15

Implementation and Study of Statistical Strategy

R. Wayne Oldford
Stephen C. Peters

ABSTRACT

In this chapter we explore the possibility of using software like that developed by the AI community as a medium in which some of the statistical strategies useful in practice may be recorded and examined. Particular attention will be paid to discerning the kinds of strategies, and which of their properties, might reasonably be studied in this medium. Examples of the implementation of some of these ideas are given.

15.1 INTRODUCTION

Two seemingly disparate topics in statistics are presently generating much interest and activity. The first is concerned with the development of theory that better captures fundamental aspects of statistical practice than those theories currently available (Bayesian, Decision theory, Fiducial, Likelihood, Structural, etc.) (for example, see Mallows and Walley (1980)). This development seems related to the widely held view that to achieve effectiveness in statistical applications, formal training in theoretical statistics must be augmented by practical experience. A number of authors have begun research toward the development of theory (possibly non-mathematical) to help analyze and understand what it is that distinguishes good statistical problem solving (Finch, 1967, 1979; Cox, 1977, 1981; Tukey, 1982; Mallows and Walley, 1980; Good, 1980; Cox and Snell, 1981; Dempster, 1983; Mallows, 1983; and Mallows and Tukey, 1983). The second topic is concerned with statistical software. Programs and programming environments developed by the Artificial Intelligence (AI) community are being investigated with an eye toward their possible value in statistical software, principally to aid a statistically unsophisticated user to a better statistical analysis (Nelder, 1977; Campbell and Woodings, 1981; Chambers, 1981; Chambers et al., 1981; Hájek and Ivánek, 1982; Gale and Pregibon, 1982, 1984; Portier and Lai, 1983; Hand, 1984a, 1984b; Oldford and Peters, 1984; Pregibon and Gale, 1984; and Hahn, 1985).

Here, we explore the possibility of using software like that developed by the AI community as a medium in which some of the statistical strategies useful in practice may be recorded and examined. We focus on discerning the kinds of strategies, and which of their properties, might reasonably be studied in this medium. The software we are developing is therefore primarily directed at professional statisticians rather than statistically naive users. By paying careful attention to the organization of the statistical procedures and the user interface, we hope that the statistically unsophisticated user may be encouraged to perform a better analysis than he or she might have done otherwise.

In what follows we stress two major points. First, it seems to us that a reasonably accurate software representation will be possible only for certain kinds of statistical strategies, namely, those that would not require computer determination of the relevant properties of the substantive subject-matter context. For other kinds, only certain aspects of the strategy are likely to be implemented well in software. The first point will lead to the second, namely, that an implementable statistical strategy is best located *within* a statistical analysis system, as opposed to acting as an "expert" *interface* between the user and the system. This stands in contrast to, for example, the approach of Pregibon and Gale (1984) in REX.

The next section discusses some of the events that might occur in the course of a statistical analysis, defines what is meant by "statistical strategy" in this chapter, and finally identifies basic properties of any strategy. In Section 15.3, we identify the kinds of statistical strategies that are most likely to have complete software representations. Section 15.4 briefly describes a strategy for diagnosing collinearity in least-squares regression, an actual representation, and some of the implementation issues involved. Section 15.5 discusses implementation of some attributes of those statistical strategies unlikely to have complete representations. Section 15.6 briefly addresses some issues involved in evaluation, and the final section contains a summary of the main points and some concluding remarks.

15.2 THE PROBLEM

To use software as a medium within which statistical strategy may be represented, it is important to have a software environment where as much as possible of the statistician's task can be performed. It follows, then, that consideration should be given to the manner in which an analysis is actually done and to those elements of the analysis that can be committed to software.

Much of what a statistical consultant does involves asking questions of the subject-matter specialist in an effort to better understand the substantive problem being faced. In the process, it often happens that the relevant features, including inherent difficulties, of the problem are illuminated for the subject-matter specialist as well as for the statistician. We see no reason for optimism regarding the computer implementation of this part of the statistician's task (see Dreyfus (1979) and Weizenbaum (1976) for relevant discussion).

Once the substantive problem is understood by the statistician, much of the subsequent analysis can take place on a computer. Typically, this second phase of the analysis will involve an unordered exploration of a variety of different descriptions, approaches, and methods applicable to the problem. The data might be divided into subsets to be studied individually, possibly to get summary statistics to be used in a later analysis of the entire data set. Many different estimation procedures and mathematical models may be used on the same problem (including new procedures developed on the spot). Questions may arise such as "Why is the fourth observation so different?" and "What should be done about it?" These will require further interaction with the subject-matter specialist for resolution. Particular analyses can be either suspended temporarily or completely abandoned so that others can be taken up, each addressing different and illuminating aspects of the statistical problem. Each such sub-analysis may have been suggested by a previous sub-analysis, by something recalled from the subject-matter domain, by the interpretation of an exploratory graphical procedure, and so on (see also Thisted (1984) or Huber (1985)).

