AN EXPERT SYSTEM INTERFACE

Julie A. Elston, Boeing Computer Services
David J. Purdon, Boeing Computer Services

ABSTRACT

In general, statistical software requires more analytical sophistication on the part of its users than most software. Users must be familiar with the syntax of the package, knowledgeable about statistical theory, and be able to converse with a variety of systems in order to use the software optimally. This places a tremendous burden on the amateur and occasional statistical software user.

One answer to this common problem is to have a user-friendly front-end system to help users in being more productive with statistical software. This implies an interface which could access data from a variety of sources and run desired procedures. But more importantly, it also implies the ability to assist the user in applying basic statistical theory to the data analysis process.

Expert system methodology presents a uniquely effective approach in developing such a system. We have built a demonstration prototype called THESA (The Expert Statistical Assistant). The *expert* in the title refers to the branch of artificial intelligence known as expert systems. The THESA concept is based on using external routines or packages, like SAS [1], to perform the statistical computations. The THESA prototype demonstrates that an expert system interface imbedded with theory can be used to improve statistical research methodology for nonstatisticians.

INTRODUCTION

Many new users of PC statistical software are unpleasantly surprised to find that it requires a fair amount of knowledge about statistical theory to use effectively. For many of these users this problem can no doubt be traced to the popular myth that any software that runs on a PC is easy to use.

Clearly this would be true if the only difficulty in using the software was based on its mainframe qualities. Unfortunately, some software may be difficult to use because it requires expertise outside the domain of the average computer user. In fact statistical software falls into this category.

One of the inherent difficulties in using this software lies in the fact that the computational procedures available in statistical software do not in themselves derive conclusions from the data. The user is still expected to be thoroughly familiar with the problems in the data, (and indeed what constitutes a problem) and be able to correct them.

Often the user will also need to strategize the sequence of statistical tests to collect information to conclude anything meaningful about the data.

In theory this process may appear straightforward, but in practice users frequently leave out critical steps which can unwittingly lead them into drawing inaccurate conclusions about their data.

For example, a common problem new statistical software users encounter is proper development of the multiple regression model. Often users neglect looking for problems in the data and decide they have a good model when R-squared values are high and the F statistic looks significant. However, these values can be convincing in the presence of a strong interrelationship between explanatory variables.

This problem, more commonly known as multicollinearity, means that some of the explanatory variables may be surrogates for others and one of the damaging results is that parameter estimates may not be accurate. This type of problem can usually be detected by checking to see if standard errors are unusually large or by reviewing the correlation matrix. Once the problem has been diagnosed, it can usually be handled in a number of ways so that an effective regression model can be developed. The point is that statistical software does not warn the user if these problems exist nor does it tell the user how to fix them. It is the user's responsibility to know how to diagnose problems and affect appropriate remedies.

In addition to proper application of statistical theory, effective use of statistical software also requires users to be familiar with the procedures, syntax rules, and the different systems used to access data and program the desired analysis. These are a lot of demands for the novice or occasional statistical software user.

These demands can be met by a front end system for statistical software which we will call THESA (for The Expert Statistical Assistant). We feel that artificial intelligence programming techniques can be used to build a system that will be capable of assisting any user in accessing data and guiding the user through a basic analysis of the data.

In this paper we will describe some related work in this area, explain what THESA should do to assist the user, and discuss why an expert system approach makes sense for building THESA. We will then discuss how artificial intelligence programming techniques can be used to build THESA and describe a demonstration system of the THESA concept that was built. The final section will identify the benefits that can be realized by such a system.

WHAT OTHER WORK IS BEING DONE IN THE AREA?

We found that problems with statistical software

usage have been recognized [2] and that other systems had been built to deal with the effective use of statistical software [3]. These systems also confirmed our belief that an expert system could be used to bridge the gap in statistical skills and provide some insight into this common problem.

The system which seemed most similar to the one we wanted to build was REX [4], the Regression Expert system built by William Gale and Daryl Pregibon at Bell Labs. Like REX our system was attempting to improve the quality of regression analysis applications, the working tool of so many disciplines. Unlike REX, we wanted our system to emphasize exploratory data analysis procedures, as well as extend the error checking capabilities to include multiple regression analysis. This approach was intended to reduce the chance for errors in the overall data analysis process as well as more precisely serve the needs of our users.

