sample size, a button to simulate new data and a checkbox to control whether the common mean of the groups is displayed. The following three panels show the results of different simulations. In the *bottom row*, the *left hand panel* shows radiobutton controls for sample size, a button to simulate new data, a slider to control the value of the correlation coefficient and a checkbox to control whether the contours of the true bivariate distribution are displayed. The following three panels show the results of different simulations, with the correlation coefficient set to 0, 0.5 and 0.9, respectively

simulation and a checkbox to control whether or not the underlying true mean is superimposed. The resulting boxplots can sometimes show marked differences, especially for small sample sizes, despite the fact that the underlying means of the groups are identical. Of course, plots like this can be created by repeated execution of a small segment of code, but the GUI controls are extremely convenient and allow rapid and easy investigation of the effects of sample size. This is particularly convenient in a lecture or classroom setting but it also has advantages for student use as it focusses attention on the concept of interest and avoids the distraction of repeated direct execution of code.

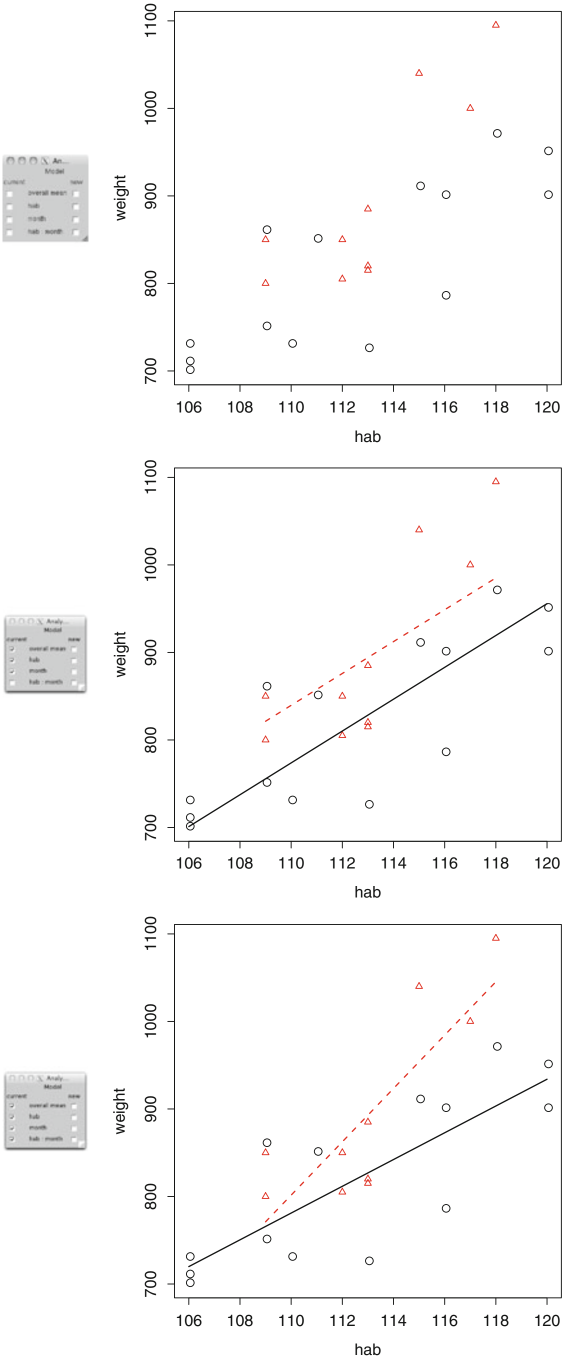
The concept of correlation is another case where intuition and experience are very valuable in interpreting the strength of association in observed data. The lower panels of Fig. 2 show a control panel which allows easy repetition of sampling with specified values of sample size and true underlying correlation coefficient. The contours of the true bivariate normal distribution from which the data are samples can also be superimposed. These are elementary operations in R and, as indicated in Sect. 2, the addition of GUI controls is very straightforward. The advantage of this relatively small amount of additional effort is that an effective and easily used display tool can be created.

4 Interacting with Models

The idea of a model is central to statistical analysis and it is very helpful to be able to plot and compare models and their suitability for observed data. Again, there are potential advantages in encapsulating this in tools which use GUI controls, for example allowing interactive specification of the particular terms involved in the model. Analysis of covariance provides a good example of this. It is straightforward to write code to display the fitted models graphically on a scatterplot of the data. The addition of an interactive control to specify these terms enhances the meaning of each model by giving immediate graphical feedback on the associated changes.

