Lost Opportunities: Why We Need a Variety of Statistical Languages

Sanford Weisberg

University of Minnesota

Abstract

To the worker who only has a hammer, we are told, everything looks like a nail. Solutions to applied statistical problems are framed by the limitations imposed by statistical computing packages and languages. For better or worse, we can do what the packages do; we cannot do what the packages won't do. Statistical languages like R have basic tools that allow the analyst to design new hammers, but even in R we cannot build an arbitrary hammer, only ones within the limits imposed by the R language. XLISP-STAT imposes different limitations, so we can produce different hammers.

In this article, I look at some of the tools in XLISP-STAT that allow the user to think about graphics in ways that cannot be easily replicated in other statistical languages. The interactive graphical methods available in XLISP-STAT lead to very different methodology than would be developed without the tools that XLISP-STAT provides. The general approach to graphics and indeed to data analysis in general is quite different in a package like **Arc** that is built on top of XLISP-STAT, than it is in other statistical packages. We discuss why that might be true, and why this depends on design options created by XLISP-STAT.

Keywords: graphics, computing, XLISP-STAT, **Arc**.

1. Introduction

In the spring of 1988, Dennis Cook and I submitted a proposal to the National Science Foundation to study "statistical graphics," which was to be a project for developing statistical methodology and theory that would complement recently developed ideas forwarded by John Tukey and others for graphical display of data, such as motion for three-dimensional plots, plot linking (Stuetzle (1987)), and other interactive methods; the state of the art in 1988 is summarized by the papers in Cleveland and McGill (1988). We thought at the time that new methodology and theory should to be developed that could properly exploit these graphical methods. NSF agreed with us, but they did not believe that we could actually *do* this work

because neither of us were expert programmers, and no extensible high-level software for the type of graphics we wanted to use was available. On the contrary, we told them, we could do all the computing necessary using Tierney's then-nascent XLISP-STAT (Tierney (1990)), and to prove it we flew off to Washington lugging an Apple Macintosh modified to have a video-out port for projection. We did our show for the NSF staff, and our project was funded.

I repeat this anecdote because it illustrates my general theme in this article. XLISP-STAT was, and indeed still is, a brilliant platform for statisticians who either are not experts in core languages like C, or are impatient, and unwilling to invest the time needed to program a new idea in a general-purpose programming language. It includes (1) access to Lisp, a well documented computer language for writing code; (2) high-level computing tools, such as function minimizers and linear algebra routines; (3) high-level graphical constructions that could be used for quickly creating a variety of plots; (4) lower-level graphical tools that could be used to build a variety of graphics or graphical constructions as needed; and (5) an object-oriented computing system. All of these features exist on their own in various places, but the brilliant feature was putting them all together in one package. The object oriented system turns out to critical to many of the uses of XLISP-STAT. Everything that is computed in XLISP-STAT can become an object, including a graph, a point in a graph, a slide bar added to a graph, a regression model, and so on. By keeping track of the objects that have been created, the objects seem to come alive, meaning that the user can interact with them and change them. For example, a point in a graph can be selected using the mouse. The analyst can then decide what action should be taken. Simple actions might include coloring the point or changing its symbol, or identifying the point either by writing a label for the point of the graph, or highlighting the label in an associated list of labels. More complex actions might include deleting the point from the graph, notifying a list of other objects that the data has been changed and they must update themselves, and then redrawing graphs as needed to correspond to the modified data. This, in turn, can change the way that one thinks about statistics. For example, Cook's distance (Cook (1977)) summarizes change in a regression when a case is deleted. Like many other statistics, it is an abstraction since it is based on a norm, and does not directly indicate the changes that will occur if a case were deleted. With a regression system that is alive, it is possible to delete the case and *see* what happens

The XLISP-STAT code that I wrote was eventually collected into a package, at first for linear regression only, and then eventually for other types of regression problems. This collection of functions and prototypes was called the R-code, short for *regression computer code.* It was a collection of mostly graphical extensions to the original prototype for regression that Tierney included with XLISP-STAT, not a separate package. Many of the methods that were included in the R-code were developed from our original and subsequent NSF grants. Cook and Weisberg (1994a) uses many of the features of the R-code.

