Robust and Exploratory Analyses Automated

by

Gerry Hobbs and
E. James Hamer
West Virginia University

1. Introduction

The Interactive Data Analysis System (IDAS) is a system of macros developed within the SAS™ environment to calculate selected statistical quantities. The interface with the user is friendly and forgiving of user errors. All required information for an analysis is requested and, if appropriate, a list of possible responses is provided. Input is checked for validity whenever possible. Error messages are printed if improper entries are made. The user is given an opportunity to change or correct input after each entry, even if it is correctly verified. The system also remembers all entries and thus the user only needs to enter changes. These features, coupled with the ability to quit the system at any time, are the principal design items of interest to the user.

Certain principles underlie the programming design. Individual macros typically perform a single task in order to achieve modularity. The object is to isolate the code to the extent possible in order to allow easy maintenance and extension. These last two activities are related since extending a system often increases the required maintenance.

The current implementation contains two statistical macros. The first computes the odds ratio and associated statistics for matched case-control studies with a variable number of cases and controls per strata. The second calculates quantities obtained from binomial experiments. These are viewed as the beginning of a "complete" modeling environment. The design of IDAS will allow other statistical macros, such as logistic regression modeling and various exploratory methods, to be integrated easily.

2. The Main Control and Utility Macros

IDAS can be conceptualized as a library consisting of a main control macro, utility macros, and statistical macros. The first two are discussed in this section and the last is discussed in a later section. The main control macro is named IDAS. It consists of general messages for the user, comments for the programmer, and the main control loop. The loop allows the user to select the appropriate statistical macros which currently are PSI (for estimating the odds ratio) and BIN (for estimating and testing binomial parameters). A number of other macros are in various stages of development, including robust modelling and graphical exploratory modules.

All entries by the user are read by a macro named GET. GET must determine if the entry is new or if previously entered information is to be used. It also calls a macro named ENTER, which asks the user to verify the entry.

Four macros are used to fetch and verify the contents of a SAS data set. These are GETDATA, CHKSET, CHKNAME, and VARLIST. GETDATA fetches the SAS data set name from the user (with GET) and initiates the checking macros. It also prints the variable names in the SAS data set up to the 132 byte limit of the buffer. CHKSET checks whether or not the named SAS data set exists either in a permanent SAS data set library or in the temporary work library. It calls CHKNAME which determines whether or not the SAS data set name (one or two level) specified by the user is valid. VARLIST, the last macro, creates macro variables which contain lists of all variables, the numeric variables, and the character variables in the selected SAS data set.

Three macros are used for obtaining and verifying variable names, which subsequently will be used in the analysis. These are GETCLAS, GETVAL, and WORDS. GETCLAS is used to get a single variable or list of variables. Contrary to the name of the macro, the variable(s) may or may not be classification variables although the current statistical macros only use categorical (binary) variables. GETCLAS checks that the variables are in the SAS data set and assigns them to a type depending on whether they are character or numeric. WORDS counts the number of entries in an input string, e.g. the number of variables in a variable list.

GETVAL obtains the values of a classification variable which meets the conditions of the calling statistical macro. The values may be character or numeric depending on the type of the parent variable.

The next three macros, VALUE, CHKVAL, and CHKNRG, read and check a constant entry. VALUE reads the entry using GET. CHKVAL determines if it is a valid number. Finally CHKRNG checks whether or not the constant is in the appropriate range.

The last utility macro is CHKRES. It establishes whether or not any of the SAS data set or variable names are in a reserved list for the statistical macro. The user is requested to rename a SAS name if it conflicts with a reserved word.

These utility macros provide the interface with the user and a front end for appending statistical macros.

3. The Statistical Macros

The statistical macros perform basic analyses in case-control and case-series studies. The macro PSI computes the odds ratio and other statistics for a matched case-control design and allows both the number of cases and controls to vary from strata to strata. The macro calls the utility routines to ask the user for information about the SAS data set and variables which will be used in the analysis. First the name
of the SAS data set is obtained using GETDATA. Then the variable identifying the matched sets, the strata variable, is found using GETDATA. It is necessary for the SAS data set to be sorted on this variable before the macro is invoked. The variable itself can be either character or numeric.

Next the variables labeling the cases and controls and the exposure are specified by the user as asked by CETCLAS. These variables can be either character or numeric but must be binary. Therefore, the macro queries the analyst for the case and control values and the exposure and non-exposure values using GETVAL. Any of these four values can actually be a list of values which combine the actual categories of the variable into the defined class values. For example, the character values B and C both represent control categories of the variable into the defined class values. For values for the personal characteristics data.

