F. Faulbaum, R. Haux und K.-H. Jöckel (Hrsg.) (1990). SoftStat '89
Fortschritte der Statistik-Software 2. Stuttgart: Gustav Fischer, 291 - 299

Is there Something New that Can Be Learned or Solved by Building Statistical Expert Systems?

P. Naeve

Summary: Several statistical subjects truely benefit in one way or the othere from all those efforts put into the field of statistical expert systems. Firstly there is a new and very fruitful discussion of the old question "What is statistics?" Secondly we do not only understand better what could be meant by intelligent statistical software but can take the first steps to implement this better understanding. Thirdly we became aware how strongly we depend in our daily work as statisticians on graphical methods of all kinds. These points are worked out and are illustrated by examples.

A wise man's advice. Let me start with a very general answer and - what seems to me more impotant - a warning. I will do this by quoting John Nelder:

> "...We have much to learn about this new kind of software which should not be thought of as a panacea. Procedures unhelpful to the scientists and technologists can be codified as rules just as easily as helpful ones, and we can confidently predict that the misuse of packages will be followed by the misconstruction of expert systems ..." (Nelder, 1986)

Learning by doing is a well-known advice. So I am glad my talk is based on experiences I made during a larger research project in the field of statistical expert systems. I will discuss three questions the answers of which might benefit in one way or the other from our efforts on building statistical expert systems. These questions are

> What is statistics?
> What is intelligent statistical software?
> What are we doing with graphics?

There is but one way to tell the truth many people do not want to hear. Say it again and again. So before going into any details let me present my main message again. All those fruitful impulses I will talk about that statistics might get by our working on statistical expert systems might be outnumbered by all evils if we cannot convince the statistical layman that inspite of our working on statistical expert systems there will never be something like a true statistical expert system on the market.

It might be helpful to introduce briefly that research project I mentioned at the beginning.

The (my) background. In the year 1987 the University of Bielefeld and the Nixdorf Computer Company established a joined research project on statistical expert systems. The goal was – at least from Nixdorf's point of view – 'building a statistical expert system with the help of existing, commercially available software'. This will say take a statistical package (the choice was P–STAT because those P–STAT people promised to be cooperative) and a shell (the choice was TWAICE because it is a Nixdorf product). TWAICE belongs to the class of shells constructed according to the EMYCIN paradigm – although the Nixdorf people did not like this classification very much. From the university's point of view we liked to express the goal of the research project somewhat more in line with a scientific attitude. 'Is it possible to build an expert system relying on existing software components?'

It was agreed upon that we should not take special AI–hardware. So the work was done on the university's main frame (a Hitachi computer) and on a HP 9000/320 computer. The operating systems were UTS and UNIX respectively.

Can one give a definition for statistical expert system? Having used the term statistical expert system several times it seems to be mandatory to present a definition. In a somewhat technical way what we were doing was to take an expert system shell and a statistical package as building blocks and to add statistical expertise. So the problem of having a definition for a statistical expert system is changed to 'What is statistical expertise?' Here are the types of expertise we wanted to incorporate into our system.

1. Statistical expertise stands for the ability to find models for the description and analysis of problems. To be sucessful a common background made out of statistics and subject matters must be established among the statistician and client.

2. Statistical expertise stands for methodological knowledge. The statistician knows about the prerequisites, the algorithms, data input and output etc. of statistical procedures. He not only knows how to apply a special procedure but also when to apply this procedure. This includes the knowledge how the interpretation of results should be changed when some prerequisites are not fulfilled.

3. Statistical expertise stands for the ability to describe data. This means the statistician has a concept of an abstract data model from which he deduces the specific model suited for the problem at hand. (The situation resembles somewhat the relation between scheme and subscheme in a data base.)

4. Statistical expertise always incorporates teaching expertise. Hopefully not only a problem is solved during a consultation but the client knows more about statistics then before.