Even though much of the remaining statistical analysis can be carried out on a computer, many pivotal steps depend upon the ability to perform functions that are relatively easy for a human analyst and exceedingly difficult, if not intrinsically impossible, for a computer (cf. Gale and Pregibon (1984) or Gale (1985)). These include: the ability to interact repeatedly and effectively with the subject-matter specialist; the ability to do so for arbitrarily many different subject-matter domains; the ability to interpret the relevant properties of exploratory graphics where "relevant" may depend both on the particular graphic *and* on the reasons for using the graphic; and the ability to recognize unusual or unexpected circumstances including those *never before encountered* and hence those that may require rather ad hoc usage of currently available statistical procedures to understand.

The term "statistical strategy" will be used here to label the reasoning used by the experienced statistician in the course of the analysis of some aspect of a substantive statistical problem. The problem could be as narrowly focused as how to interpret a particular instance of a statistical graphic, or as diffuse as how to determine the relevant points to be addressed in some proposed subject-matter study. In this context, the word "problem" will refer to a particular instance, such as "Dr. A's heart attack study." Often in what follows, the plural "strategies" will be used for two reasons. First, two experienced statisticians may reason about the same problem in quite different ways and may therefore produce different analyses. Second, in the course of the overall analysis, as described above, most strategies will be relevant only at certain points in the analysis and be quite irrelevant at others (e.g., the reasoning used by the statistician in building a linear model will not likely be relevant at that point in an analysis where it becomes necessary to construct a density estimate). A statistical strategy, then, is a method used by the statistician to address some class of actual problems. The determination of both the method and the class of applicable problems is important.

If the goal is to build a software representation of some of the strategies used in statistical problem solving, then it is essential that the representation be *well targeted*, in the sense that its performance nearly matches that of the strategy used by the statistician on the same problem. What constitutes "nearly matching" will, of course, depend upon both the strategy and the aspects of the strategy one is interested in studying. A complementary requirement is that the representation be *complete*, in the sense that it is well targeted over the entire domain of problems for which the strategy is deemed to be applicable. Note that a complete representation of a strategy may be better targeted for some problems in the domain of applicability than for others.

For example, it is certainly possible to record an analysis in software after it has been performed by the statistician and to have this software representation exactly duplicate the statistician's strategy on the same problem. For that particular problem, the representation of the strategy is precisely targeted. If the class of problems for which the strategy is to be applied consists of the single observed problem (i.e., a genuinely ad hoc strategy), then the representation is complete as well. However, if the class of applicable problems is enlarged, then the representation may fail to be well targeted for the strategy on other problems in the domain. At that point the representation fails to be complete.

The nature of a statistical analysis suggests that, for many of the strategies involved, complete representations will not be reasonably achieved. The problem, then, is twofold: first, to determine which strategies of statistical practice could have complete software representations, and second, to determine which properties of the remaining strategies might usefully be committed to software. In the next section, we address the first of these issues. Section 15.5 deals with the second.

15.3 WHAT CAN BE DONE?

The argument of this section runs essentially as follows. First, if the strategy depends very little upon any intrinsically human ability, then it should be possible to build a well-targeted representation for it on some number of problems. Second, a strategy that always appears in small sub-analyses will typically have a relatively small class of problems for which it is applicable. Therefore, those strategies satisfying both properties seem to have the greatest potential for complete software representations.

To clarify this argument, consider the following two components of statistical strategy. The first of these is simply the level at which the strategy is expected to operate within the overall analysis. The second is the importance of input from the surrounding subject-matter context to the strategy's performance. Although the two are typically interrelated, they are not the same.

Figure 15.1 illustrates a partial ordering of some sub-analyses, and hence strategies, according to their operational level along the vertical axis.

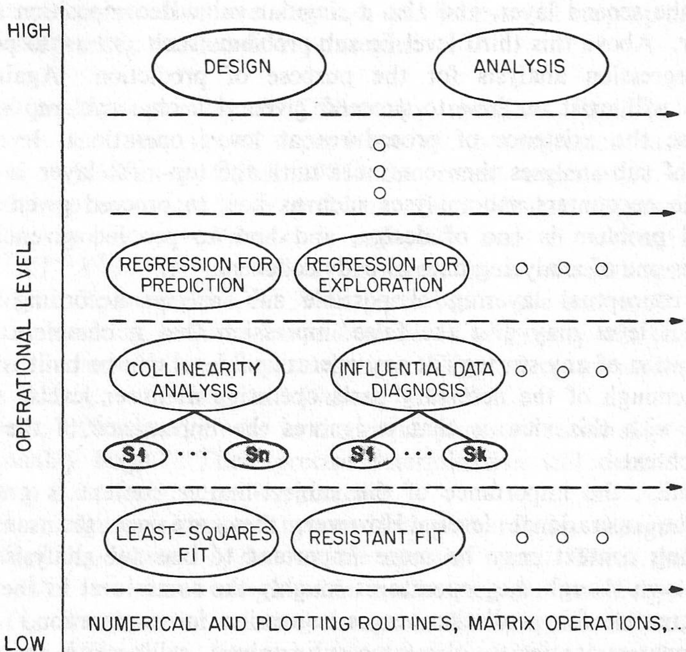

Figure 15.1 A partial ordering of strategies by operational level.

