Statistical Science
1991, Vol. 6, No. 4, 339–362

Lisp-Stat: Book Reviews

EDITOR'S INTRODUCTION

This feature consists of four reviews, and a rejoinder by the author, of the recently published book *LISP-STAT: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*, by Luke Tierney, School of Statistics, Minneapolis, Minnesota. John Wiley & Sons, xiii, 397 pp, $39.95.

In the author's own words:

> This book describes a statistical environment called Lisp-Stat. As its name suggests, Lisp-Stat is based on the Lisp language. It includes support for vectorized arithmetic operations, a comprehensive set of basic statistical operations, an object-oriented programming system, and support for dynamic graphics.
>
> The primary object of this book is to introduce the Lisp-Stat system and show how it can be used as an effective platform for a large number of statistical computing tasks, ranging from basic calculations to customizing dynamic graphs. A further objective is to introduce object-oriented programming and graphics programming in a statistical context. The discussion of these ideas is based on the Lisp-Stat system, so readers with access to such a system can reproduce the examples presented here and use them as a basis for further experimentation. But the issues presented are more general and should apply to other environments as well.

This book can be used as a supplement to several courses on statistical computing and computational statistics. A course emphasizing the use of different programming paradigms could be based on the material in Chapters 3–6. A course on dynamic graphics could use primarily the material in Chapters 6–10.

The chapter headings are:

The Editors hope that this feature will serve as a timely focus of discussion of statistical computing and graphics environments and related man-machine interface issues.

Comment

Ron Baxter and Murray Cameron

In the past, new software systems have been developed as the result of research to find methods to integrate statistical analyses and to simplify their specification and computation. Examples include the development of general methods for specification and computation in analysis of variance by Nelder, Wilkinson and James, which led to Genstat, and the development of generalised linear models and iteratively reweighted least squares by Nelder and Wedderburn, which led to the development of GLIM. In the case of Lisp-Stat, the book describes an abstract statistical environment that brings together:

- methods in dynamic statistical graphics developed in the last 20 years,
- the object-oriented approach of computer science applied to both graphical and statistical objects and

Ron Baxter is Manager, Computers, Software and Networds Program, and Murray Cameron is Manager, Signal and Image Analysis Program, CSIRO, Division of Mathematics and Statistics, P.O. Box 281, Lindfield, New South Wales, Australia, 2070.

- the advances in graphical computing and bitmap displays provided by UNIX workstations and the Apple Macintosh.

This abstract environment has a concrete realisation in XLISP-STAT, a freely available software system by the same author. For the purposes of this review, we will treat the two as one and simultaneously review both the book and the software. We begin with a brief description of the software and book and then discuss Lisp-Stat, in turn, as a general purpose statistical package and as a research tool for investigating the object-oriented approach and for investigating developments in statistical graphics.

OVERVIEW

Lisp-Stat is described as a statistical environment. It is an interactive program, strongly influenced by S (Becker, Chambers and Wilks, 1988), for data manipulation and display and for performance of a number of statistical calculations. It runs on Apple Macintoshes and on UNIX computers, providing dynamic graphics using the SunView and X windowing systems. It is programmable—the language of Lisp-Stat is the Lisp programming language (with some useful extensions) and the capabilities of Lisp-Stat can be extended by dynamically linking compiled C functions (Macintosh) or Fortran subroutines and C functions (UNIX) or by writing Lisp extensions.

The statistical capabilities of the system are basic univariate summary statistics, probability functions, linear algebra and regression, non-linear regression, general maximum likelihood methods including numerical function optimisation methods and some approximate Bayesian calculations. Also available in XLISP-STAT (but not mentioned in the book) are software for one-way analysis of variance and for generalised linear models. Available in the book, (but not apparently in XLISP-STAT 2.1, the version that we have) is code for performing robust linear regression. Because of the general tools available (linear algebra, function maximization) it should be possible to program in Lisp a large fraction of current statistical methodology. As an example, the generalised linear model calculations are performed by iteratively calling the code used by Lisp-Stat for fitting linear regression models.