5. Statistical expertise nowadays also stands for statistical software expertise. A software package is not only a collection of useful statistical procedures but usually incorporates some model building strategies. Unfortunately this point is not always made very explicit. And here is the place too to remind us that far to often you have to quite a lot of computer science knowledge to get some statistical package work on your problem.

This is quite an impressive list. But we are convinced that you cannot delete anything from this list without giving up building an expert system in statistics.
Take this list as necessary attributes of a statistical expert system.
If one would insist on having a proper definition I would like to join the one given by John Nelder (Nelder, 1984) in his paper on statistical computing. But I think what has been said so far suffices to make my position clear enough with respect to the question: What is a statistical expert system? So let me deal with the first question.
What is statistics? This question has a long tradition. For instance again and again it was theme of the presidential address in the Royal Statistical Society. For a long time the main problem seemed to be a clear distinction between statistics and mathematical statistics. Nowadays the question sounds like: What are statistical strategies? This shift in focus is due to John Tukey's stressing the importance of explorative statistical analysis.
Speaking about or analysing statistical strategies cannot be done without a suitable language. AI offers many language components which can help to describe knowledge and inference processes involved in statistical strategies.
Two recent papers by Thomas Westerhoff and the author present this point in more details than can be given here (Westerhoff, 1988a,b) But be careful!
Abstraction is not reduction and vice versa. We all know we need language to express our thoughts. And the novel 1984 is a fine example what happens to the people's minds if one only allows for such poor language as 'new speech'. And what about programming languages of all kind? They are still so close to 'new speech' that one faces a danger which I will call reductionism. Let me give you an example.
Recently I found this quote whilst reviewing a paper:

> "Wir wollen eine Maschine ... genau dann als Expertensystem bezeichnen, wenn sie Funktionen, die für einen Experten spezifisch sind, übernehmen kann. ... Unter (spezifischen) Funktionen ... verstehen wir diejenigen Fähigkeiten, die, außer von Expertensystemen, nur von Experten übernommen werden können. ..."

This seemed to be a worked out version of the definition for 'intelligent' given by the Merriam Webster (Webster, 1983).

> intelligent: able to perform functions of a computer.

In our work we had to face such problems too – especially when you consider that our partner had strong commercial interests. There was always the temptation to abandon what could not be modeled within the possibilities of the shell.
The so called emptiness of the shell was one of the greatest source for this danger of reductionism. A shell is not empty. It has an inference engine which has strong – but not always revealed – views on how your inference processes take place. Those two papers by Thomas Westerhoff und myself deal in some length with this kind of problems.
Having read through many papers reporting on 'building expert systems' I got the impression that more than one author was trapped in this reduction sink.
Reductionism is not just a possibility it is sad fact.
What is intelligent statistical software? There is a long plea for intelligent statistical software. But it is not always clear what is meant by intelligent.
One interpretation seems to be

i) The software should be able to prevent the user from doing something silly e.g. calculate the mean for 0,1 coded sex.

Here AI tools for knowledge representation like frames and rules certainly offer new possibilities. Another interpretation could be

ii) The software should not hinder the statistician doing his analysis the way he wants – it should behave like a slave to his master obeying commands and adapting to new ideas.

Here I doubt if expert system technology really brings something new – at the contrary sometimes it seems as if one goes backward. One of the essential features of expert systems is the ability to explain.

> Why this question?
> How did we reach this point?
> etc.