No ordering is implied horizontally. The ellipses indicate sub-analyses where a particular strategy might exist. Those shaded ellipses connected to "collinearity analysis" and "influential data diagnosis" are there to indicate actual strategies, the point being that for any of the larger ellipses many strategies might exist. (Indeed, part of the interest in studying strategies to discern which ones perform better, when, and why.)

In the bottom-most layer are numerical and plotting routines, and matrix operations which are usually considered to be primitives in a statistical analysis. The next layer above contains the standard fare of statistical procedures available in the literature. Shown here are a procedure that performs a least-squares fit on a linear regression model, and one that performs a resistant fit (e.g., Tukey's biweight) on a regression model. Each of these presume the existence of methods in the layer below in order to perform their tasks and hence operate at a higher level in the analysis. Coded in software, selections from these two layers constitute the minimal components of a statistical package.

Above this two-layer base are the relatively unstudied strategies which interest us here. Just above are such sub-problems as collinearity analysis and influential data diagnosis. Again, each of these presupposes the existence of certain procedures in the two levels below. For example, strategy S1 for a collinearity analysis may require variance inflation factors in a least-squares fit from the second layer, and also a singular value decomposition from the first layer. Above this third level lie sub-problems such as how to perform a proper regression analysis for the purpose of prediction. Again, many strategies will exist on how to proceed given this objective and each will presuppose the existence of procedures at lower operational levels. The layering of sub-analyses then continues until the top-most layer is reached. There one encounters sub-analyses such as how to proceed given that the statistical problem is one of design, and how to proceed given that the problem is one of analyzing data already collected.

This conceptual layering of possible sub-analyses according to their operational level may give the false impression that a complete software representation of any strategy at any operational level can be built, simply by building enough of the necessary parts operating at lower levels. A major difficulty with this view is that it ignores the importance of the subject-matter context.

Typically, the importance of the subject-matter context is greater the higher the operational level. However, they are not the same. The surrounding context may be more important to one sub-analysis than to another, even though they operate at roughly the same level in the analysis (e.g., regression for prediction versus regression for exploration.) Further, two different strategies for the same sub-analysis, collinearity, for example, may interpret the relevance of the subject-matter very differently.

The argument may now be restated with respect to these two components of strategy. First, sub-analyses and strategies should be selected that depend as little as possible on the surrounding subject-matter context. These offer the greatest promise for well targeted representations. Because of the general interrelationship between context and operational level, it would seem that the most likely candidates for computer implementation are those that operate at lower levels in the overall analysis. Second, the choice of these strategies has the favorable side-effect that the applicable domain of substantive statistical problems is also generally much smaller for those that operate at low-levels. Thus, low-level, low-context strategies seem to be obvious choices for complete software representations.

15.4 IMPLEMENTING LOW-LEVEL STRATEGIES

Since low-level strategies seem to be the most likely of any to have complete representations, we have spent some effort on the implementation of one low-level, low-context strategy for collinearity diagnosis. In this section, we detail some of that experience (see also Oldford and Peters (1984)). First the problem itself is described, then the strategy. Those points where the subject-matter context are important to the strategy's performance are identified. The representation is then briefly described, including reference to the AI software we found to be helpful. Finally some discussion is given to those general points which we observed, and believe will crop up again in the implementation of other low-level strategies.

The sub-problem considered: Collinearity manifests itself in regression analysis as the near linear dependence of the regressor variables (see Belsley et al. (1980) or Gunst (1983) for recent overviews). It will be regarded here as a sensitivity problem. In particular, suppose the usual regression model $y = X\beta + \epsilon$ is adopted and β is estimated by $b = (X^TX)^{-1}X^Ty$. Of interest is the effect on b of small perturbations added to X. To be precise, let X be perturbed to the matrix $X + \delta X$, and denote the resulting perturbation in b as δb. Collinearity will be said to be present if *reasonably small* perturbations δX produce *unreasonably large* changes δb (e.g., see Belsley and Oldford (1984)).

Clearly, difficulties here lie in the vague phrases "reasonably small" and "unreasonably large." Their precise determination will depend upon the application (i.e., the surrounding context).