The user must also enter the confidence level using VALUE. This latter macro invokes the macro that checks if the entry is numeric and between 0 and 1. The user is then given the option of entering identifying text for the analysis.

Once the statistical quantities are computed, the output is usually sent to both the screen and the SAS output file. The details of this process are explained later in this paper. The output summarizes the important information relating the case/control to the exposure variable. The actual output received depends on the pattern of missing values.

The estimate of the odds ratio, PSI, is always given even if it is "undefined." The details of the computational methods are given in the FACE Protocol (Seal, I., (2), (d), 1986) and in Schlesselman (1982). If PSI > 0, the log odds ratio, the estimated standard error of the log odds ratio, and a confidence interval based on Hauck's method are printed. The chi-square test value is given if its denominator is positive. If PSI and the chi-square value are both positive, the test-based confidence interval is also given. These are the basic summaries (Schlesselman, 1982) utilizing the matched data.

Summaries (without significance levels) are also printed which ignore the matching. The estimated proportions of exposed cases and controls and their estimated standard errors are given next. In order to assess the magnitude of the "non-response," the numbers and percentages of missing cases and controls are also summarized. The final summary gives the counts for all possible matching patterns. This is determined from the number of case and control values which the user has already supplied. For example, if the design calls for 1 case and 2 controls, the output will give the numbers and percentages of 0 cases with 1 and 2 controls and of 1 case with 0, 1, and 2 controls.

The statistical macro BIN computes various summaries about a binomial parameter. The SAS data set is determined as in the previous macro. Only one variable, the response variable, is required for this analysis, however. This variable must be binary and thus the user is prompted for both the event value(s) and the non-event value(s).

The final entries are to obtain the confidence level and the hypothesized null value. Both must be between 0 and 1. A message can also be entered to identify the output.

The output consists of the point estimate and the estimated standard error of the probability of the event denoted by P. Next the confidence interval based on the normal approximation is given. The test statistic, also based on the normal approximation, and the P-value are then printed. Finally, the number and percentage of missing and non-missing values are summarized.

4. User's Reference

IDAS is extremely simple to use and is tolerant of user errors. This section describes the basic commands for running the software.

IDAS is designed to be used in SAS's interactive mode. It is possible to operate IDAS under the Display Manager, but the process is somewhat awkward. Assume the code for the macros is in a regular 80 column fixed format CMS file on your A-disk with a filename of IDAS and a filetype of SAS. The steps for operating IDAS enumerated here are given for IBM's VM operating system.

The user must first invoke SAS from CMS by the command:

SAS (options)

This will initiate SAS's interactive mode. Several options are generally requested. NODMS specifies interactive line mode rather than the Display Manager. The log and the output files are sent to your terminal by default. No other SAS command options are necessary if this is all you want.

Usually a permanent record of the output (and possibly the log) is desired. These files can be routed to CMS files and to your screen. For example, the command

SAS (NODMS PTYPE PDISK NAME IDAS)

would send the log to the screen (by default) and the output to both the screen and a CMS file called IDAS LISTING A.

This is often the choice selected. The default filetype is LISTING for SAS output, but this can be changed by the user (as well as the filename) using the NAME option. Omitting PTYPE would cause the output to be sent to the CMS file but not the terminal.

The log file can also be sent to an external CMS file with the command

SAS (NODMS LTYPE LDLISK NAME IDAS)

This sends the log to the screen and to a CMS file named IDAS SASLOG A. The filetype is SASLOG by default and the filename is arbitrary. Both of these can be specified on the NAME option. LDLISK by itself (without the NAME option) results in a CMS file SAS SASLOG A.

Using LTYPE and LDLISK is not recommended, however, since the macros are expanded and written to the screen. Thus, thousands of log lines are sent to the screen (and the disk file). A better solution is to use the CP command

CP SPOOL CONSOLE TO * START
which sends all console (screen) text to your virtual reader. It can then be sent to a printer or file. When spooling is complete the following command must be issued:

```
CP SPOOL CONSOLE CLOSE STOP
```

SAS system options are also specified on an OPTIONS statement in IDAS. These are DQUOTE which recognizes double quoting, a requirement for proper operation of the macros. MAUTOSOURCE insures that the autocall libraries will be invoked. NONOTES suppresses notes on the SAS log. Both the line size and the page size are set to 80. Other OPTIONS, e.g. NODATE, could be specified on either the OPTIONS statement or on the SAS command.