But when linking the shell TWAICE to P–STAT we soon noticed that this amounts to the shell has to have control over whole dialogue with the user – how should it otherwise understand what was going on. So we had to fall back into the stone age of batch processing with P–STAT.
Although one could pile a lot of critical remarks about software packages their standards are much higher than promoter of expert system technology want us to believe. To mention just on point: interactivity!
Other drawbacks when using shells like lacking interfaces, doing calculations in PROLOG, will be passed just by mentioning. But even within their own domain shells do not have the kind of intelligent software components one should expect. Here are two examples.
How smart are shells with respect to context? This is a very short example showing what has to be done to get something like a 'natural' conversation.
The problem is just this. For a statistician the term variable is something familiar. Variables come along with the problem but not always in the appropriate form at the beginning. Sometimes they have to be transformed. The statistician will speak of using a transformed variable – later on when he is working on the problem for the second time it is just a variable to him. But let us concentrate on the first time. Here is the dialog:

```
(4) >pH
  Wie lautet der Name der 2. Variable des 1. Modells ?
  ("-", falls keine weitere normale Variable betrachtet werden soll)
(5) >Ca
  Wie lautet der Name der 3. Variable des 1. Modells ?
  ("-", falls keine weitere normale Variable betrachtet werden soll)
(6) >-
  Wie lautet der Name der 1. transformierten Variable im 1. Modell ?
(7) >pHtrans

SETUP sind in diesem Modell bisher folgende Variablen bekannt, die
bei Erzeugung neuer Variablen durch Transformation verwendet werden
duerfen :
Ca
pH
```

and here is what has to be done to get something like a context dependent conversation.

```
RULE  1221
IF   Variable . ist_transformierte = nein
THEN Variable . Name = PROC ( prc_frage_normal )
END

RULE  1222
IF   Variable . ist_transformierte = ja
THEN Variable . Name = PROC ( prc_frage_trans )
END

proc(prc_frage_normal,'Variable',INST,'Name',[],[(_,0)]):-
        dyn_fact(_,'Variable',INST,'Nummer',[(NR,1000)]),
        dyn_son('Variable',INST,'Modell',M_INST),
        dyn_fact(_,'Modell',M_INST,'RELORD',X),
        tw_write(dialog_out,['Wie lautet der Name der ',NR,'. Variable',
        'des ',M_INST,'. Modells ?',nl(1)]),!,
        make_number(NR,Numb),
        Numb > 1,
        tw_write(dialog_out,['("-", falls keine weitere normale Variable',
        'betrachtet werden soll)',nl(1)]),!,fail.

proc(prc_frage_trans,'Variable',INST,'Name',[],[(_,0)]):-
        dyn_fact(_,'Variable',INST,'Nummer',[(NR,1000)]),
        dyn_son('Variable',INST,'Modell',M_INST),
        dyn_fact(_,'Modell',M_INST,'RELORD',X),
        tw_write(dialog_out,['Wie lautet der Name der ',NR,
        '. transformierten Variable im ',M_INST,'. Modell ?',nl(1)]),!,
        make_number(NR,Numb),
        Numb > 1,
        tw_write(dialog_out,['("-", falls keine weitere transformierte',
        'Variable betrachtet werden soll)',nl(1)]),!,fail.
```

Much work just to get the simple phrase 'transformed' at the right place.
Are we really always consistent? This is an example of how the state of the art is lacking behind of what is needed. It is of a more technical nature to make the point of deficiency more explicit.

```
        Bei der von Ihnen gewaehlten Analyse und dem angestrebten
        Ziel schlagen wir als Verfahren "Boxplot" vor.

(18) Sind Sie mit dem vorgeschlagenen Verfahren einverstanden ?
>nein

Bitte waehlen Sie nur aus den folgenden Verfahren aus :
Boxplot
Letter-Values
Stem-and-Leaf

(19) Welches Verfahren waehlen Sie aus ?
>'Letter-Values'
```

The figure exhibits a short part from a dialog SETUP (the name of our 'system') had with the user. It seems quite natural that the user and the system discuss what method should be applied. Obviously we have the possible chains of conversation

system suggests a method	—	user accepts
and		
system suggests a method	—	user disagrees
system offers a list of methods	—	user makes his choice.