Some ambiguity is removed by considering the relative changes $[\delta b]_2/[b]_2$ and $[\delta X]_2/[X]_2$, where $[\cdot]_2$ denotes the spectral norm. Then, the condition number $\kappa(X)$ serves as a rough guide in assessing the sensitivity. For fixed $[\delta X]_2/[X]_2$, the potential relative change in the estimator is an increasing function of $\kappa(X)$. Note, however, that reparameterizing from β to $A\beta$, for some non-singular A, will result in examining the condition number $\kappa(XA^{-1})$ instead of $\kappa(X)$. Assessment of the collinearity necessarily depends upon the parameterization or, equivalently, upon the choice of X and δX. Care must be taken in these selections to prevent the analysis from being rendered meaningless. It has been argued elsewhere (e.g., Belsley (1984), Belsley and Oldford (1984)) that the determination of the appropriate X and δX must come from the subject-matter context.

If it has been determined that a near dependency exists, the questions that naturally follow are: How many near dependencies are there? What variables might be involved? What aspects of the rest of the analysis are affected? What can be done to temper any bad effects?

The strategy: The strategy that we have been working on determines the number and strength of the near dependencies. This analysis is very low-level and hence a complete representation for the strategy involved seems promising. The strategy to be implemented is that employed by our colleague, David Belsley, who has had more experience with collinearity analyses than either of the present authors.

Given the regressor variables, the first step in the strategy is to obtain the appropriate X-matrix for the application. This done, the columns are scaled to unit (Belsley et al. (1980)). The condition indices $\eta_i = \lambda_1/\lambda_i$ for $i = 1,...,p$ of X are found, where $\lambda_1 \geqslant \cdots \geqslant \lambda_p \geqslant 0$, are its singular values $(\eta_p = \kappa(X))$.

Belsley's strategy for identifying the number and strength of any near dependencies amounts to examining the location and clustering of points, along the positive real-line, given by $Y_i = \log_{10}\eta_i$ for $i = 1, ..., p$. Experimental evidence suggests that moderate to strong dependencies are associated with values of Y_p around 1.5 and that increments of 0.5 on this scale are interpretable in terms of the multiple correlation coefficient of a single underlying dependency (see Belsley et al. (1980) pp. 128-131). For example, the X-matrix is very nearly rank deficient if $Y_p \geqslant 4.0 (\kappa(X) \geqslant 10{,}000)$.

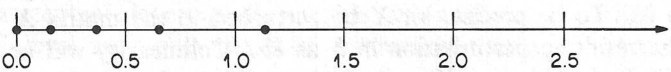

Figure 15.2 Log$_{10}$ condition indices.

Figure 15.2 shows an example ($p = 5$) of the sort of data the strategy interprets (no near dependencies here). The rest of the strategy was not easily articulated. Rather, it involved going over many plots like that in Figure 15.2 to determine what should be recommended in each case. Only in this way could the strategy be articulated clearly enough to be coded.

The representation: The first item of the strategy is the one most dependent upon the context and hence the most difficult to implement. Rather than build an elaborate rule-based procedure which would somehow elicit the appropriate X-matrix for the analysis, we chose to avoid the matter by forcing responsibility for the correct interpretation of the context onto the user. More than just an expedient, this decision is consistent with the idea of a professional statistician as the target-user (the professional statistician being more competent than software at eliciting such information from the subject-matter specialist). The representation assumes it is being handed the X-matrix which the user considers appropriate for the analysis. Even so, initial checks are made to circumvent obvious errors on the user's part. It also checks to see if the columns of the X-matrix are mean-centered. If they are mean-centered, then it prints a note about the importance of having "structurally interpretable" variables in a collinearity analysis (i.e., the appropriate X-matrix, see Belsley (1984)), asks the user if the analysis should proceed on the centered variables, and then does so if the user indicates yes. This part of the representation is coded in a rule-based language and therefore has a standard explanation facility which allows the user to query the representation about the questions being asked.

Given the appropriate X-matrix, the matrix is column-scaled and the Y_i's found. The largest of these is checked to detect whether a gross error has occurred, one indicating that the data are either so collinear, or so nearly orthogonal, that a detailed analysis is not worth the trouble (i.e., $Y_p \geqslant 4.0$, or $Y_p \leqslant 0.5$). Otherwise, the sorted Y_i's are run through a procedure designed to reproduce Belsley's interpretation of plots like that in Figure 15.2. This procedure (coded in LISP, not in the rule-base) divides the line into four segments or bins as shown in Figure 15.3 (two near dependencies here).

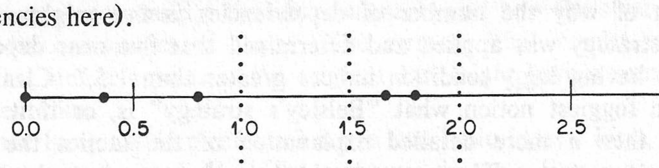

Figure 15.3 Dividing lines used by the representation.