WHAT SHOULD THESA DO?

In deciding what THESA should do we came up with the following wish list. First of all, the system should do as much of the drudgery work of data analysis as possible. This includes accessing the data from various systems, building desired subsets of the data, and writing the code to invoke the desired statistical procedures. The system should also help the user get *a feel* for the data by producing informative plots and basic descriptive statistics of the data set.

It should be able to test for violation of linear model assumptions and make suggestions to help the user handle the problems if the violations exist. In some instances the system would of course have to warn the user that the advice of a professional statistician should be sought before proceeding with an analysis.

The main objective was to build a system that would help users make more informative decisions regarding the data analysis process and help them implement their decisions more easily.

In general, it should not only assist the user in examining the data in a methodical fashion but teach the user what to look for in the data and why.

WHY SHOULD THESA BE BUILT?

Although the general difficulty in using statistical software merits the development of such a system, we were actually called on to investigate the feasibility of building such a system in response to a statistical computing support requirement for our marketing division. Both the consulting statisticians and the market analysts expressed an interest in using such a system to decrease the dependency on the statisticians for very basic statistical consultations. The system could also be used to assist the statisticians in getting a quick overview of the data and allow additional time for more critical types of research that we decided THESA should not attempt, such as experimental design.

Another factor which made an expert system front-end a natural choice was the type of work performed. After examining the work load in the marketing division we determined that much of the statistical work could be termed routine in nature. The division was frequently calling on statisticians for information on means, totals, variances and plotted values of various subsets of data. Further, it seemed the market analysts had had some experience in developing linear models of the data and analyzing the output but were unable to generate the analysis themselves. All of these factors indicated that an expert system could be used to bridge the gap in statistical computing skills.

HOW ARE EXPERT SYSTEM TECHNIQUES USED?

Conventional programming techniques, while being well suited to doing statistical computations, would be a poor approach to building an interface like THESA. The design of even complex statistical computations can be completed prior to any code construction. However, it would be very difficult to completely describe the design of THESA on paper even when restricting its application to a particular domain, such as marketing research. This is because the knowledge required for such a system comes from a variety of sources, including both written text and the personal experience of key individuals. It would use basic statistical theory and have the capability to access statistical packages. It would also require statistical application experience and an understanding of the system users. It's the kind of information that you don't think about until you need it. [5] The way to develop such a system is to first write code that solves a very small problem and then incrementally add to the code as you use the system to improve its performance. Continually rewriting conventional software in this manner would be very costly.

An expert system is designed to be coded a little bit at a time. At each stage of coding users can run the system to evaluate its strengths and deficiencies. The approach for THESA is more precisely called a rule based system or a production system. [6] The term production system, used in this context, refers to a type of computer language that is fundamentally different from FORTRAN or PASCAL. The organization of production system software provides a better model of the task which THESA should perform. A production system has three basic components: production memory, working memory and the inference engine. Production memory contains rules (also called productions) which tell the system what to do and when to do it. Working memory contains packets of information called working memory elements which describe the state of the world.

Specifically, production memory is a set of *condition-action* rules. (Rules were historically referred to as productions, and hence the name.) The *condition* part of each rule describes when the rule would be appropriate. The *action* part of each

rule describes what should be done if the *condition* part is satisfied. It may modify working memory (changing the state of the world) or call external routines to perform computations or ask the user for opinions. For example, the following rule, written in a particular production system language called OPS5 [7], says that the rule is appropriate when a simple linear regression model has been built and the corresponding action is to assume that no serial correlation exists in that data.

```
(p assume-no-serial-correlation
  (data-model ↑ name <model>
              ↑ instance simple-linear-regression)
  -->
  (make claim ↑ instance serial-correlation
              ↑ property-exists no
              ↑ made-from assumption
              ↑ made-by $<model>)
```

The creation of a new working memory element may cause another rule (dealing with testing assumptions) to be appropriate and its *action* will be performed. The production system will continue in this manner until there are no more appropriate rules.

The inference engine software, which is part of the language, continually looks (in an efficient manner) for rules in production memory that are appropriate for the current state of working memory and performs their actions. The process is continually repeated as long as working memory is being changed to allow new rules to be appropriate.