Due to lack of space I cannot present the code explicitely as I did above. To give you a feeling we had to implement eleven rules (each has approximately the size of those two shown above) and several special objects and attributes to get this little piece of 'natural' dialog. The reason for this is the fact that TWAICE (like all other shells at that time) was very anxious to be consistent, will say: have a consistent knowledge base all the time. Therefore you were not allowed to change an attribute value once it was set. This was one way to stick to TWAICE's design principle: The knowledge base has to be consistent at any moment.
I think especially when doing exploratory statistical analysis we are usually working on top of a large stock of compiled knowledge which is not consistent at all. Hopefully at the end of our work we can come up with a 'consistent piece of work' – just by discarding all parts that would destroy consistency. And expert systems in statistics should allow and support such kind of thinking and working.
What are we doing with graphics? If we had not been aware of the fact that graphics are indispensable tools for statisticians the project would have taught us. I think it is no offend to Roald Buhler when stating that P-STAT is not to famous for its graphical capabilities. But compared to the possibilities allowed for by the shell P-STAT looks like a giant. I will deal with this point in a moment.
First let me briefly summarize how we as statistician work with graphics.

i) We use graphics to present results. A picture is worth thousand words. The many systems for presentation graphics on the market stress this import application of graphics. (In passing I want to say that in my opinion the quality of these products is often more than questionable.) This usage of graphics causes no great problems with respect to statistical expert systems but presentation graphics are of no great value if one considers expert system applications.

ii) We argue with the help of graphics. 'I see linearity in this scatter plot, therefore ...'. 'This point seems to be an outlier ...'. When acting in this way we are looking for something in the displayed data and want to share our observation with our partner, client or supervisor. What are we looking for? The answer is either the expected (the confirmative view on the displayed data) or the unexpected (the explorative view on the displayed data). Especially

the later point is now in the focus of the statistical community due to Tukey's EDA (explorative data analysis). Here is a small example how we play around with graphics. The figures are from a forthcoming Ph.D. theses on 'Literate EDA' by P. Wolf (Wolf, 1989).

The well-known data represent measurements of brain-weight (in g) and body-weight (in kg) of different brands of animals (figure 1). Unfortunately I can only show very few of those pictures one would use to tackle the following questions:

1. Are there outliers? This is a two dimensional problem. The scatterplot of the original data suggests the answer yes. But look at the scatterplot of the log-transformed data. Is man an outlier?

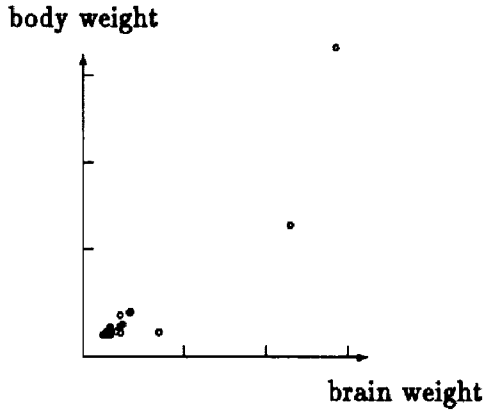

Figure 1

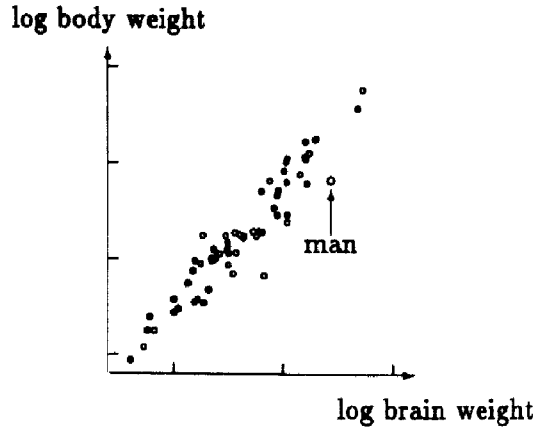

Figure 2

2. Let us turn to a one dimensional problem. Are the log-brain-weight data symmetric? Here is a series of boxplots that might be helpful to get an answer.