Figure 3 illustrates this on data, available in the `rpanel` package, from a study of the weight changes in herring gulls throughout the year. Some birds were caught in June (coded as month 1) and others in December (month 2). Since weight is dependent on the size of the bird, this information is recorded in the form of the head and bill length, `hab` (in mm), the distance from the back of the head to the tip of the bill. The first graph displays the weight data, plotted against `hab` and colour coded by month. The following two graphs show two fitted models of particular interest, one corresponding to additive effects of weight and month and therefore producing parallel regression lines, while in the other the interaction model relaxes this constraint. The GUI control panel allows terms to be specified simply by checking the appropriate boxes, with immediate graphical feedback in terms of the fitted model. This helpfully reinforces the meaning of the models. There is also an opportunity to give feedback on inappropriate models, such as the presence of an interaction term without both main effects. In a small way, the software then plays the role of a tutor, giving appropriate prompts which encourage suitable modes of thinking in the

Fig. 3 The *left hand panel* shows checkbox controls to specify the terms to be fitted in an analysis of covariance model. The following three panels show the data, a model with *parallel lines* and a model with *different lines*, respectively



student. These facilities are available in the `rp.ancova` function in the `rpanel` package, where a comparison model can also be specified and an F-test used to assess its suitability.

A second illustration is of logistic regression, another model which is a very standard tool in statistical analysis and yet which, in the experience of many students, requires some time and effort to understand when it is first met. Figure 4 shows a well-known set of data on budworms, discussed in Collett (1991) and used as an illustration in Venables and Ripley (1994), where R code is also provided. Data on the numbers killed (from groups of 20) which were exposed to different doses of a chemical are plotted here, for males budworms only. The grouped nature of the data helps in motivating the shape of the logistic link function. The ability to superimpose a logistic regression, with the values of the intercept and slope parameters in the linear predictor controlled by "double-buttons", allows the effects of changing these parameters to be investigated. This promotes intuition on the meaning of the parameters and the process of selecting suitable values to describe the observed data leads naturally to a discussion of scientific principles which can be used for model fitting. Where appropriate within the syllabus, likelihood can be introduced, with further opportunities for graphical display and GUI interaction. Bowman (2007) discusses the uses of `rpanel` in a likelihood setting. The right-hand panel of Fig. 4 shows the fitted model which is produced by maximising the likelihood. These facilities are available in the `rp.logistic` function in the `rpanel` package.

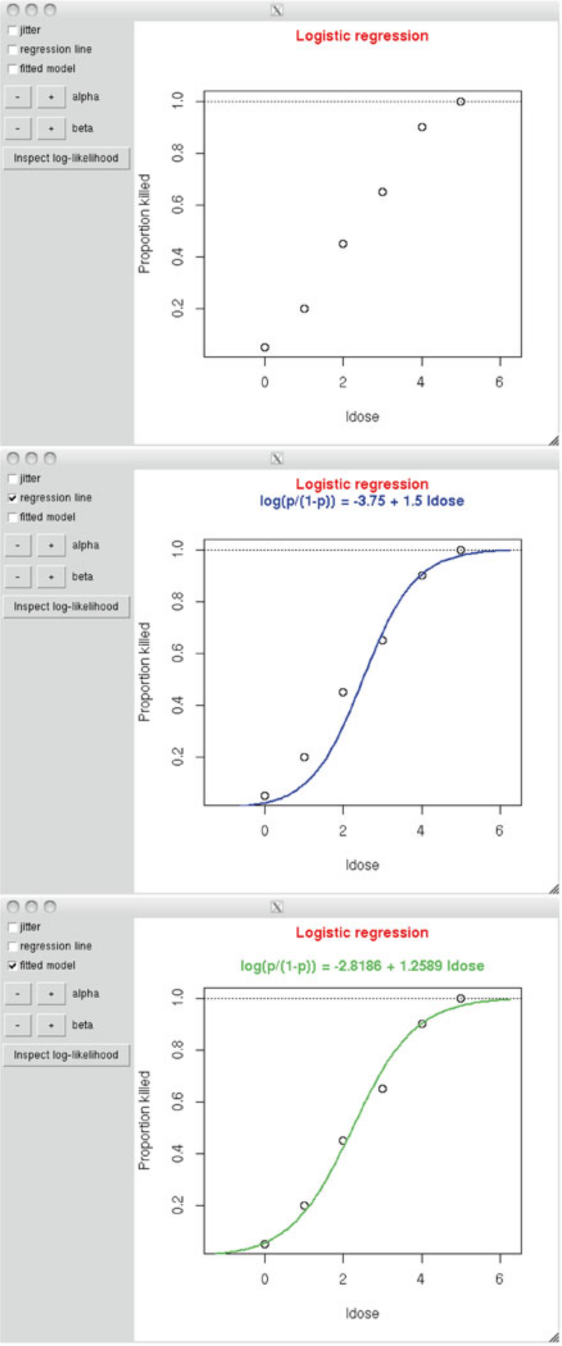
5 Discussion

Opportunities to support the understanding of statistical data, concepts and models have been explored and the use of interactive GUI controls has been advocated. This has been discussed in the context of the R statistical computing environment, which is now very widely used and which provides an enormous variety of data and modelling tools.