Every Y_i appearing in the rightmost bin ($Y_i \geqslant 2.0$) definitely indicates a near dependency. These are the clear-cut cases. The reason for the existence of the other dividing lines is to make sense out of the remaining Y_i's. Based on the number and location of points in each of these bins, the set of anticipated configurations was factored into fifteen prototypical scenarios. Each gave different amounts and kinds of information about those near dependencies one needed to worry about, including, for two prototypical situations, the information that it was not certain about the number of near dependencies. In this case, an opinion about the number and strength of the near dependencies is given, but the user is invited to look at the plot and change the decision of the procedure. These situations are a consequence of the difficulty involved in numerically interpreting subtle features of a plot.

General points: First, a note on the software with which we are currently experimenting. The current incarnation is written in InterLisp-D (code is available upon request). Previous ones were written in a rule-based language based on Van Melle's (1979) EMYCIN. We found that the simple rule-based languages we have used are not flexible enough to deal with the requirements of our collinearity representation (Oldford and Peters (1984)). In the current incarnation, we are mixing rule-based programming somewhat freely with other methods in an expert-systems building package called LOOPS, which itself is written in InterLisp-D (see Stefik et al. (1983)).

Given that strategies are being implemented that may never have been formally articulated, it is imperative that the reasoning underlying the strategy be made available to the user. Unfortunately, the explanation facility of rule-based procedures is not yet rich enough to allow the user to ask questions like "What do you mean 'structurally interpretable'?" We are planning to make canned text and examples available to the user upon request by means other than the explanation facility, perhaps as a menu choice. This would be at least a partial solution to a problem we have noted elsewhere (Oldford and Peters (1984)), namely, the need for multiple levels of explanation. Any particular strategical procedure may sometimes be perceived as tactical by the user, and other times as strictly mechanical. Consider the above procedure which determines the number of dependencies. If we regard it as a mechanical way of determining the number of dependencies (i.e., we are somewhat familiar with the underlying logic and are not too concerned with how it operates it detail), then a satisfactory explanation of why the number of dependencies is five might simply be "Belsley's strategy was applied and determined that five near dependencies existed, all having $\log_{10}$ condition indices greater than 1.5." Clearly if we haven't the foggiest notion what "Belsley's strategy" is, or if it is at all suspicious, then a more detailed explanation of the tactics the strategy employs is necessary. The user should be able to select the detail of explanation required. (Ellman (1985) considers a knowledge representation scheme which addresses some of these issues.)

Note also that for some of the prototypical situations it seemed best to go to the user with the option of changing the inference made by the representation. We anticipate that this will often be the case with strategies that are based on interpreting graphical procedures (our graphic was only unidimensional!). Those places in the representation, where an inference from the user may be required, must be identified.

Finally, such low-level strategies will typically play rather small roles in the overall statistical analysis so that, for example, a "collinearity expert interface" between the user and some statistical system will be of little use. Of more interest would be a representation that is as easy to access and use as the common statistical procedures, that is, one that is part of the statistical system.

15.5 IMPLEMENTING HIGHER LEVEL STRATEGIES

Complete representations will be very difficult, if not impossible, for strategies that either operate at high levels or depend critically upon the substantive context of the problem. That said, it may still be possible to represent certain aspects of these strategies in a useful way.

A mental image of the aspects to be captured may be evoked by considering the following scene. An inexperienced statistician knows that some subject-matter specialist will shortly require advice on how to perform a regression analysis. What the substantive problem is about, why a regression analysis is to be performed, or even what subject-matter area is involved are all completely unknown. The inexperienced statistician, having never actually performed a regression analysis before, decides to telephone a far more experienced statistician to be briefed on the best way to proceed, what to watch for, and so on. The strategy of interest, then, is the one given over the telephone by the experienced statistician.

To a large extent, the resulting strategy will consist of a simple cataloguing and grouping of known statistical techniques. With this organization in place, some suggestions will be given for each technique regarding its proper usage. Since the experienced statistician has no access to information on the substantive problem, these suggestions must remain context-free, consisting only of hints and examples.

For strategies that are not likely to have complete representations, the idea, then, is to consider the above scenario and how the strategy of interest could be communicated there. The strategy that results, although only reflecting simple aspects of the original one, might have a complete software representation.

Underlying the implementation are two basic ideas. First, the statistical procedures to be used should be organized in the software according to this resulting strategy. Second, the interpretation of the substantive context should be left entirely to the user; only context-free hints and examples should be given by the representation.

As yet, we have not gained enough experience to recommend the most appropriate user interface. Our work is preliminary and still at an experimental stage. Nevertheless, an example should indicate the effect these two ideas have on the implementations we are considering.

