Statistical Models in S
edited by
John M. Chambers and Trevor J. Hastie
1992
608 pages
ISBN 0-534-16765-9
Wadsworth and Brooks/Cole Advanced Books and Software

Contents:

New programming functionality has been added to the *New S* language since the publication of the *New S* manual in 1988 (The New S language: A programming environment for data analysis and graphics, by R.A. Becker, J.M. Chambers, and A.R. Wilks). By using this extension to the *New S* language, the ten authors of this manual are able to develop a unified approach to the fitting and analysis of a fairly complete collection of response models (traditional and recent). The book represents the first major effort in this area. I highly recommended it to anyone interested in using New S, or in applying more recent response models, or in research in statistical computing.

The book is well organized and reads more like a single treatise than a collection of papers. The editors and the authors are to be congratulated for a remarkable job. Each chapter is organized into four primary sections. The first describes the statistical methodology, the second how to use the S functions and data structures, the third how to extend or specialize the given software, and the fourth contains more detail on the computations. Consequently, by reading only the first two primary sections of each chapter one comes away well equipped to use the software in a host of applications. For most readers this will be enough. As interest and circumstance demand, the remaining sections of any chapter can be read with profit. Although early chapters are required reading for later chapters, chapters 7 through 10 can be read independently of one another.

The language extension has moved New S in the direction of object-oriented programming. For S users this is an important and exciting development and will expose many of them to some of the ideas of object-oriented programming for the first time. Because of this it is a pity that the extensions of the language take New S such a small step towards realizing the power of object-oriented programming.

As it stands, New S is quite different from what is generally understood to be an object-oriented programming language. As is pointed out in Appendix A of the book, New S has much in common with object-oriented languages *but differs in a number of respects related to the nature of S* (p. 457). Chief among these differences is that while New S purports to be a *class based* object-oriented programming language, it falls short of satisfying any definition of one. Despite the impression given to the casual reader, there is no such thing as a *class* in this New S; there is merely an attribute called *class* which can appear on any S data structure.

Specifically, New S's functional programming style has been extended so that the user can write functions which dispatch to other functions (called *methods*) depending only on the value of the *class* attribute of one of its arguments. True, this makes it possible to write functions which are *generic* but in my opinion it is not quite object-oriented programming. Instead, the function dispatching (*method lookup*) in the New S model has been confounded with the definition of a class.

By contrast, in a class-based object-oriented programming language, classes are data structures which can themselves be manipulated. Minimally, they can be related one to another through the notion of inheritance. No such data structures exist in New S. Because New S's class attribute only determines the method lookup to be used for a particular instance it might be better understood by the reader had it been named the *method-precedence* attribute.

Why should the reader care about the absence of classes? There are many technical software engineering reasons but the statistical reasons are that classes can be used to directly represent statistical concepts and the relationships between them. For example, consider implementing generalized linear models (glm) and standard linear models (lm) with genuine classes. Because a linear model is really a special kind of generalized linear model one might naturally define two classes, say *lm* and *glm* and assert that *lm* is a specialized subclass of *glm*. As a consequence, whatever property one expects of a *glm* would also be found on an *lm* through *inheritance* since it is simply a special kind of *glm*. The two models have statistical meaning and conceptually are interrelated; having related class structures preserves and enforces this meaning.

By contrast, in the extended New S system, the class attribute of a *generalized-additive-model object* or *gam* is defined to be (*gam*, *glm*, *lm*). If this were a true class hierarchy list one would read this as follows: the primary class of the object is *gam* which inherits from *glm* and failing that inherits from *lm*. This seems completely backwards to me. I would place (as we have in the Quail system) *gam* at the top of the class hierarchy and *lm* at the bottom. Thus the class attribute of a gam would be simply (*gam*) while that of a linear model would be (*lm*, *glm*, *gam*). This seems to me to better capture the statistical meaning of these models.

No class data structures means no precise definition of a class. Two New S objects can have completely contradictory class information, yet share the same class as their primary class. For example one object could be defined with class attribute (A, B, C) while another has class attribute (A, X, Y). Can either represent the class hierarchy of A? One is reluctant to attach too much meaning to a New S class.

In summary, the book and the attendant software are interesting, valuable, and important. The book should be of interest to a wide audience. The authors are to be congratulated for an excellent well organized book that represents the first major effort towards a unified presentation of statistical response models and which takes New S an important first step towards object-oriented programming. My only caveat to the consumer is that as an object oriented programming system the extended New S system is unusual and in my opinion falls short of fulfilling the promise of object-oriented programming.

R.W. Oldford
Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario
N2L 3G1
Canada