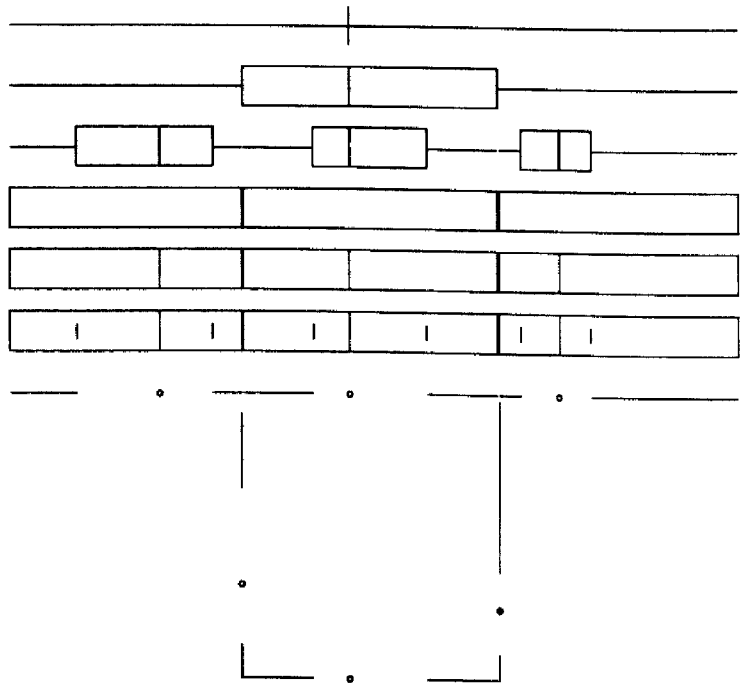

Figure 3

3. Another one dimensional problem is posed by asking what about empty spaces. Here are two plots that might be of use.

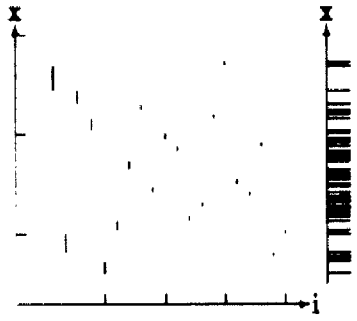

Figure 4

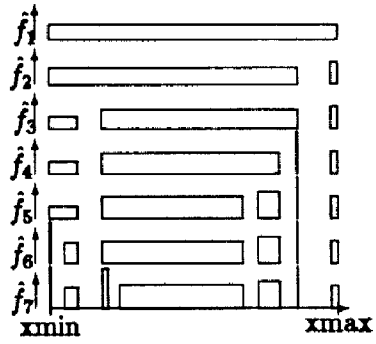

Figure 5

4. Are there isolated values? This picture incorporates many ideas of EDA.

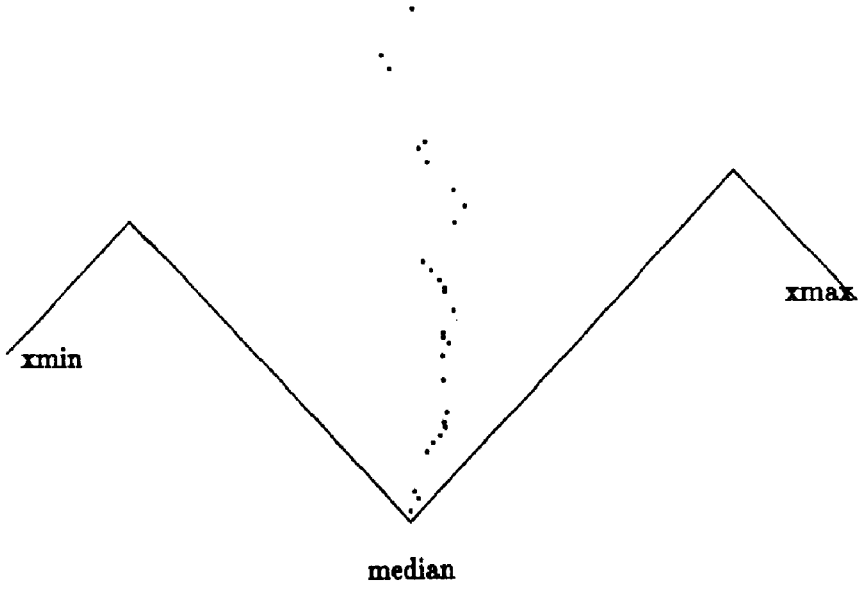

Figure 6

How could such a playing around be done when one of the partners is an expert system? Here is how we managed the problem to let SETUP see what P-STAT offered as graphical display to the user. We had to provide a numerical analogon for every tupel of problem and graphic so that SETUP could compare the clients answer to question such as 'Does this picture exhibits symmetry' etc. The following lines exhibits the structure behind SETUP's ability