THE BAYESIAN COMPUTER-ASSISTED DATA ANALYSIS (CADA) MONITOR*

Gerald L. Isaacs
Melvin R. Novick
The University of Iowa

Many steps are involved in completing a Bayesian statistical analysis. Some are skilled tasks requiring the expertise of a professional, others are purely mechanical. The former include such tasks as choice of model, specification of the prior distribution and interpretation of the posterior distribution; the latter include such things as the arithmetic necessary to combine the prior distribution with the data to produce the posterior distribution and to produce probability statements from that distribution. Unfortunately, it is all too often the case that the arithmetic gets in the way of the professional's decision-making responsibilities by breaking concentration and line of thought; and at times the sheer bulk of computation precludes the use of advanced techniques by the unaided researcher. What is required is a monitoring system that does all of the arithmetic and, even further, sees to it that all of the steps in the analysis are performed correctly and in their proper sequence. Also, within an instructional process, it can be very useful to have a system that helps a student learn by guiding his steps through a valid statistical analysis even if he doesn't yet fully understand what he is doing. For these and other reasons, a system of Computer-Assisted Data Analysis (CADA) was developed at the University of Iowa (Novick, 1971, 1973). Further investigation into available computer technology coupled with expansion of the theoretical base on which the original system rested resulted in the refinement and expansion of the available programs and the construction of a monitor to facilitate their use (Christ, 1973).

Since CADA was meant for use both as an instructional tool and as a research tool for general application, a search was made to find the most effective means of facilitating wide distribution of the Monitor for use on many computing systems. Due to limitations in available time, manpower, and money, reprogramming on a system-by-system basis was rejected as a viable method of implementing CADA. Since no entirely transportable language for all interactive systems existed, it was decided to pursue a strategy which would permit interdialect translation rather than actual reprogramming. Examination of available hardware and software pointed toward the BASIC programming language as the only possibility for translatability across several manufacturers. A study was then made by Isaacs (1974) which showed that programs written in one dialect of BASIC could easily be translated into that of many other manufacturers' dialects, provided certain specified constraints on the initial programs were observed. The first BASIC version of CADA was then written by Isaacs and Christ in the BASICX dialect for the CDC 3600 at the University of Massachusetts. This was then easily and quickly translated into versions for the Hewlett-Packard 2000C and the Digital Equipment Corporation PDP-11, thus validating the assertions made by Isaacs (1974).

The detailed outline for the current Monitor was developed based on considerations falling in three basic areas -- user interaction, systems constraints and programming considerations. The user interaction is by far the most important consideration. Although the user may be highly skilled in his own subject area, he may be quite unsophisticated in terms of computer skills. The first design rule was then that the user not be required to have any programming skills. He need know only three system-related commands: (1) how to sign on the system; (2) how to start the monitor running; and (3) how to sign off the system.

The second design rule was that the Monitor be self-documenting in terms of options available. The Monitor should be modifiable to include new models, new techniques and improvements in current programs without the user having to wonder whether he has the latest "newsletter" or update sheet.

The third design rule was that the user should not be left "hanging." If an invalid response is given by the user or a numerical integration fails to converge, an error message followed by the stopping of the program is not enough. Control must branch to a subroutine that gives the unsophisticated user enough information so that he can proceed on the information provided to him. Furthermore, whenever possible, input from the user must be checked for validity to avoid system errors such as division by zero, taking the root of a negative number, etc.

The constraints of any language limit that which can be implemented in that language. When programming for translatability across several systems, the constraints become somewhat more de-

* Preparation of this report was supported in part by NSF grant EPP73-00164. The opinions expressed herein are solely those of the authors and do not reflect the opinions or policies of the supporting institutions.

manding and at times preclude the use of features that may be present on one system only, or that differ radically from one system to the next. This consideration, taken in conjunction with the three design rules mentioned above, has governed most of the design of the Monitor and the programs.