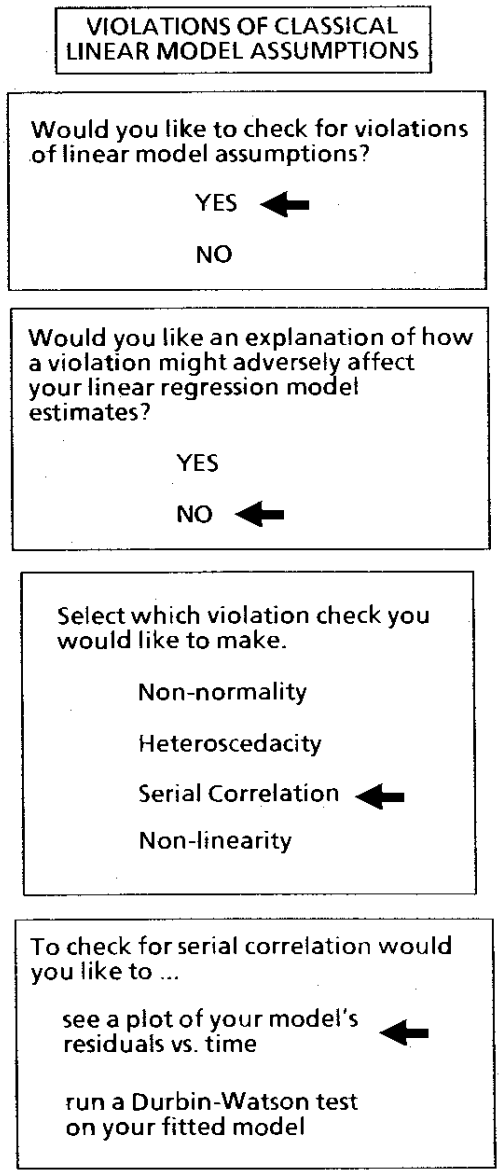
The programmer writes rules for production memory which describe the individual steps for THESA to take. After rules which make assumptions for a linear regression model are written, the programmer would then add rules that look for assumptions and try to test them. If during the system testing stage it was discovered that an important test was not suggested, then the programmer would add some rules to perform the test. The programmer does not have to be concerned with placing the new rules in a specific location because the inference engine will search all of production memory to find appropriate rules. This special architecture would allow THESA to be built incrementally and thus enable a complicated problem to be solved a little bit at a time.

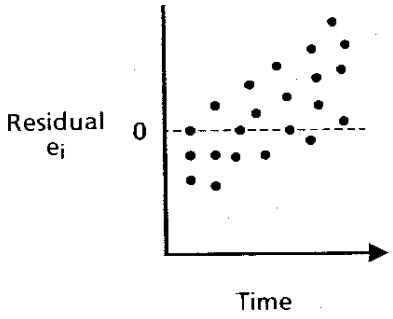
A SAMPLE SESSION

A small demonstration system was written using Knowledge Craft [8] software on the Explorer [9] computer workstation. The Common Lisp environment of the Explorer was used to perform the small number of statistical computations required by the system.

Expanding the demonstration system into a complete prototype would require use of existing statistical software on another machine that was linked to the Explorer. Currently the demonstration system is able to build a simple linear regression model, test that model for some violations of assumptions, and make suggestions on how to build a more reliable model to fit the data.

To get a feel for the interface, let's take a look at an idealized sample session where a user is depicted using THESA to test for violations of classical linear assumption regarding particular regression analysis. In this scenario, the system does not automatically perform the analysis but offers the user both computational and visual methods for detecting problems with a linear model.





To continue to check for serial correlation would you like to ...

see a plot of your model's residuals vs. time

run a Durbin-Watson test on your fitted model ←

Dubin-Watson Statistic: **.3033**

There is evidence to suggest that your data has a positive autocorrelation problem.

Would you like to see residual vs. time plots which indicate a serial correlation problem for comparison with your plot?