Once within the SAS line-mode environment, the macro code must be compiled for your session. This is done by the SAS statement

```
%INCLUDE IDAS;
```

The filetype SAS is assumed. IDAS is then invoked by:

```
%IDAS
```

At this point you are under the control of IDAS until QUIT is issued. The first prompt will ask for a procedure, currently either PSI or BIN. Your response will invoke the proper statistical macro which then asks for the information described in Section 3.

Be aware that much time can be saved by using the enter key if the information has been entered previously and nothing has changed. Also, responding with QUIT at any time will cause IDAS to terminate and return control to SAS.

Any entry must be verified by the user before it is permanently associated with the appropriate macro variable. If the entry is correct, the respondent can answer by YES, OK, or by the enter key. If not, the user can respond NO or by anything else and the prompt will ask again for the input.

Sample outputs for the log and output SAS files are given in Appendix B. User responses are given in bold.

5. Extensibility

IDAS was designed to be easily extended by adding other statistical macros. The descriptions of the macros in Sections 2 and 3 should serve as a guide for this process. The code is given in Appendix A. The comments should aid the identification of program segments and the names of the major macro variables.

The process of adding macros is simple, but SAS's macro language must be known before an attempt is made to extend this system. SAS's macro language does not have many of the features available in other languages, e.g., complex data structures and sophisticated debugging facilities. Thus the language cannot be used to parse an expression and debugging can be time consuming. The best way to debug code is to run IDAS under the Display Manager and to use SAS command options such as MPRINT. A command for this purpose would be

```
SAS (MPRINT)
```

Diagnostic printing is also a common method of debugging. For example, the summary count information for each strata can be obtained by placing the print statement

```
PUT A= B= C= C= N=;
```

just prior to the RETURN statement in the block of code beginning with the label REPEAT. This information can be used to verify the calculations in the macro PSI.

Using IDAS is inconvenient and inefficient if it is made available to SAS with the %INCLUDE statement. This causes all of the component macros to be compiled even if they are not used. This becomes a more important issue as the size of the system grows. The solution is to place the macros in a CMS MACLIB. Then IDAS is invoked directly by %IDAS from an autocall library. Only those macros actually invoked are compiled.

IDAS provides an excellent foundation for building a statistical modeling system. The most useful macros, not currently available, would be for regression, logistic, analysis of variance and covariance, loglinear, discriminant modeling and various exploratory data analyses (including perhaps graphics). This environment would provide a more complete (depending on the contents of the macros) and an easier environment to work in than that currently available using only existing SAS PROCs.

6. Implementing the Spread versus Level Plot

Any number of data analysis tools could be added to the system of macros discussed in earlier sections of this paper. One such tool is the spread versus level plot. An extensive and quite readable discussion of both the methodology and rationale is available in a Chapter written by Emerson in a book edited by Hoaglin, Mosteller and Tukey, (1983). A brief explanation of the methodology follows appears below.

Various notions of variance stabilization through transformation have existed for at least sixty years. The assumptions necessary to justify the most common traditional statistical procedures and even many nonparametric procedures often involve an assumption of homoscedasticity. That is an idea that is clearly untrue in many cases, difficult to intuitively justify in most cases and certainly unprovable in all cases. In many ways it is much more reasonable to believe that the variance is related to the mean where, for instance, if the mean is \( \mu \), the variance is a function of \( \mu \), say \( \sigma(\mu) \). In a multisample problem each sample has its own mean and a variance that is a function probably, though not necessarily, increasing in that mean. One fairly standard approach to finding a variance stabilizing transformation uses a truncated Taylor series expansion. The result of that development is a suggested transformation, \( \Phi(x) = K \int \sigma^{-1}(x) \, dx \). Here \( \sigma^{-1}(x) \) refers to the reciprocal (not the inverse) of the mean/standard deviation relation.

Hopefully the transformed observations will be more in consonance with the equal variance idea than are the original observations. Several authors have discussed particular instances where the results are less than satisfactory.

More recent approaches to the problem have utilized some EDA ideas. Particularly, location and spread are measured by medians and fourth-spreads instead of the less resistant simple means and variances. Here we will assume the
median is v and the fourth-spread is related to the median by the function, \( t(v) = v b \). Some common assumptions can be subsumed under this model including homoscedasticity \((b=0)\) and constant coefficient of variation \((b=1)\), at least if one is willing to substitute medians and fourth spreads for the usual parameters referenced in such characterizations.