While the latest version of the Monitor is currently available for operation on only two systems -- the HP2000F (or HP2000C) and the CDC Cyber -- an attempt has been made to minimize the dependence on features not available in BASIC dialects for other computers. The features used which might be the most limiting are chaining, formatted print statements and external files. However, the systems in which we are most interested have these features available. The formatted print statements were used to present the output material in a visually pleasing way. This is not necessary, per sè, but it is desirable in order to facilitate man-machine interaction, since the intended user is not presumed to be a computer expert. The formatted print statements and files do have analogs in the other dialects we propose to use; however, they will be the ones needing the most change from machine to machine.

Chaining, which is necessary in some larger machines and most smaller machines, is much more central to the logical design of the system. The first consideration was that the user need only know how to sign on the system and would not need to know the names of the individual routines. This implies either a main routine-subroutine system or a monitor program which causes the loading and execution of the proper program. The latter is the system used by us, dictated by the design of most BASIC systems. The main routine-subroutine system has the advantage of ease of parameter passing. However, the number of parameters to be passed in our system is few and the values can easily be stored in files and thus passed from one routine to another. This main routine-subroutine system would allow the user to easily do an analysis in steps at different times. The chaining as used here has the advantage of having in core only the program in use and thus reducing system overhead.

A second consideration supporting this choice for the system was that it should be expandable with little effort on the part of the programmer and with no operational change visible to the user. The monitor system used here permits this. The only change seen by the user is that he is given a choice among a larger set of routines and techniques. The programmer need add only about three lines of coding to the monitor to make a new routine available to the user. At the same time the developer of a new routine will typically be able to use many of the building blocks already part of the system. For example, in the computation of expected utilities the available cumulative normal and cumulative t modules make the required additional programming negligible.

A third consideration supporting this choice was that the user should never be left dangling after he makes an error. In the CADA Monitor, when a program fails, the system chains to a routine in which the user is told to save the output for use by the person maintaining the system and is then returned to the Monitor to continue the session if he so wishes.

An important feature of the Monitor is that user input is screened for validity. Also, since string-handling capability is not highly developed in all BASIC dialects and handling a finite set of responses can be done by much simpler coding, user responses to questions within the program segments have been forced to numeric form. Our decision to forego the extensive use of strings thus greatly enhanced the transportability of the system. The only strings used are for file names and screen control on a CRT.

Programming ease was also considered. A modular method was used in building the routines themselves. Many routines were common across programs (e.g., integrating a beta distribution, calculating an inverse chi highest density region) and were assigned specific line numbers above 800. These routines were coded only once and after being debugged were usable without further effort on the part of the programmer. The programmer then referenced these routines by GOSUB statements to predetermined line numbers with no need to worry about where to put them. Unique portions of programs were then programmed with line numbers below 800. As noted above, the monitor system used enables new programs to be added with little programming effort.

Appendix 1 identifies the various routines that are used in the Monitor. (The user has no actual contact with these.) Appendix 2 shows the chaining sequence in the Monitor connecting the various subroutines. Appendix 3 shows the start of a sample run.

Description of Routines

I. Supervisory Routines

There are three general routines. These are named START, CBCADA, and CERROR.

The first module is called START. This module sets up the necessary restart file and also determines from the investigator what type of terminal he is using. Control is passed directly to the module, CBCADA, which is also referred to as the "Monitor."

Execution of CBCADA or the Monitor gives a short explanation of the data analysis package and gives a listing of the different routines that are available. Also, it zeros the values which will be used for data passage. In other words, any time the Monitor is entered, all previous saved data is lost. Therefore, any data the investigator might wish to retain must be recorded before he returns to the Monitor. The Monitor passes control to any of the fifteen available routines.

CERROR is the routine which can be called from any module in the case that an unexpected error might have occurred; for example, disk error, failure of a routine to converge, etc. Control is passed from the routine which had the problem to

CERROR. CERROR then gives the investigator a message indicating that an error had occurred and passes control back to the Monitor (CBCADA).