R-code did not have its own user interface, but rather depended on the Lisp interpreter. Reading data into the program required writing a few lines of Lisp, an intimidating prospect for those unfamiliar with Lisp. As a result, the startup cost for R-code was too high for most people. Research-oriented programs like S had a much simpler command-line interface with much better documentation, and commercial programs were appearing with graphical user interfaces. Using the tools provided in XLISP-STAT, I wrote a menu-based graphical user interface for user interaction with the program. In addition, a simple mechanism for the program to get data from a plain data file that could be produced by any standard statistical, spreadsheet or database package was provided. R-code was renamed **Arc**, not an acronym for
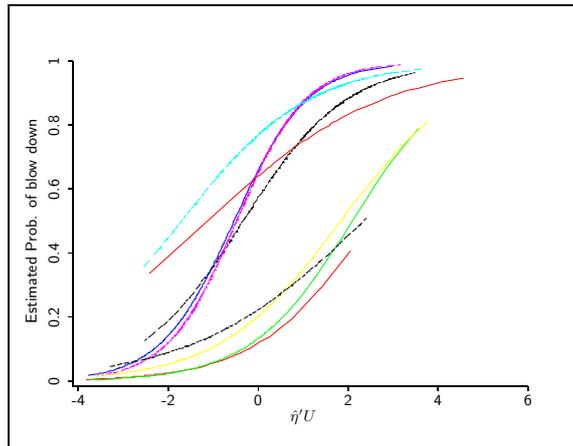
Figure 1: Estimated probability of blow down for the BWCAW data. Based on the left-endpoints of the lines, the species are, from highest probability of blowdown to lowest, jack-pine, black ash, aspen, red maple, paper birch, cedar and red pine. The horizontal axis is a linear combination of predictors estimated using a partial one-dimensional model, Cook and Weisberg (2004).

anything, but rather a simple name with a vague illusion to a graphical object. **Arc** became both a platform for research into graphical methods (see, e. g., Cook (1998)), and a statistical package that can be used in teaching and doing regression analysis. We wrote a regression textbook, Cook and Weisberg (1999a), that provides an approach to regression based on graphics with *Arc* as an integral part of the work. The book provides the documentation for the program. For the student, the presence of XLISP-STAT in **Arc** is nearly hidden, as the user need never write a single line of Lisp code to analyze data. But without XLISP-STAT, there could be no **Arc**.

In the rest of the article I discuss methods and ideas that occur naturally when using the object system available in XLISP-STAT for graphics, but are more difficult, or even impossible in statistical computing systems that lack these features. I illustrate my point using three general ideas: designing graphics for analysis rather than for presentation; the development of an analytical graphical system, and a particular problem of selecting transformations using graphics. The article ends with more discussion and conclusions.