The model is a "log-linear" one in that the usual parameters referenced above are: the median/fourth-spread relation we get:

\[
\phi(x) = K [x^{-\frac{1}{b}}] \frac{dx}{x^b} = K_c \int x^{-\frac{b}{2}} dx = K_1 x^{1-b} K_2 \quad \text{if } b > 1
\]

\[
= K_3 \ln x + K_4 \quad \text{if } b = 1
\]

where the \( K\)'s can be used for "matching" but they do not alter the presence or absence of a relationship between the median and fourth-spread. In sum then, if the plotted points are roughly linear with slope \( b \), the suggested transformation is a power transformation chosen from Tukey's "ladder of transformations" with power 1-b. Note that logarithms take the place of "0" in this scale.

Figure 1 displays a spread versus level plot (axis information intentionally expressed) for some oil shale pyrolysis data wherein a certain characteristic of the shale residue was measured, repeatedly, after subjecting the shale to various heating rates. The plot exhibits points which are in rather good agreement with a straight line of slope equal to one-half. The suggested transformation is therefore logarithmic (it doesn't matter what the base of the logarithm is).

<table>
<thead>
<tr>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>Q</td>
</tr>
<tr>
<td>U</td>
</tr>
<tr>
<td>I</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>

+++++++++++++++

Figure 1 HEAT RATE

Of course the spectre of automated expert systems or expert guidance in problems of a statistical nature looms on the horizon. We are not aware of the forms those systems will eventually take but it seems clear that suggesting appropriate transformations for user supplied data is one fundamental feature that would be desirable in such a system. Spread versus level plots are certainly one vehicle likely to be in use in order to assist an analyst in choosing a transform.

It is reasonably easy to cause SAS software to produce these transformation plots. PROC UNIVARIATE can be used with a BY statement to produce a dataset which contains the medians and inter-quartile ranges (almost the same thing as a fourth-spread at least for fairly large samples) for each of the batches. A simple data step can be used to get the logarithms of those quantities and PROC PLOT (or G3PLOT) used to generate the plot itself. A "robust" line can be fitted using simple data step manipulations or least squares results can be gotten from PROC REG. The "robust" line would perhaps be more consistent with the other methodology used here. It is also true that for even modest sized batches that methodology mitigates against outlying observations in the individual batches creating influential plot points. At such, the resistance offered by robust lines is of questionable value (plot outliers caused by batches with unusual median/fourth-spread combinations are an entirely different matter).

Certainly other strategies for fitting the line are possible. One suggested method is an "eyeball fit". This is difficult to automate of course but there is some evidence that a principal component fit might be a decent substitute, at least for many people. See Mosteller, et al (1981). From the standpoint of easy automation the slope of the first principle component can be gotten from the eigenvectors produced by PROC PRINCOMP and very simple data step manipulations can be used to exactly the slope coefficient. Other strategies for eyeball fitting that allow for a subjective component might include "sliders" in a Macintosh™ setting. There the slider would control a line through some data centroid with an almost constant spread variable. The line of whichever line looked best. Such an approach would likely require an object oriented programming environment and we have not tried to implement it yet.

7. The Effectiveness of the Plot

If the spread versus level plot is to be used in an expert system for the purpose of finding a variance stabilizing transformation it would be useful to determine which version of it is most effective. Here "versions" are created when the general strategy of the methodology is implemented using slightly different tactics (i.e. a "robust" line versus a least squares line and/or using various definitions of spread) To that end we have conducted a Monte Carlo study using the SAS language which is particularly aimed at comparing the robust line/least squares line question. As a frame of reference we will consider the case of the Poisson distribution where of course \( \mu = \sigma^2 \) and \( \sigma(\mu) = \mu^{1/2} \). The usual "fix" for this kind of mean/variance dependence is to use the square root transformation on the original variable.

This is an exact recommendation from an approximate method and the fact is that the only transformation that really stabilizes the spread is a constant transformation. In that case all the batch variances are zero and are thus equal. Nevertheless, the square root transform is a standard remedy in this situation and various authors report that it has been found to work well in practice.