to look at a picture. "lastresult" contains the picture the user looks at. "internresult" is needed so that SETUP can pretend to look at the same picture.

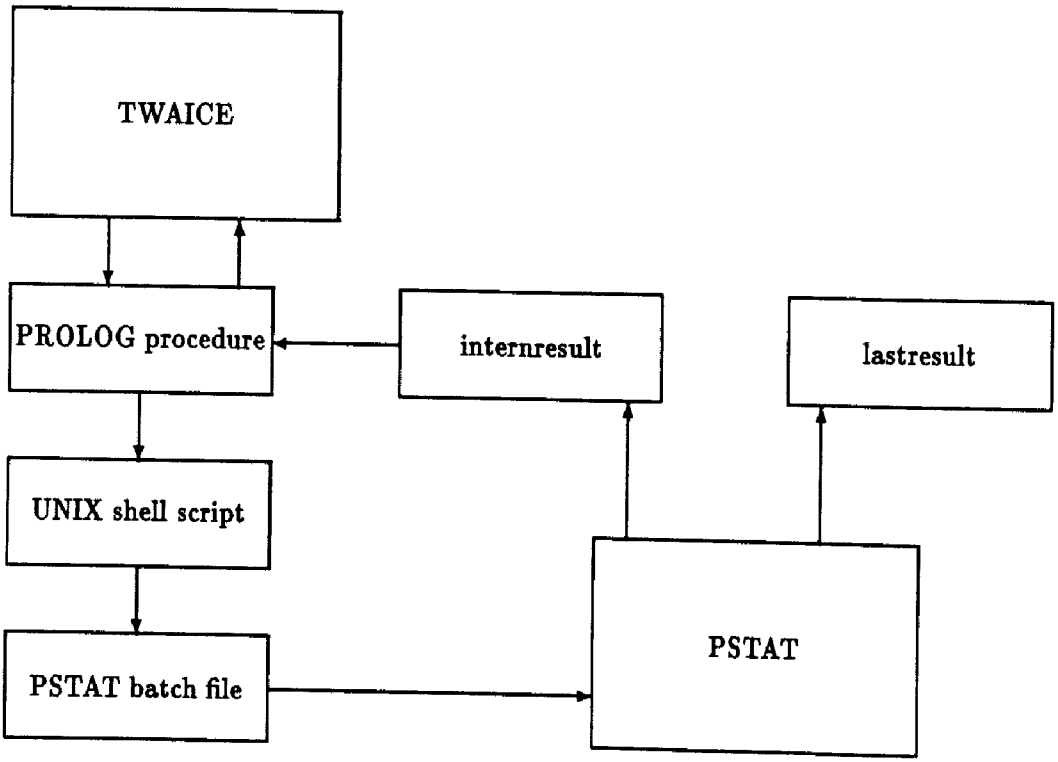

Figure 7: Structure of SETUP's look feature

This is a typical content of the P–STAT batch file (see figure 7).

```
echo "BATCH\$
MAXERROR 1\$
FIND $1, LIBRARY $2\$
PR lastresult\$
PRINT.PARAMETERS lastresult, OUTPUT.WIDTH 80, NO ECHO\$
EDA $1;
BOX.PLOT $3;
\$
PR internresult\$
PRINT.PARAMETERS internresult, OUTPUT.WIDTH 80, NO ECHO\$
EDA $1;
LET $3;
\$
PR\$
END\$" > pstatin
```

This is not a very promising way to do it on a large scale. But it is a good proof for Tukey's claim that we need more cognostics. (P–STAT knows a lot of facts, numerical values etc. about the boxplot shown but there is no possibilty for the user to lay hand on those.)