2. Presentation graphics versus analytical graphics

I like to make a distinction between *presentation graphics* and *analytical graphics*. Presentation graphics are designed to be shared with others, usually on the printed page or more recently via a web page. They are designed to tell a clear, elegant, persuasive, story. The graphs in newspapers and magazines are extreme examples of presentation graphics, often cluttered with distracting art or misleading graphical elements. Graphs produced by statistical software usually avoid the distractions, but have the same goal of allowing the creator of the graph to tell a story. For example, Figure 1 summarizes the probability of a tree being blown down during the July 4, 1999 storm in the Boundary Waters Canoe Area Wilderness, as a function of two predictors, with a separate curve for each of seven species (see Cook and
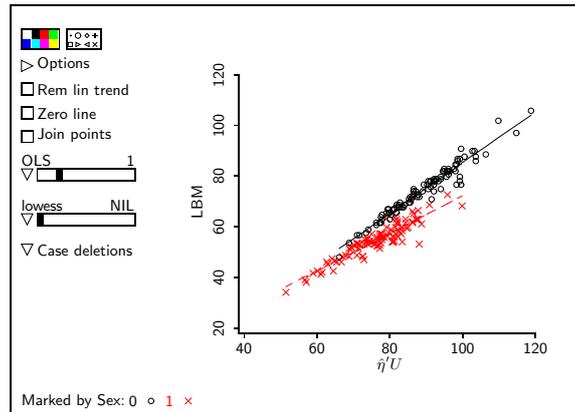
Figure 2: Australian athlete data. Horizontal axis is a linear combination of *Height*, *Weight* and *RCC* the red cell count, and the vertical axis is for *LBM*, the lean body mass. Separate colors and symbols are used for males and females. The slider plot control at the left of the graph marked OLS at the left of the graph as been moved to the value 1, indicating that the OLS regression of degree one, simple linear regression, has been fit to each colored group. Separate fitting to each group was specified by an item in the popup menu for this slide bar.

Weisberg (2004), for details). The graph was produced in **Arc**, and summarizes the results of the analysis without indicating how the analysis was done, what sort of models were used, or how well the models used represent the data. What is shown is that for all species the probability of blowdown increases with the linear combination plotted on the horizontal axis, and that the species form two groups. I think of this as a *dead* graph, although the less evocative term *static* graph might be more appropriate. While the graph tells a clear and possibly interesting story, the viewer is not invited to interact with the graph.

Analytical graphics are designed for discovery, and include both the *exploratory* and *confirmatory* graphics advocated by Tukey (1980). The creator of the graph may not know what to expect or the story that the graph will tell. Interacting with the graph can be helpful. The statistical literature includes countless analytical graphs, for example of residuals, likelihood functions, and many others, that are primarily for the consumption of the analyst, not for a larger audience. Discovery tools should invite the viewer to interact, quickly, efficiently, and using a consistent interface. To facilitate interaction, graphs can include *plot controls*, which are sliders, checkboxes, menus and possibly other graphical elements not yet devised to let the analyst choose an appropriate interaction. *Direct manipulation tools*, in which the mouse is used to change a graph, have an intuitive appeal for viewers, but we did not use them much in **Arc**. **Arc** includes a core set of standard plot controls that depend on the type of plot, so, for example, the plot controls for a 2D scatterplot are different than the controls for a histogram. Additional controls might be available for a particular 2D plot, depending on the quantities plotted.

Figure 2 exemplifies an analytical graph in **Arc**. The plot controls at the left provide the analyst with visual cues about the interactions that are available. As shown, smoothers, really simple linear regression fits, have been added to the plot separately for each of two colored groups. Were this smoother to be inadequate, others could be selected from the plot controls. Feedback is immediate because the plot is updated according to the controls used.
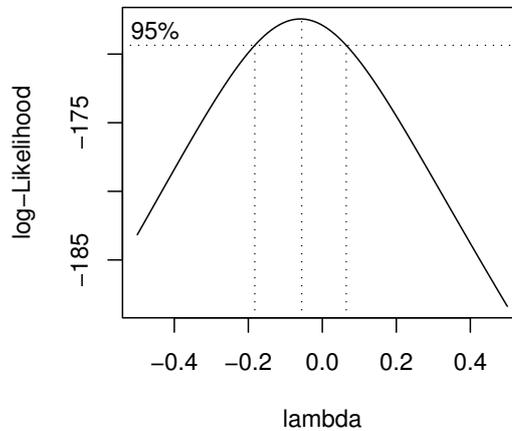
Figure 3: The profile log-likelihood for selecting a response transformation using the wool data, with `boxcox` from R.

This leads to the general research question of which controls should be in the core set and how these interactive tools can be used; we discuss these at length in Cook and Weisberg (1994a), Cook and Weisberg (1999a). I'm sure there are more ways to use interactive graphics both for regression and many other statistical problems. Without a system that allows for interactive graphics, we will never know just how important interactivity is to the analyst.