Suppose that a user has already declared that a multiple linear regression is to be performed with the variate y as the left-hand-side variable and the variables $x_1, x_2, \ldots, x_p$ as possible regressors. The user might then be presented with a menu of items like that in Figure 15.4.

Fit	>
Inspect x-data	>
Inspect y-data	>
Plot	
Suggestions	

Figure 15.4 First menu after regression invoked.

Associated with each item is some brief documentation. For example, the documentation associated with the Suggestions item would read "Produces a list of suggestions which are often useful at this stage." Selecting the Suggestions item would produce canned text like "When performing a regression analysis, it is generally a good idea to inspect the variables involved before fitting the model." This text would of course be much lengthier, perhaps describing what to look for when inspecting the data and how the analysis might be affected by the findings. The important thing to notice is that the suggestions would not be tailored to the actual problem faced by the user (i.e., Dr. A's heart attack study). Instead, any context that might be important to making a relevant suggestion could be described in the suggestion itself, as in "If the problem here is similar to the following one ..., then it is usually a good idea to do such and so." In this way, the user is obliged to make the decisions about interpreting the context. Selecting 'Plot' would call up some general plotting facility, allowing the user to make plots at will. Selecting 'Inspect y-data' might produce a stem and leaf plot of the y-data, whereas selecting 'Inspect x-data' might produce a stem and leaf plot for each regressor, and a scatter plot-matrix (see Chambers et al. (1983)) for all regressors and y. A least-squares procedure is invoked if 'Fit' is selected.

Those items that have a > to their right would have another sub-menu associated with them. This sub-menu would appear only if the user did not want to invoke the default methods assigned to the original menu item (like a least-squares fit). For example, the 'Fit' sub-menu might be that shown in Figure 15.5. Three items indicate that there are further sub-menus attached to them.

Least-squares	
Robust-estimation	>
Bounded-influence	>
Ridge-regression	>
Plot	
Suggestions	

Figure 15.5 Sub-menu of Fit selection.

'Suggestions' here might indicate that the influence on the overall fit of observations outlying in x- or y-space could be reduced somewhat by using one of the bounded-influence estimators, and so on.

Selecting 'Least-squares' would perform the fit and produce the usual statistics. In addition, another menu could be produced which contained, among other things, the item 'Collinearity diagnosis.' When selected it would offer access to the usual statistics used in a collinearity analysis, but more importantly it would also offer something called 'Belsley's strategy.' This would contain a rule-set which would invoke the various procedures involved in the representation of Belsley's strategy described in the previous section. Along these lines other implementable low-level strategies could be attached where they make the most sense.

Although the preceding description does not represent an existing working implementation, it does illustrate an organization of somewhat standard statistical software according to their operational level. In this way, the user is given some idea of when certain procedures are often used, so that this aspect of the statistical strategies involved will be captured. Further, by the use of suggestions and other documentation (including examples), the user may get some idea as to which methods are applicable to the problem at hand. Thus, those places where the context is important and, perhaps, those aspects of the context that are most relevant are made explicit.

The software we plan to produce will be of interest to the professional statistician (both for study and use) and to the less sophisticated user who is willing to follow his or her nose through the software. Of course, no real attempt has been made to prevent disaster (cf. Chambers (1981)), rather, the hope is that the user will be persuaded to give more careful and focused thought to the analysis, thereby producing a better one.

15.6 EVALUATION

How should our software be evaluated? It must be evaluated on how well it meets its stated goals and purposes. The purpose we have laid out in previous sections is to build complete representations of statistical strategies in order to study the strategies they represent. Therefore, there are two natural focal points for evaluation. The first and most obvious is the evaluation of the representation. Is it complete? If so, then we can get on to the more interesting focal point, namely, that of evaluating the strategy itself.

Unfortunately, completeness of the representation does not seem to be easily verified, since it would almost always require the examination of every problem in the applicable domain of the strategy. The exception would be a strategy for which the problems in the domain all fall into one of a finite number of subclasses such that taking one problem from each subclass would suffice to test the representation. Incompleteness, on the other hand, can be demonstrated with a single problem in the domain. If the representation is not complete, then further investigation is required to determine whether the necessary correction to the representation is reasonable or symptomatic of a more serious difficulty.

It is the nature of the software we are developing that the representation undergo continual evaluation. The representation is applied to problems in the strategy's domain and its performance compared to that of the strategy used by the statistician. If the representation for the strategy is not well targeted at a particular problem, it is improved to accommodate it. If this incremental improvement is nothing more than a patch, there is a real danger that the representation will again fail to be well targeted at another problem in the domain that differs only slightly from the test problem. Therefore, careful examination of the representation's performance on each test problem is required to determine whether fundamental rethinking of the representation is required. By contrast, with Student it is envisioned that a more general program could be built which would make incremental improvements to a representation as required (Gale and Pregibon (1984) and Gale (1985)).