The "theory" outlined above derives a suggested transformation solely from the relationship between the standard deviation and the mean. That means that the transform suggested in Poisson cases is the same one as is suggested for other cases where the same relationship between the spread and level holds. One problem with that is that the standard deviation of the transformed variable depends not only on the standard deviation of the original variable but, very likely, on the other moments as well. Certainly the same may be said about other characteristics of a distribution as well. So, it is not necessarily the case that the same transform will work equally well in all situations where it is "suggested". One aspect of our study focuses on the anomaly just discussed. In addition to the Poisson case already mentioned we consider Normal and "Gamma plus a constant" batches generated so that the mean equals the variance. Since the batch size and the number of batches as well as the actual separation in batch means can all influence the effectiveness and maybe the relative effectiveness of the methodologies a number of different combinations of those factors were considered. We did limit ourselves to fairly large sample sizes since a recently published paper, [Frigge, Hoaglin and Iglewicz, 1989], establishes that for small samples "fourths", and thus related measures of spread, can vary substantially under current definitions of that term. PROC UNIVARIATE currently permits five different
definitions of quantiles to be used, none of which exactly
matches those authors recommendation as to the "best" way
to define a fourth. Some results from a Monte Carlo study
are shown in the following tables. The numbers displayed
correspond to the number of incorrect identifications (out of
100 tries) using a robust line (on the right) and the least
squares line (on the left) in each situation. "Correct
identification" means the fitted slope was between .25 and
.75 (.5 therefore being the nearest half-power
transformation). Obviously, other definitions of correct or at
least helpful could have been used, with, perhaps, different
results.

<table>
<thead>
<tr>
<th>BATCHES</th>
<th>k</th>
<th>n</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td></td>
<td>25</td>
<td>42/54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>29/30, 37/50</td>
</tr>
<tr>
<td>Poisson</td>
<td></td>
<td>25</td>
<td>47/55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>33/45</td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>25</td>
<td>37/50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>22/28</td>
</tr>
</tbody>
</table>

Table 1 The Easy Cases

The "easy cases" were ones in which the batch means (and
variances) varied from 3 to 9. For k = 4 the means were 3,
5, 7 and 9 while for k = 7 each integer between 3 and 7,
inclusively, was used. The value of n refers to the batch
size. Though no formal inferential procedures are reported
here we note that in all 12 cases the robust line produces
more wrong answers than does the least squares line. In all
cases the number of mistakes decreases as the batch size
increases from 25 to 50 and in most cases the number of
mistakes decreases as we increase k from 4 to 7. Holding k
and n fixed we see that the Poisson distribution produces the
most errors while the Normal produces the fewest with the
shifted Gamma falling somewhere in between.

<table>
<thead>
<tr>
<th>BATCHES</th>
<th>k</th>
<th>n</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td></td>
<td>25</td>
<td>80/77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>90/80</td>
</tr>
<tr>
<td>Poisson</td>
<td></td>
<td>25</td>
<td>62/85</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>56/81</td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>25</td>
<td>39/46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>31/35</td>
</tr>
</tbody>
</table>

Table 2 The Hard Cases

The "hard cases" were ones in which the batch means and
variances varied from 0.75 to 2.25 in steps of size 0.25 (for
k = 7) and in steps of size 0.50 (for k = 4). Here, again, the
least squares line outperforms the robust line for the Poisson
and Normal cases. For the Gamma plus a constant
distribution however the least squares line does better. In
the case of the Poisson distribution with small mean many
and perhaps most of the observations are zero and that has
predictable consequences when logs of medians and lower
quartiles have to be taken. When an analyst does a spread
versus level plot "by hand" the problem of logs-of-zero is
quite clear early in the process and some compensatory action
can be taken. In the expert system scenario the user may be
entirely unaware of the problem and it is instructive to know
that the least squares approach adapts to that situation better,
at least if the least squares algorithm "ignores" observations
with missing values as PROC REG does. Of course traps
could be built in to the software so that either the user is
warned about the problem or some corrective action taken on
the robust side too.

References

2. NIOSH (1986), Fatal Accident Circumstances and
Epidemiology (FACE): Occupational Fatalities Due to
Contact with Electrical Energy.
4. Schlesselman, J. J. (1982), Case-Control Studies:
Design, Conduct, Analysis, New York: Oxford
University Press.
5. Hoaglin, D.C. et al. (1983), Understanding Robust and
Exploratory Data Analysis, New York, John Wiley & Sons
6. Frigge, M., et al. (1989), Some Implementations of the
Boxplot, The American Statistician, Volume 43, Number 1
(Febuary) pp. 50-54
7. Mosteller, F., et al. (1981), Eye Fitting Straight Lines,
The American Statistician, Volume 35, Number 3 (August)
pp 150-151