2.1. Response transformations

The basic tools available can determine how we study and solve a problem. In linear regression analysis, a now-standard problem is to select a transformation $t(Y)$ of the response Y with predictors X so that in the transformed scale $t(Y)|X \sim \mathrm{N}(X\beta, \sigma^2 I)$. The most familiar methodology was proposed by Box and Cox (1964). They suggested first parameterizing $t(Y)$ via a family of transformations, so $t(Y) = t(Y; \lambda)$, often using the power family

$$t(Y; \lambda) = \begin{cases} (Y^\lambda - 1)/\lambda \mathrm{gm}(Y) & \text{if} \quad \lambda \neq 0 \\ \mathrm{gm}(Y) \log(Y) & \text{if} \quad \lambda = 0 \end{cases} \tag{1}$$

where $\mathrm{gm}(Y)$ is the geometric mean of Y. If we behave as if the distribution of $t(Y, \lambda)|X$ were normal for all λ, then we can estimate λ from the profile log-likelihood for λ.

This idea is implemented as a graphical procedure by Venables and Ripley (2002) for the power family of transformations in their **MASS** library for R/S-PLUS. For the well-known wool data from the Box and Cox paper, the graph produced by the `boxcox` function in R is shown in Figure 3. This graph summarizes the need to transform via the abstraction of the profile loglikelihood function. The estimate of λ is close to zero, meaning a logarithmic transformation, maximizes the log-likelihood, and we can read a confidence interval from the graph using analogies to standard likelihood theory.

If we start with interactive graphics, we can take as the goal an appropriate graph that allows
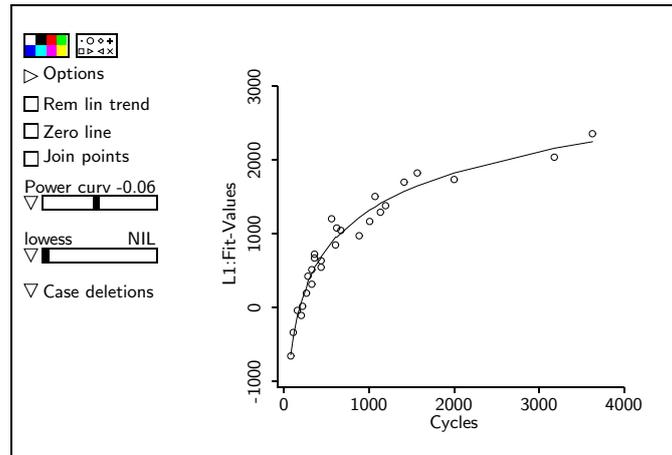
Figure 4: Inverse response plot from **Arc**.

us the *see* the result directly. After developing a bit of theory, Cook and Weisberg (1994b) suggested that under reasonable conditions on the predictors, first fit the regression of the untransformed Y on the predictors, and then plot with Y on the horizontal axis and the fitted values from this regression the vertical axis, what we called an *inverse response plot.* If a parametric family of transformations is used, λ can be estimated by fitting a mean function of the form $\mathsf{E}(\hat{Y}|Y) = \gamma_0 + \gamma_1 t(Y, \lambda)$ to this graph. Since for given λ this is just a linear regression fit, λ can be selected visually using a plot control that will select λ, do the fit, and add the fitted curve to the plot. Alternatively, λ could be selected via the plot control to satisfy some optimality criterion. This plot, with the plot control set for $\lambda = 0.06$ is shown in Figure 4.