The ability to add interactive controls in this rich computational environment allows very useful teaching tools to be constructed. These can be used in a lecture or classroom setting by teachers, to illustrate topics in a convenient but dynamic manner. This mode of use allows discussion to move beyond the scope of static diagrams and communicates the fact that analysing data is a dynamic and exploratory process. The use of animation in particular supports more formal presentations of concepts by illustrating the meaning of parameters or models in a more intuitive manner. The liveliness of this form of presentation also often has a beneficial effect on the attention levels of the audience.

This type of material can also be used by students in laboratory or self-study mode. Again, the aim is to promote more intuitive and conceptual understanding but now with the additional use of reinforcement and interaction. We all know from our own experience of the use of software that users are most comfortable when they have a sense of active control of activity and speed, rather than being placed in the passive role of an observer. Learning is promoted when some degree of self-direction is present, as the learner takes on a degree of responsibility for the process.

Fig. 4 The *top panel* shows checkbox and button controls for fitting in a logistic regression model. The following two panels show a model for specified parameters and a model fitted by likelihood



☐ jitter
☐ regression line
☐ fitted model

☐ jitter
☑ regression line
☐ fitted model

log(p/(1-p)) = -3.75 + 1.5 ldose

☐ jitter
☐ regression line
☑ fitted model

log(p/(1-p)) = -2.8186 + 1.2589 ldose

The particular illustrations discussed in the paper have been at the elementary end of the statistical syllabus, but the tools and techniques described can be applied to more sophisticated data, concepts and models with similar ease and to similar good effect.

6 Software

The `rpanel` package for R is available at `cran.r-project.org/web/packages/rpanel`. Further information on `rpanel` is available at www.stats.gla.ac.uk/~adrian/rpanel.

7 Note

Developed from a keynote presentation at the Sixth Australian Conference on Teaching Statistics, July 2008, Melbourne, Australia.

This chapter is refereed.

References

Bowman, A. W. (2007). Statistical cartoons in R. *MSOR Connections*, *7*(4), 3–7.

Bowman, A., & McColl, J. (1999). *Statistics and problem solving*. London: Arnold.

Bowman, A., Crawford, E., & Bowman, R. (2006). rpanel: making graphs move with tcltk. *R News*, *6*(4), 12–17.

Bowman, A., Crawford, E., Alexander, G., & Bowman, R. W. (2007). rpanel: Simple interactive controls for r functions using the tcltk package. *Journal of Statistical Software*, *17*(9), 1–18.

Bowman, A., Gibson, I., Scott, E. M., & Crawford, E. (2010). Interactive teaching tools for spatial sampling. *Journal of Statistical Software*, *36*(13), 1–17.

Collett, D. (1991). *Modelling binary data*. London: Chapman and Hall.

Dalgaard, P. (2001). The `R-Tcl/Tk` interface. In K. Hornik & F. Leisch (Eds.), *Proceedings of the 2nd International Workshop on Distributed Statistical Computing*, 15–17 March, 2001. Technische Universität Wien, Vienna, Austria.

Lang, D., & Lawrence, M. (2006). Rgtk2: R bindings for gtk 2.0. http://cran.r-project.org.

Rahim, K., & Burr, W. (2012). `multitaper`: Multitaper spectral analysis. R package version 1.0–2. http://cran.r-project.org/web/packages/multitaper.

`R` Development Core Team (2006). `R`: *A language and environment for statistical computing*. Vienna: `R` Foundation for Statistical Computing.

Stirling, D. (2012). Cast: Computer-assisted statistics textbooks. http://cast.massey.ac.nz.

Tierney, L. (2005). tkrplot: Simple mechanism for placing r graphics in a tk widget. http://cran.r-project.org.

Urbanek, S., & Theus, M. (2003). iPlots: High interaction graphics for R. In K. Hornik, F. Leisch, & A. Zeileis (Eds.), *Proceedings of 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, Vienna.

Velleman, P. (2004). *ActivStats*. Glenview: Addison Wesley Longman (Pearson).

Venables, W. N., & Ripley, B. D. (1994). *Modern applied statistics with S-Plus*. New York: Springer.

Verzani, J. (2007). An introduction to gWidgets. *R News*, *7*, 26–33.