The standard statistical graphics in Lisp-Stat include line and scatterplots, histograms and boxplots, scatterplot matrices with brushing, spinning plots and dynamic links between different plots of the same dataset. At a deeper level, the software provides very general access to the underlying graphics system, and it is possible to program a wide range of dynamic graphical methods.

LISP-STAT AS A STANDARD STATISTICAL PACKAGE

Whether a reader's aim is to use Lisp-Stat as a research tool or as a statistics package, the first five chapters of the book will need to be mastered, for they describe the language and its main statistical and graphical features.

Chapter 1 provides a brief explanation of the reasons behind the development of Lisp-Stat and of some of the basic decisions made in its design. A summary of these arguments is that Lisp-Stat is a research tool at a number of levels. First, it is an experiment in embedding a statistical system in an existing language. Second, by making that language Lisp, a number of different programming paradigms (and in particular the object-oriented approach) can be investigated and compared. Finally, within the object-oriented approach, different specifications of objects can be tried, along with menu, dialog and graphical interfaces and dynamic graphical methods.

A tutorial introduction is contained in Chapter 2. It describes the main aspects of the language—both the Lisp-Stat extensions to Lisp and the aspects of Lisp that are relevant to its use in a statistics package. These differ in emphasis from a standard introduction to Lisp. The discussion here includes basic calculations, histograms, boxplots, scatter and line plots, getting help, basic data manipulation, dynamic and linked plots, regression, defining functions and methods, nonlinear regression, maximization and approximate Bayes methods.

With a knowledge of this chapter, the package could be used as a basic statistical calculator and graphics system. Several points are worthy of comment.

- On page 60, a simple example is given of the use of a slider to vary the power used in a power transform. In this plot, moving the slider alters the power used and this immediately alters a Q-Q plot of the transformed data. The programming is simple and the response of the animation to variations in the power is immediate.
- The nonlinear regression is very simple to set up and appears to be as good as any that we have seen in a general purpose package. We have not tried it on examples where the maximisation is numerically difficult, however.
- The software seems to make no allowance for missing values. This will severely limit its use

for practical data analysis without substantial programming by the user.

A major problem is the level of documentation of functions available. There is an on-line help function, which makes it easy for the programmer to provide some documentation to each function. However, that help is typically two or three cryptic lines, which is more useful as a reminder than as a way of learning something new. Moreover, we could find no way of printing out a function definition (e.g., in S one just enters the name of the function) or of getting a list of all the functions available (or all the supplied statistics or graphics functions). In addition, this information is not available in the book, so that a browser cannot get a feel for the capabilities of Lisp-Stat by reading a page or two. The software does come with a Technical Report, which has this sort of information available as an Appendix. Some of these problems may be dependent on the version of Lisp upon which Lisp-Stat is built.

Chapter 3 describes the elements of programming in Lisp that are needed to use Lisp-Stat effectively. In this chapter are methods for writing functions, performing iterations and recursions, executing conditional statements and mapping (applying a function to each element of a list). In Lisp-Stat many of the functions have been vectorized, but others have not and this is where mapping is needed.

(Vectorizing is common to all high-level statistics packages. A simple example is that if X represents a vector (or list) of observations then log(X) (or in Lisp-Stat (log X)) represents a vector of the logarithms of the elements of X. The book and its index are a bit sloppy in treating vectorized arithmetic. It is only properly described on page 158, although material on pages 78 and 99 assume that the reader understands all about it. This is a problem on a first pass through the book.)

Lisp is certainly capable of doing all the basic manipulations that are available in existing packages. We have had extensive experience in using S but none previously in using Lisp. For us, Lisp sometimes seemed clumsy by comparison, but it is possible that this is just a matter of familiarity. Selecting subsets of a vector or array (a common task) is simpler in S.

Chapter 4 provides more necessary nuts and bolts material for programming in Lisp, including input/output, and it contains descriptions of the Lisp data types and functions.

Chapter 5 describes the basic statistical functions of Lisp-Stat not described in Chapter 2. The main additions are kernel smoothing and the lowess algorithm, a healthy range of probability distributions, some basic linear algebra and some further discussion on regression computations, including code for robust regression.

If you go no further than this, Lisp-Stat represents a good package for data manipulation and for programming of regression calculations.