This interactive approach to selecting the transformation has several advantages over simply displaying the log-likelihood profile. First, the data appear in the plot, and we can see if the fitted curve matches the data. For the wool data, the curvature is apparent for all the points in the graph. Outliers and influential cases for selecting the transformation can be easily discovered. Small changes in λ can be assessed. This graph can show if a transformation outside the family $t(Y, \lambda)$ is desirable, and can be helpful to estimate the transformation, perhaps using a smoother. Fitting inverse plot curves like the one used for selecting a transformation can be useful in other circumstances, such as transforming predictors (Weisberg (2005)), and so this control is a standard plot control for 2D plots in **Arc**.

The interactivity between the analyst and graph makes the inverse response plot an attractive method for selecting a transformation, because the analyst can really see what the transformation does, and whether or not it works. Without the interactivity, this method becomes much more tedious, and it is not certain that it is then superior to looking at the log-likelihood.

2.2. Differences

The distinction between analytical and presentation graphics is certainly not always perfectly clear. For example, the trellis/lattice graphics (Becker, Cleveland, and Shyu (1996)) for S-PLUS/R provide an elegant graphical system that produces beautiful presentation graphs. Although not interactive, plots produced with trellis can also be used in the midst of an

analysis. Interaction can sometimes be simulated by redrawing a graph after modification, or in trellis graphics by using the ability to condition on extra variables to view several related graphs at once. Nevertheless, these are static graphics, and the way one thinks about graphs is different with trellis graphics than it is with truly interactive graphics. In principle one would implement trellis type graphics in an interactive system, which could lead to interesting analytical results. To reiterate again, interactive graphics can lead to using different statistical methodology.

3. Plot controls

The plot controls shown on Figures 2 and 4 are the standard set of controls that appear on all two-dimensional plots in **Arc**. Both consistency and flexibility are important. A consistent interface allows a user to learn the function of a control, and then its appearance is expected on all similar graphs. Check boxes are for on/off type enhancements, like adding a horizontal line at zero. A triangle pointing to the right implies a popup dialog. The "Options" dialog allows changing labels, ranges for the axes, and other enhancements that can help particularly when the graph is to be converted from an analytical graph to a presentation graph. Popup menus are indicated by a triangle pointing down and are used to select from a list of options. The "Case deletions" menu on Figure 2, for example, has several options on deleting, restoring or highlighting points corresponding to deleted cases. Slide bars allow selecting a value like a smoothing parameter; plots are immediately updated. In addition to allowing the analyst select a smoothing parameter visually, this particular slide bar serves a pedagogical goal in showing just what an under-smoothed and over-smoothed curve might look like. Slide bars can also have popup menus that change what the slide bar controls. For example, the slide bar marked "OLS" is called the *parametric smoother slide bar*. As the slider is moved to the right by clicking the mouse button, a polynomial of degree shown in the slide bar is added to the plot, and so the choices on the slide bar are integers from one to five. The option "Power curve" from the popup menu was selected in Figure 4 to draw the power function described previously. Additional options in the pop-up menu allow adding choices to the slide bar, fitting curves to differently color or marked points, and the like. While not all options are relevant to all graphs, the common interface helps the user know what to expect. The variety of tools available to the user from the few plot controls is large.

Flexibility is also helpful to allow additional controls to be added to particular graphs. An important feature of XLISP-STAT is that it is possible to modify the standard plot controls for a particular type of graph. For example, a slide bar can be added to select the quantity plotted on one or both of the axes according to a specified list. This can allow the analyst to view many graphs quickly, for example to view a set of added-variable plots. Tools to translate this low-level graphical function to a higher-level were created for **Arc**.

Three-dimensional "spinning" plots are a poor choice for presentation graphics because they require motion to be effective; see Figure 5. The statistical community has mostly dismissed three-dimensional graphics with motion as long on the "wow" factor, but short on usefulness. One serious attempt to use three-dimensional graphics outside of XLISP-STAT is in Xgobi (Swayne, Cook, and Buja (1998)), but the goal of that work was to use three-dimensional graphics to address a particular type of problem, not to create general purpose tools.