Once a strategy has been formalized by a complete representation, it may be evaluated. This will necessarily require detailed documentation of the strategy (on both the method and the domain of applicability), most of which should be part of the software itself. Descriptions of where in the domain the representation is best targeted, where it is worst, and what is meant by "well targeted" for this strategy are also required. Such information should be supplemented with the actual case studies examined. Here, just as with mathematical and verbal formalizations, peer review seems essential. In addition to the usual difficulties associated with the peer review process, a few more accompany the evaluation of the strategy: the standards to be met are not as clearly established; any "failing" of the strategy may be attributable to the quality of the representation (e.g., the reviewer's idea of, "well targeted" may be quite different from the author's); the reviewer would ideally have access to the software environment in which the strategy is encoded — something not easily accomplished at the present time. A reasonable first step would be to build a complete representation for two strategies that claim to have the same domain of applicability. Then, any difference in performance between the two could be studied to determine whether the differences are important or inconsequential.

15.7 SUMMARY AND CONCLUDING REMARKS

We began by pointing out two different themes in current statistical research: first, the development of statistical theory, which better reflects fundamental aspects of statistical practice; and second, the development of "intelligent" statistical software. The claim made here is that the two go naturally together. That is, an understanding of what constitutes good statistical practice is necessary for the development of significantly more "intelligent" statistical software, and the tools currently available to address the software issue could be helpful in the development of statistical theory.

The idea, then, is to extract and record in software those "strategies" that practicing statisticians employ in their analyses. If this can be done at all successfully, then the recorded strategies may provide input for the development of theory closer to actual practice. Minimally, the process should help clarify the strategies themselves.

To this end, we have identified what we feel to be important features of statistical strategy. First is the distinction between the method and the domain of applicability. Next are two components that affect both the method and the domain, namely, the identification of the importance of the substantive context to the performance of the strategy and the identification of the level at which the strategy operates within the overall analysis.

Also identified is that property of a representation which requires careful examination and continual evaluation, namely, its completeness for the strategy. If the representation is an incomplete one, then those problems in the domain for which it is not well targeted must also be identified.

Based on this analysis, we have proposed the kind of strategies we feel are most likely to have complete representations. They are typically low-operational-level, low-context strategies that have well-defined and relatively small domains of applicability. Some kinds of higher level strategies with larger domains of applicability have also been identified, for which certain aspects of the strategy may be reasonably represented. The approaches we propose adopt the principle that interpretation of the substantive context of the problem should be the responsibility of the user. In this way, the user is recognized as accountable for the analysis.

These considerations on statistical strategy have led to the conclusion that the software representations of statistical strategies are best located within a statistical analysis system. Low-level strategies will not be of general enough interest to warrant implementation as an expert interface. Similarly, the aspects of high-level strategies that can be reasonably implemented amount to the grouping together of existing statistical routines. This suggests a different structure for the statistical system, rather different from an expert system placed on top of an existing package.

Finally, we would like to close by briefly addressing what is probably the most obvious question about the entire enterprise, namely, "Why software?" There are many answers to this question. The two most important seem to be, first, that for any strategy committed to a software representation, its author is forced to be precise about the strategy, its domain of applicability, and its limitations; and second, that by offering representations of strategies within a statistical system, the strategies can actually be used. In this way, good strategies may see more use, poor ones will be recognized, and new ones will be created.

Acknowledgments

This research is part of an ongoing project at CCREMS, MIT, partially funded by the National Science Foundation grant IST-8218759. Thanks are due to our colleagues in this project, D. A. Belsley, E. Kuh, A. Samarov, and R. E. Welsch.

REFERENCES

Belsley, D. A. (1984) "Demeaning conditioning diagnostics through centering" (with discussion) *The American Statistician 88*, 73-77.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980) *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, New York: John Wiley and Sons.

Belsley, D. A., and Oldford, R. W. (1984) "On the general problem of ill-conditioning in statistical analysis." *Proceedings of the A.S.A.: Business and Economic Statistics Section,* 431-436.

Campbell, N. A., and Woodings, T. L. (1981) "Improved diagnostic output from statistical packages" (with discussion) *Proceedings of the 43rd Session of the I.S.I. XLIX* (Buenos Aires), 279-293.

Chambers, J. M. (1981) "Some thoughts on expert software." *Proceedings of the 13th Symposium on the Interface*, 36-40.

Chambers, J. M., Pregibon, D., and Zayas, E. (1981) "Expert software for data analysis: An initial experiment" (with discussion) *Proceedings of the 43rd Session of the I.S.I. XLIX*, (Buenos Aires) 294-303.

Chambers, J. M., Cleveland, W. S., Kleiner, and Tukey, P. A. (1983) *Graphical Methods for Data Analysis*, Scituate, Mass.: Duxbury Press.

Cox, D. R. (1977) "The teaching of the strategy of statistics" (with discussion). *Bulletin of the International Statistical Institute, 47, Book 1*, 553-558.