OBJECT-ORIENTED STATISTICS

Chapter 6 describes object-oriented programming as it is implemented in Lisp-Stat. Broadly, object-oriented programming is an approach in which data is stored in objects that "know" how to customize generic functions for their own "type." A simple example is the command "print." Printing an object containing data from a two-way table should result in different output to that obtained if the object is a time series. The object needs to know its type and where to find the function (usually called the method) to print that type of object. It is a style of programming that has proved particularly useful in graphics and is seen as having benefits in statistical computing. Some elements of object-oriented programming are present in the current version of S.

This chapter contains descriptions of objects, how to construct objects and corresponding methods, how to construct prototypes of objects and the inheritance of methods. (There is a hierarchy of object types and, if a method is not defined for a particular object type, then it inherits its method from its parent in the hierarchy.) Some built-in prototypes of objects are then described. An important example is the linear regression model prototype, which includes the x and y data, weights for the observations, variable names and case labels and quantities computed for the model such as the residual sum of squares.

Interestingly, the nonlinear regression model prototype is defined by analogy with the linear regression model, rather than as a more general form, so that the nonlinear regression prototype inherits methods from the linear regression prototype, rather than the other way around. The argument used by Tierney in support of his approach is that the nonlinear model analysis can benefit by using approximations to linear model ideas—for example, calculating residuals, leverages, etc., using the Jacobian of the mean function at the estimated parameter values instead of the X matrix. This is a strong argument and may lead ultimately to more thorough and more uniform methods of analysis and inference. For example, inference from a robust analysis is less well defined (using most widely available software) than that from

nonrobust analysis, and even a rough approximation would seem to be better than nothing (a rough answer based on the right assumptions is better than an accurate answer based on the wrong (usually Gaussian) assumptions). If an improved method is developed, it can be included and will override the inherited method.

As mentioned above, there are now available a document and software that incorporate generalised linear models into XLISP-STAT. These models inherit from the linear regression prototype. The link function is an object in its own right.

An object may inherit methods from more than one parent—an example might arise in the case of variables that are the Fourier coefficients of a stationary time series. Such an object could inherit properties from a complex variable prototype and also some properties (methods) from iid random variables. The fact that the variables have an ordering (in frequency) also becomes relevant (e.g., for smoothing), and so the object may also inherit properties from the prototype of ordered variables. It becomes clear that inheritance relationships can be complicated.

The chapter ends with a very brief discussion of other possible approaches to objects for statistical computing, together with some examples. This chapter is 33 pages long and introduces many ideas and ways of implementing those ideas in a short space. The ideas are important and would be new to statisticians who have not been following very closely the computer science and statistical computing literatures. This chapter can serve only as an introduction to the area but should encourage others to identify systems of objects that can simplify statistical computing and unify and consolidate methods of statistical inference. Clearly there will not be universal agreement on the best structure—indeed the appropriate structure will depend on the specific types of analysis to be undertaken.

The development of generalized linear models in the 1970s had the effect of unifying methods of modeling, computation and inference for data of a number of different sorts. Object-oriented programming provides a simple mechanism for implementing this kind of unification. Regression models are the most widely used statistical models and exemplify the ideas, but these ideas will be most useful if they are used on a wider variety of data types—for example, multivariate data, time series, images, etc.

Lisp-Stat provides one approach to object-oriented programming and, by choosing Lisp, Tierney has left the way open for others to experiment with other paradigms within the Lisp-Stat environment.

GRAPHICAL METHODS

Graphics are an integral part of statistical analysis and since the early 1970s there have been investigations to find ways of enhancing plots with the computational and display capabilities of computers. Dynamic graphics are used for several distinct purposes. Some of these are the following.

1. *Identification of relationships between variables.* Three-dimensional rotating plots have been used for identifying "interesting" projections.
2. *Investigation of subsets of data.* A generic example is to highlight in a plot of Y against X those cases for which another variable, Z, is in a certain range.
3. *Algorithm animation.* For example, the bandwidth of a kernel smoother might be chosen by inspecting the different smooths obtained by moving a graphical slider that controls the bandwidth used.
4. *Model definition and interrogation.* Graphical methods may be used to designate the predictor variables to be used in a model, or to display the case labels of particular residuals from a fitted model. Some more complex versions of these are available in DataDesk, where dendrograms from cluster analyses are "interrogated" graphically.