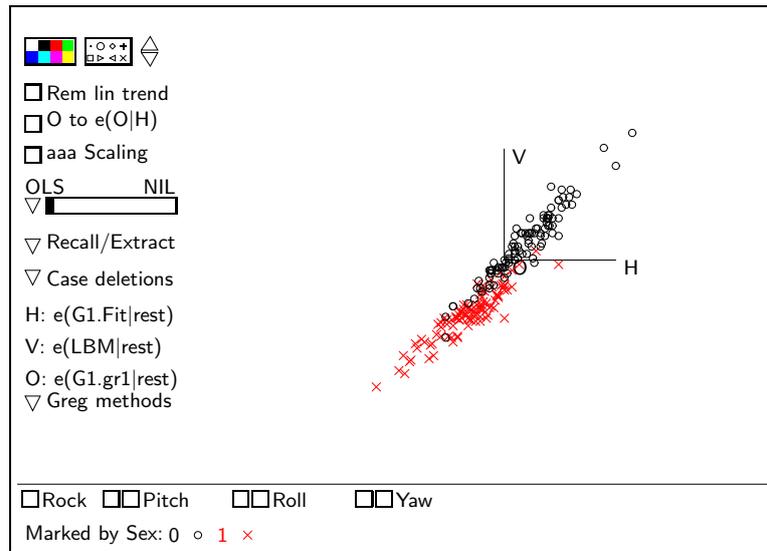XLISP-STAT has a basic prototype for drawing three-dimensional plots, and a suite of primitive

Figure 5: A static version of a three-dimensional plot. Viewing the third dimension requires motion. This particular plot is a *three-dimensional added-variable plot*, used in graphical regression as described in Cook and Weisberg (1999a), Chapter 20. Apart from the popup-menu called "Greg methods," the other plot controls shown are part of the standard interface in **Arc**.

tools for working with them. Because the tools exist, we are free to ask questions about these graphs. What can be learned from a three-dimensional plot that cannot be learned from two-dimensional plots? What theory is required to understand these plots? What plot controls should be available for them? Can others be taught to use these graphs? We have studied these questions extensively; see, for example, Cook and Weisberg (1994a), Cook and Weisberg (1999a), Cook and Weisberg (1999b), and Cook (1998). For example, in Cook and Weisberg (1999a), Chapter 20, we discussed graphical regression, in which we infer about the dependence of a response on a set of predictors solely through looking at three-dimensional graphs. Figure 5 is a three-dimensional added-variable plot that would be used in the midst of a graphical regression. We added to the graph special purpose plot controls collected in the "Greg methods" popup menu that guides the actions of the analyst. This is in addition to the standard plot controls that add smoothers, extract two-dimensional views into their own window with the usual 2D plot controls, control rotation and the like. Ideas for methods that specifically exploit graphics through interaction can be built with XLISP-STAT; without it, there would have been no development of graphical regression, or many of the other methods implemented in **Arc**.