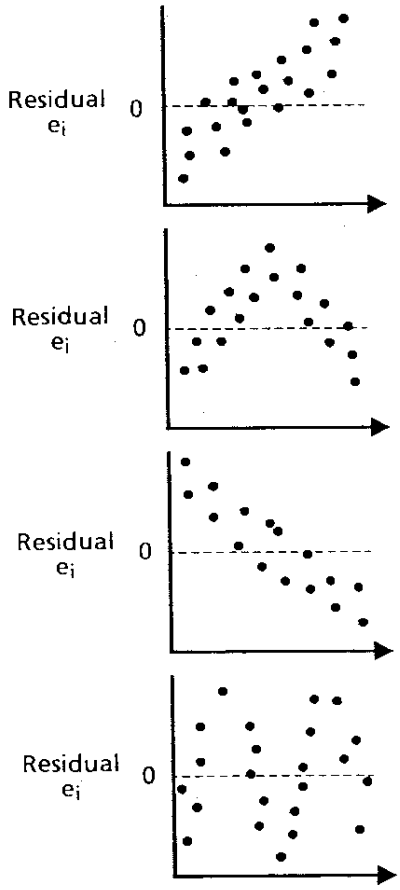
YES ←

NO

Would you like to perform some remedial measures to try and remove the problem?

YES ←

NO



Which transformation method would you like to try?

first-differencing method ←

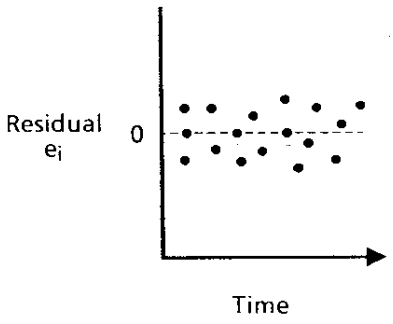
other

TRANSFORMATION COMPLETE

Would you like to verify that it was effective in removing your autocorrelation problem?

YES ←

NO



Dubin-Watson Statistic: **1.9201**

There is no evidence to suggest that your data has an autocorrelation problem.

BENEFITS

Direct monetary savings would result from letting analysts use a complete THESA system to obtain descriptive statistics and allowing statisticians to focus on more complex problems. The indirect benefits would also be significant. Among these benefits are:

- Better utilization of the existing work-force
- Better utilization of expensive data sets
- Greater consistency in statistical methodology and documentation
- Increased productivity from a wider application of statistical data analysis
- Removal of the requirement of software familiarity

CONCLUSIONS

The effective use of statistical software requires application experience. This knowledge is usually acquired from written text or the personal experience of key individuals. Incorporating some of this knowledge into the software would provide a uniform method of distributing the information to a diverse user community. Expert system programming techniques provide a good approach to developing that software.

ACKNOWLEDGEMENTS

The authors would like to thank Norm Black, Jim Flexer and Kathy Crowell of Boeing Computer Services for their assistance and support.

REFERENCES

1. a registered trademark of SAS Institute Inc., Cary, NC.

2. Chambers, J. M., "Some Thoughts on Expert Software," Proceedings of the 13th Symposium on the Interface of Computer Science and Statistics, New York: Springer-Verlag, 1981, pp. 36-40.

3. Gale, W. A. and Pregibon, D., "An Expert System for Regression Analysis," Proceedings of the 14th Symposium of the Interface, New York: Springer-Verlag, 1982, pp. 110-117.

4. Gale, W. A., "REX Review," Artificial Intelligence and Statistics, Gale, W. A. editor, Addison-Wesley, 1986, pp. 173-227.

5. Tukey, John W., "Sunset Salvo", The American Statistician, February 1986, Vol. 40, No. 1, pp. 72-76.

6. Nilsson, Nils, "Principles of Artificial Intelligence", Palo Alto, CA: Tioga, 1980.

7. Brownston, L., Farrel, R., Kant, E., and Martin, N., "Programming Expert Systems in OPS5," Reading, MA: Addison-Wesley, 1985.

8. a registered trademark of Carnegie Group Inc., Pittsburgh, Pennsylvania

9. a registered trademark of Texas Instruments Inc., Dallas, Texas.

AUTHOR'S ADDRESS AND PHONE

Julie A. Elston
Boeing Computer Services
P. O. Box 24346 MS 6R-59
Seattle, WA 98124
(office) 206-656-7213
(home) 206-324-2260

David J. Purdon
Boeing Computer Services
P. O. Box 24346 MS 6R-08
Seattle, WA 98124
(office) 206-656-7264
(home) 206-852-0255