Comment

C. Weihs

The book describes a software system called Lisp-Stat designed as a statistical software environment particularly for experimentation with new paradigms in interactive statistical graphics. Since other software systems well-known for supporting dynamic graphics, like ISP (1988), SAS-INSIGHT (1991), and (New) S-Plus (1990), suffer from the drawback of not allowing changes in the implemented graphical methods, there is a real need for systems like Lisp-Stat. Indeed, restricting oneself to an existing graphical framework, that is, suppressing ideas for new graphical tools, sometimes not only hinders creativity, but even prevents complete realization of adequate data representations, as we had to accept in our OMEGA-project (Online Multivariate Exploratory Graphical Analysis, see Weihs and Schmidli, 1990a, b). The OMEGA-system is realized in ISP, preventing the implementation of some of our graphical concepts adequately because of restrictions of ISP graphics. At the time the OMEGA-project was started, the most complete and flexible software environments for statistical graphics were the Data Viewer (see Buja, Asimov, Hurley and McDonald, 1988) and plot windows (see Stuetzle, 1988). Both systems were, like Lisp-Stat, based on Lisp, but both were only available on the exotic and expensive Symbolics Lisp machines. The hope that one could easily overcome all such restrictions with Lisp-Stat was the main motivation for me to work through the book. This hope was indeed fulfilled and itself made the book worth reading. But before discussing Lisp-Stat realizations of graphical tools thus far not implementable in OMEGA, I shall give a somewhat personal overview of the contents of the book.

The book starts with a very stimulating tutorial introducing Lisp-Stat's whole range of possible applications in a form that motivates further reading. The illustrated tools range from basic numerical and graphical operations like summary measures and histograms, boxplots and scatterplots, over the generation and modification of data items like lists, to standard dynamic graphics like spinning plots and scatterplot matrices with plot interaction and linking, and to a demonstration of how easy dynamic simulations can be implemented. Additionally, linear and nonlinear regression, maximization, maximum-likelihood estimation and approximate Bayesian computation are demonstrated. Also, the ease of implementing new functions and methods is concisely indicated.

After the tutorial, all chapters but the last describe the Lisp-Stat language by means of syntax explanation, illustrative code and more extensive examples, intended to be interesting in their own right. Chapters 3–5 present all the basic techniques and tools for programming with Lisp-Stat: the definition and usage of variables and functions, data types, data input and output, control structures, code-writing support, probability distributions and statistical and linear algebra functions. The examples include Newton's root finder, symbolic differentiation, a projection operator and robust regression.

So far, nothing special about Lisp-Stat. For example, the notion of a function is realized very similarly as in New S-Plus (1990), using functions not only in place of subroutines or macros, but also allowing functions to be handled as data, being thus able to specify, for example, the mean function of a nonlinear model very easily. Indeed, such "nonstandard" data have always had to be handled in statistical software allowing "real" nonlinear relationships. For example, one of the early systems realizing special data elements for model equations and model equation systems, the IAS-System Bonn, originated at the end of the 1960s (see Kirchen and Weihs, 1984), introduced such elements because of the need of econometric multi-equation models with nonlinearities in the variables as well as in the parameters to be estimated.

Chapter 6 describes something I had not really looked into before: object-oriented programming. The basic idea of object-oriented programming in Lisp-Stat is to build new numerical and graphical tools at the basis of certain *prototypes*. An object is built from a prototype or from another object by copying, by adding specific information like data in locations called *slots* and possibly by changing the methods to work on the data. Methods associated with an object are activated by sending the object a *message* identifying the method. Lisp-Stat contains a number of built-in prototypes delivered to support, among others, regression and the building of graphical windows, menus and dialogs. Nonlinear regression, for example, is supported by the *nreg-*

C. Weihs is a member of the Mathematical Applications unit at CIBA-GEIGY Ltd., R-1008.Z2.22 CH-4002 Basel, Switzerland.