Until the mid 1980s, dynamic graphics required expensive equipment. The Apple Macintosh and MACSPIN (Donoho, Donoho and Gasko, 1985) provided an impressive demonstration that dynamic graphics could be usefully performed on low cost, personal computers. Since then commercial software such as DataDesk(Velleman and Velleman, 1988) and S-Plus, and research software such as SPIDER (Haslett, Wills and Unwin, 1990) have used the Macintosh to develop new approaches to dynamic graphics. Other work has been occurring using UNIX workstations (e.g., Becker and Cleveland, 1987), but there has been no software environment available in which to develop new methods in a portable way. Lisp-Stat, by bridging the Macintosh and UNIX gap and providing a high level language in which to develop methods, is bound to be influential. What does it provide?

Chapter 7 describes the fundamental window system concepts and provides the ways for users to interact with the software using windows, selecting items on menus and responding to questions through so-called dialog windows. Driving this system is strongly dependent on object-oriented programming. Chapter 8 provides the basic tools from which plots of data can be developed.

Chapter 9, entitled "Statistical Graphics Windows," introduces the graph prototype, which has the important feature of "displaying two-dimensional views of points and lines in m-dimensional space." Allowing the basic data to have m dimensions rather than just 2 seems fairly important for future developments. With dynamic displays becoming widely available, the 3D view would seem to be the basic display with points, lines, planes, symbols, images and text as the "plot types." This is particularly useful in some applications where the basic data are spatial, (e.g., in medicine and geology) where the spatial context is important and where there are images of some "slices" as well as spot values of other variables. An example of the display of regional data using Lisp-Stat is given by Baxter, Cameron, Fisher and Hoffmann (1991). Another useful facility for spatial data, which has been shown in SPIDER, is the ability to overlay several different plots and plot types (e.g., an aerial photo and contour lines showing soil pH) overlayed with sampling points for some other variable also marked. Chapter 9 gives a method for doing that also, as well as providing ways of linking plots and for altering the interpretation of mouse clicks.

The final chapter contains a number of examples of dynamic graphics which use different aspects of the system. These range from a different way of programming the power transformation example, through methods of changing the effect of the mouse, grand tours of multivariate data and finally to an implementation of a different approach to linking plots. The breadth of these examples highlights the power of Lisp-Stat, and the methods used will be useful examples from which users will be able to develop their own applications.

SUMMARY

Lisp-Stat is a high-level language, in many ways like S, but with the advantage at the present time that it can get to lower level operations, particularly graphics operations. It has only a fraction of the statistical operations available that S has, but the result is a system that is leaner and more transportable. Primarily, Lisp-Stat is a research tool rather than a production system. It provides an easy way to perform statistical research and to prototype and customize systems for statistical analysis using windows, menus and graphical interfaces. It requires most users to learn a new language and provides only limited statistical capabilities and data handling facilities.

This book and software are certain to be influential. In teaching, they may well change what is taught in statistical computing and how it is taught. They will act as a considerable stimulus for research into dynamic graphics and into methods of programming statistical computations and analyses. At one level, Lisp-Stat may be found to be a suitable system for computing in data analysis and early statistics courses, for it provides many of the needed computations, but not perhaps all that may be needed and so the programming language may be used.

At another level, Lisp-Stat provides opportunities to challenge students in programming graphics and statistics and seems likely to bring more computer science to statistics than ever before (or perhaps it will make statistical computing more attractive to computer scientists). In statistical practice, Lisp-Stat will make dynamic graphical analysis more widely available and more readily performed. In statistical research, this software will make some areas much easier to be involved in without a major investment in the development of underlying software.

The development of commercial software will also be influenced by the presence of Lisp-Stat. This follows the tradition of more widely known, freely available software such as the Berkeley modifications to UNIX and the X windowing system. In both cases the freely available software has set standards but most users run an implementation from a commercial vendor.

Lisp-Stat cannot be ignored by anyone interested in statistical computing.