Where should we go next? I do not want to discuss all kind of repair work to exiting software packages – front-ends might be seen just as such. Neither do I want to deal with small pieces of programs which concentrate on very limited goals. These can be very useful to understand what can be done with expert system technology. But sometimes one cannot help thinking the old saying publish or perish has got a modern version program or perish.

What I want to stress instead is the need for something genuine for statistical work. As I put it elsewhere (Naeve, 1986a). "To make explicit that we are looking for more let us coin a new

name for it. I think statistical environment would be the right name ... – it must contain tools to enlarge the language for personal or common usage."
Our discussion makes us add to the features to be included in a statistical environment

— rules
— frames
— inference strategies i.e. forwared chaining, backward chaining.

Is a statistical environment a dream or could it be reality? I think S (Becker, 1988) is a good prototype of a statistical environment (for all who know my bias towards APL let me say this is a candidate too). So what I intend to do is to enlarge S by adding some AI tools. To be more precise we want to incorporate some frame like structure and mini inference engines.
I had in mind to give some first results. But before you can do this you have to inplement S – the new S.
We did not succeed in finishing this task as fast as we planned.
I do not want to discuss at length why we failed. But you may guess some of the reasons when you hear me stressing: We – the programmers – have not succeeded in writing and documenting correct software. When implementing S we found examples of almost every sin one is warned for in books and lectures on programming and software engineering. So be aware: a statistical expert system is from one point of view just a program. Be humble!
The answer is yes and no I hope to have shown that many things can be learned when working in the field of statistical expert systems. If you do it the right (humble) way you can learn a lot about statistical strategies, statistical software and statistical graphics. And you get a bunch of new and interesting problems. You are on the right way if you are aware of the ever existing danger of reductionism. If you fail to stay on the right way you will increase the large crowd of fooling – most of them do not intend to do but nevertheless do – tailors of the emperor claiming to offer new clothes where there is just nothing. For the message is as stated at the beginning of my talk: ...there will be never anything like a true statistical expert system on the market.

References

Becker R.A., Chambers J.M., Wilks A.R.(1988). The new S language. Pacific Grove
Naeve P.(1986a). Programming languages and opportunities they offer to the statistical community, in: Proceedings of the 1st World Congress of the Bernoulli Society. Y.Y. Prohorov, V.V. Sazonov (eds.), Tashkent
Naeve P. e.a.(1986b). How SETUP was set up. Diskussionspapier 183/1986, Fakultät für Wirtschaftswissenschaften, Universität Bielefeld
Nelder J.A.(1986). Statistics, Science and Technology. JRSS Series A 149, p 109f.
Nelder J.A.(1984). Present position and potential development: Some personal views. Statistical Computing, JRSS Series A 147, p 151f.
Webster's Ninth New Collegiate Dictionary (1983). Merriam Werbster, Springfield
Westerhoff Th., Naeve P.(1988a). On Inference processes. Proceedings in Computational Statistics. COMPSTAT , D. Edwards, N.E. Raun (eds.), Heidelberg, p 193f.
Westerhoff Th., Naeve P.(1988b). Vom menschlichen Inferenzprozeß zum maschinellen Inferenzprozeß am Beispiel statistischer Inferenz und TWAICE-Generationen. Proceedings 2. Anwenderforum Expertensysteme, A.B. Cremers, W. Geisselhardt (eds.), Duisburg , p 307f.
Wolf P.(1989). Grundprobleme der EDA. Literate EDA als Antwort auf Kommunikationsprobleme einer explorativen Datenanalyse. Ph.D. Thesis, Fakultät für Wirtschaftswissenschaften, Universität Bielefeld, to appear