XLISP-STAT was not designed to produce beautiful graphs. On Unix/Linux and Windows, XLISP-STAT would save only a bitmap version of a window; the Macintosh version was better. Bret Musser and I wrote an add-on to **Arc** that uses the information in a graph object to write a file that, when compiled using LaTeX, produces higher-quality PostScript graphs, optionally stripping off the plot controls. This effectively converts an analytical graph into presentation graphs like those shown in this article. The add-on is available from the web address given at the end of Section 5.
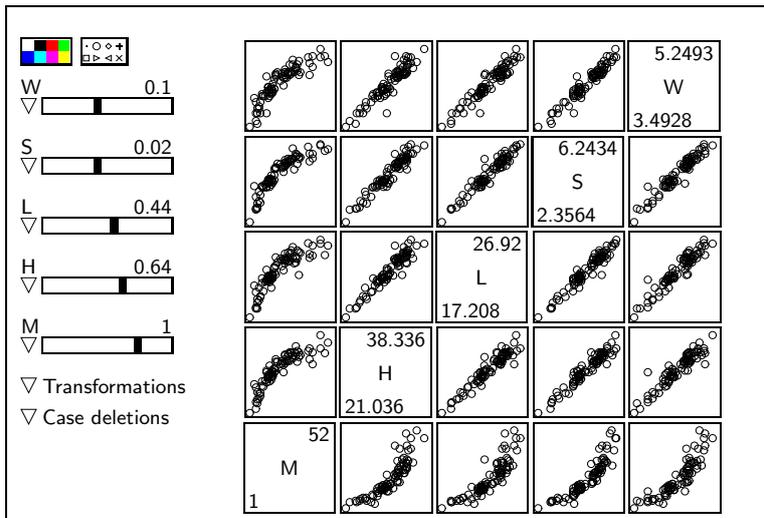
Figure 6: Scatterplot matrix for the mussels data described in Cook and Weisberg (1999a). Each variable in the plot has its own slider as a plot control for selecting a power.5 transformation for that variable. The "Transformations" popup menu has additional options for transformations. By pointing the mouse at a plot with option-shift-click, the 2D plot is opened in a separate window with the 2D plot controls available. The plotted values are scaled power transformations of the variables shown, with transformation given by that variable's slide bar. For example, the plotted values for L are $(L^{0.44} - 1)/.44$. The scaling is used so that the shape of a plot does not abruptly change when the power changes from positive to negative.

4. Transforming predictors in a regression problem

We return to the transformation problem, but this time in a multivariate setting. Suppose we have a set of strictly positive variables $X_1, \ldots, X_p$, and the goal is to find transformations $t(X_1, ; \lambda_1), \ldots, t(X_p; \lambda_p)$ using the transformation family defined at (1) so the transformed predictors are as close to multivariate normal as possible.

One characteristic of the multivariate normal is that the mean function for all conditional distributions is linear. In particular, multivariate normality implies that the mean function for all 2D scatterplots is linear; linearity of all 2D scatterplot is a weaker condition than multivariate normality, however. Using an interactive graphical system like XLISP-STAT, we can explore transformations toward normality using a scatterplot matrix. Figure 6 is a typical scatterplot matrix that appears in **Arc**. In addition to a few plot controls inherited from 2D plots, there is a slide bar for each variable plotted; this will transform that variable and immediately update the whole plot. The popup menus include items related to the transformation slide bars, such as using the marginal Box-Cox method to transform the variable, adding a value to the slide bar, or adding a transformed variable to the data set.

These tools make selecting several transformations simultaneously possible, even with many variables. Of course, there is nothing to stop us from adding a few numerical tools to help guide the choice of transformation. Velilla (1993) proposed a multivariate extension of the Box-Cox method, which amounts to selecting the transformation parameters to minimize the determinant of the sample covariance matrix of the predictors in the transformed scale. This minimization is remarkably easy to program in XLISP-STAT, requiring only about three lines

of code, and in concert with the graph and the slide bars provides a powerful visual system for selecting transformations. Unlike the univariate case, we cannot draw a high-dimensional log-likelihood profile, so there is no useful graphical summary of the log-likelihood abstraction.

How are graphics to be used without interactive tools? The answer appears to be, with great difficulty. Simultaneous transformations require looking at many things at once, and simply redrawing plots is too slow. The interaction seems critical to me in this problem.

Fox (2002) has implemented Velilla's method in R/S-PLUS, using almost the same numerical methodology and output that is in **Arc**; see also Weisberg (2005). The lack of interactive graphics make the methodology much less appealing, and will probably inhibit its widespread use.