Cox, D. R. (1981) "Theory and general principle in statistics" (with discussion). *Journal of the Royal Statistical Society, Series A, 144*, 289-297.

Cox, D. R., and Snell, E. J. (1981), *Applied Statistics: Principles and Examples*. London: Chapman and Hall.

Dempster, A. P. (1983) "Purposes and limitations of data analysis." In *Scientific Interface, Data Analysis, and Robustness*, G. E. P. Box, T. Leonard, and C-F. Wu, (eds.). New York: Academic Press, 117-133.

Dreyfus, H. L. (1979) *What Computers Can't Do: the Limits of Artificial Intelligence, Revised Edition.* New York: Harper and Row Publishers.

Ellman, T. (1985) "Representation of statistical computations: Toward expert systems with a deeper understanding of statistics." In this volume.

Finch, P. D. (1976) "The poverty of statisticism" (with discussion). In *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science, Volume II,* Harper and Hooker, (eds.), Dordrecht, Holland: D. Reidel Publishing Company, 1-46.

Finch, P. D. (1979) "Description and analogy in the practice of statistics." *Biometrika, 66*, 195-208.

Gale, W. A. (1985) "Student phase 1 – A report on work in progress." In this volume.

Gale, W. A., and Pregibon, D. (1982) "An expert system for regression analysis." *Proceedings of the 14th Symposium on the Interface,* 110-117.

Gale, W. A., and Pregibon, D. (1984) "Constructing an expert system for data analysis by working examples." *COMPSTAT-84*, (Prague), 227-236.

Good, I. J. (1980) "The philosophy of exploratory datum analysis." *Proceedings of the A.S.A.: Business and Economic Statistics Section*, 1-7.

Gunst, R. F. (1983) "Regression analysis with multicollinear predictor variables: Definition, detection, and effects." *Communications in Statistics – Theory and Methods, 12*, 2217-2260.

Hájek, P., and Ivánek, J. (1982) "Artificial intelligence and data analysis." *COMPSTAT-82*, (Vienna), 54-60.

Hand, D. (1984a) "Statistical expert systems I: Design." *The Statistician* (to appear).

Hand, D. (1984b) "Statistical expert systems II: Necessary attributes." *Journal of Applied Statistics* (to appear).

Hahn, G. J. (1985) "More intelligent statistical software and statistical expert systems: Future directions" (with discussion). *The American Statistician, 39*, 1-6.

Huber, P. J. (1985) "Environments for supporting statistical strategy." In this volume.

Mallows, C. L. (1983) "Data description." In *Scientific Inference, Data Analysis, and Robustness*, G. E. P. Box, T. Leonard, and C-F. Wu, (eds.). New York: Academic Press, 135-151.

Mallows, C. L., and Tukey, J. W. (1983) "An overview of techniques of data analysis, emphasizing its exploratory aspects." In *Some Recent Advances in Statistics*, J. Tiago del Oliveira and B. Epstein, (eds.). London: Academic Press, 111-172.

Mallows, C. L., and Walley, P. (1980) "A theory of data analysis?" *Proceedings of the A.S.A.: Business and Economic Statistics Section*, 8-14.

Nelder, J. A. (1977) "Intelligent programs, the next stage in statistical computing." In *Recent Developments in Statistics*, J. R. Barra et al., (eds.). North-Holland Publishing Co., 79-108.

Oldford, R. W., and Peters, S. C. (1984), "Building a statistical knowledge-based system with Mini-MYCIN." *Proceedings of the A.S.A.: Statistical Computing Section*, 85-90.

Portier, K. M., and Lai, P-Y (1983) "A statistical expert system for analysis determination." *Proceedings of the A.S.A.: Statistical Computing Section*, 309-311.

Pregibon D., and Gale, W. A. (1984), "REX: An expert system for regression analysis." *COMPSTAT-84*, (Prague), 242-248.

Stefik, M., Bobrow, D. G., Mittal, S., and Conway, L. (1983) "Knowledge programming in LOOPS: Report on an experimental course." *The AI Magazine, 3*, 3-13.

Thisted, R. A. (1984) "Computing environments for data analysis." *Dept. of Statistics, U. of Chicago Technical Report Number 166.*

Tukey, J. W. (1982) "Another look at the future." *Proceedings of the 14th Symposium on the Interface*, 2-8.

Tukey, J. W. (1982b) "Introduction to styles of data analysis techniques." In *Modern Data Analysis*, R. L. Launer and A. F. Siegel, (eds.) New York: Academic Press, 1-11.

Van Melle, W. (1979) "A domain-independent production-rule system for consultation programs." *IJCAL*, 923-925.

Weizenbaum, J. (1976) *Computer Power and Human Reason.* San Francisco: W. H. Freeman and Company.