available, free-of-charge, to others. There is, in essence, no support provided, other than an effort by Tierney to fix bugs, and minimal on-line help. This, of course, is a mixed blessing, but since the quality of the software is very high, and since there are few bugs, there seems to be little need for support.

The book, of course, is not free, though it is reasonably priced. I have already described all of the chapters in the book, emphasizing that the book is especially strong on dynamic graphics. The book is useful as Lisp-Stat documentation, providing a tutorial and examples of using and extending the system. The book is also a good introduction to functional and object-oriented programming, as used in statistics, and to Lisp.

### 10. CONCLUSION

Lisp-Stat is the most important, exciting and promising development in computational and graphical statistics in recent years. It provides a foundation on which computational statisticians can build a statistical system offering all types of statistical and data analysis tools—from basic to advanced—to all types of users—from novices to sophisticates. As it stands, Lisp-Stat is not (and does not claim to be) a statistical system that provides a wide range of analysis tools for a wide range of users. However, with the proper extensions, Lisp-Stat could become the standard by which other systems are judged. In sum, Lisp-Stat is the statistical environment "for the best of us," not "for the rest of us"—yet.

# Comment: Two Functional Programming Environments for Statistics — Lisp-Stat and S

**David J. Lubinsky**

### 1. GENEALOGY

There is a German saying, "Tell me where you come from and I will tell you who you are," and this is perhaps even more true for the two statistical environments that are the subject of this review. They are both the products of many ancestors and each reflects its heritage. Both Lisp-Stat ($L_S$; the idea of this notation is that Lisp-Stat is a Lisp system specialized for statistics) and S are descendants in the line of interactive, interpretive systems, starting with APL and Lisp; $L_S$ also draws inspiration from S, Smalltalk and dynamic graphics systems (Cleveland and McGill, 1988). The two systems are both interpretive programming environments using functional languages. They each have vectorized arithmetic operations and support a wide set of statistical primitives. In addition each has strong support for graphical display of data.

In the family of statistical and computing systems, $L_S$ and S are very close, and as in all families, there is a natural rivalry, but also a natural affinity between them. S is the older brother, more mature and more complete. Whereas $L_S$ is faster, incorporates many new ideas in graphics and object-oriented programming, but still has a long way to go before it can compete with S in all areas.

They are both designed to be used for more than canned analyses of data. Each allows users to combine standard analyses in nonstandard and flexible ways and, more importantly, to implement and experiment on new techniques.

The rest of this section introduces each system by presenting the same function coded in each language and discusses the general areas of application in which each system is stronger. The following section is a more detailed comparison of languages and primitives in each system. This is followed by a comparison of performance of the systems, and the last section discusses the their documentation.

#### 1.1 A Running Example

To help make the similarities and differences more concrete, Figure 1 shows how one would implement a running smoother using S and $L_S$. Each function takes two required arguments, **x** and **y**, and returns a set of smoothed values at equally spaced points along the range of the **x**'s. They also take two optional arguments, the function to be used to find local values of the smooth, and the number of points in the returned smooth. Two obvious examples of local smoothing functions would be the mean (the default) and the median. These

*David J. Lubinsky is a member of the technical staff at AT&T Bell Laboratories, Room 25524, Crawfords Center Road, Holmdel, New Jersey 07733.*