5. Summary and remarks

If XLISP-STAT is so wonderful, why are we writing its obituary with these articles? I think there are several reasons, some that can be corrected, and some that, alas, probably cannot be corrected. The primary problem faced by XLISP-STAT is, and was, that it is not S. The native language, Lisp, is much harder to learn than is the S language. Even adding $2 + 2$ in XLISP-STAT requires Lisp programming. Students and others are not likely to invest the time to learn Lisp unless they see a payoff, and the vast majority of statistical computing problems that can be solved in XLISP-STAT can be done equally well in S-PLUS or R. I see the payoff with access to interactive graphics, but the importance of interactive graphics is not obvious to many in the statistical community. Indeed, for many practicing statisticians, all graphics are just part of the "wow" factor, and have no role in real statistics. John Tukey would be displeased by this trend.

Also, XLISP-STAT suffered from inattention. It did not evolve much after the early 1990s, some bugs were never fixed, and missing procedures were never added. For example, one type of graphical object is called a `name-list`, a simple list of labels. Like other graphical objects, a `name-list` object could respond to mouse clicks, but the double-click was never implemented. This made these objects much less useful, and methods written based on the `name-list` are less intuitive because the double-click could not be used. It also had several inherent weaknesses that were never addressed, such as few tools for handling text and categorical variables.

XLISP-STAT can be resurrected, or at least the parts of XLISP-STAT that cannot now be replicated in other programs or packages. For me, this would mean creating easy to use tools for building graphical objects, tools for interacting with them, and an object-oriented system that allows communication between objects. **Arc** hid all the features of XLISP-STAT that a novice found difficult. We need to be able to do this in other languages, too.

Arc is available from http://www.stat.umn.edu/arc. A complete, unchanged copy of the most recent version of XLISP-STAT is included in the Windows and Macintosh versions. The add-on to **Arc** that converts any graph drawn in **Arc** to LaTeX is described on the add-ons page on this website.

Acknowledgements

References

Becker RA, Cleveland WS, Shyu MJ (1996). "The Visual Design and Control of Trellis Display." *Journal of Computational and Graphical Statistics*, **5**, 123–155.

Box GEP, Cox DR (1964). "An Analysis of Transformations." *Journal of the Royal Statistical Society, Series B*, **26**, 211–246.

Cleveland W, McGill M (1988). *Dynamic Graphics for Statistics*. Chapman & Hall, New York.

Cook RD (1977). "Detection of Influential Observations in Linear Regression." *Technometrics*, **19**, 15–18.

Cook RD (1998). *Graphical Regression*. Wiley, New York.

Cook RD, Weisberg S (1994a). *An Introduction to Regression Graphics*. Wiley, New York.

Cook RD, Weisberg S (1994b). "Transforming a Response Variable for Linearity." *Biometrika*, **81**, 731–737.

Cook RD, Weisberg S (1999a). *Applied Regression Including Computing and Graphics*. Wiley, New York.

Cook RD, Weisberg S (1999b). "Graphs in Statistical Analysis: Is the Medium the Message?" *American Statistician*, **53**, 29–37.

Cook RD, Weisberg S (2004). "Partial one Dimensional Regression Models." *American Statistician*. Forthcoming.

Fox J (2002). *An R and S-PLUS Companion to Applied Regression*. Sage, Thousand Oaks.

Stuetzle W (1987). "Plot Windows." *Journal of the American Statistical Association*, **82**, 466–75.

Swayne DF, Cook D, Buja A (1998). "**XGobi**: Interactive Dynamic Data Visualization in the X Window System." *Journal of Computational and Graphical Statistics*, **7**, 113–130.

Tierney L (1990). *Lisp-Stat: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*. Wiley, New York.

Tukey JW (1980). "We need both Exploratory and Confirmatory." *American Statistician*, **34**, 23–25.

Velilla S (1993). "A Note on the Multivariate Box-Cox Transformation to Normality." *Statistics and Probability Letters*, **17**, 259–263.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer, New York, 4th edition.

Weisberg S (2005). *Applied Linear Regression*. Wiley, New York, 3rd edition.

Affiliation:

Sanford Weisberg
University of Minnesota
School of Statistics
313 Ford Hall
224 Church St. SE
Minneapolis, MN 55455
E-mail: sandy@stat.umn.edu