At present, only analyses for four models are available (Beta-Binomial, Two Parameter Normal, Comparison of Means, and m-Group Proportions). Also listed for the user are five evaluation components (Student Distribution, Beta Distribution, Inverse-Chi Distribution, Normal Distribution, and Behrens-Fisher Distribution).

II. Beta-Binomial Model

There are three modules which comprise the Beta-Binomial routine. There is a module (FASP) for obtaining a prior distribution, a module (POSTB) which combines the prior distribution and sample data, and from these develops a posterior distribution, and a module (BDIST) for evaluating a Beta distribution.

The method for obtaining a prior distribution is the Fractile Assessment Procedure (FASP). This procedure requires the investigator, as a first approximation to specify the twenty-fifth, fiftieth, and seventy-fifth fractiles of his Bayes distribution. From these three fractiles, the program calculates the Beta distribution which most closely approximates these. The two parameters A and B of the resulting Beta distribution are then reported. The investigator is then given the opportunity to respecify his prior distribution by changing the number of hypothetical sample observations it is worth and also by changing his modal estimate of the binomial parameter PI. After the investigator is satisfied with his prior distribution, he is given the opportunity to form a posterior distribution. If he so wishes, the parameters of his prior distribution are saved and control is passed to the module which forms the posterior distribution POSTB.

The posterior module (POSTB) combines sample data and the previously specified prior distribution to form the posterior distribution, which is also a Beta distribution. The mean, mode, standard deviation and the fifty percent highest density region are reported along with the two parameters which define the posterior distribution. The investigator is then given the opportunity to further evaluate this distribution.

The module (BDIST) which evaluates a Beta distribution allows the investigator the following three options:

1. Compute any P% highest density region.
2. Find the probability that is greater or less than a particular point X.
3. Find the probability between two points X1 and X2.

III. Two Parameter Normal Model

The two-parameter normal model consists of five modules.

1. PRIORS -- Develops prior distribution for the standard deviation.
2. PRIORM -- Develops prior distribution for the mean.
3. POSTN -- Combines prior distribution on the mean, prior distribution on the standard deviation and sample data to form posterior distributions.
4. TDIST -- Evaluates student t-distribution.
5. ICDIST -- Evaluates inverse chi distribution.

These modules can be executed by different routes. Any module can be accessed from the Monitor (CBCADA). The modules may also be accessed in sequence, i.e., by forming the priors, then forming the posteriors and evaluating the posteriors.

IV. Comparison of Two Normal Means

After the investigator has two posterior marginal student t-distributions on means (may be gotten by running two-parameter normal analysis twice), he then is ready to consider the difference of these two random variables. This is done by the Monitor using the modules NORCOM and BERFIS.

NORCOM requires as input the three parameters for each of the investigator's posterior marginal student t-distributions on the mean. These parameters are degrees of freedom, mean and scale factor. The program then calculates the five parameters of a Behrens-Fisher Distribution: NU1, NU2, PSI, ZETA, and EPSILON.

BERFIS allows the investigator to see the following percentage points and central intervals of a Behrens-Fisher distribution.

	Percentage	Central Interval
1.	60%	20%
2.	75%	50%
3.	80%	60%
4.	90%	80%
5.	95%	90%
6.	97.5%	95%
7.	99%	98%

V. M-Group Proportions

There are three modules that comprise the m-group proportions routine.

PRIORP is the module which assists the investigator in fitting a prior distribution to his beliefs about a proportion PI.

PROPOR is the module which calculates the Bayesian and Model II estimates for the true proportions of each group of data from the prior distribution and sample data. The following values for each group of data are then reported: Number of successes, number of observations, sample proportion, Model II estimates and Bayes estimates. If some of the Model II estimates are zero or negative, then the Model II estimates are not given. Finally, the averages across groups are reported and the posterior estimate of Phi Gamma is given.

MARPRO is the module which is used to estimate the true value for the point estimates and marginal distributions for each group being analyzed. Probability statements about a random group also may be made.

VI. Miscellaneous Modules

There are two miscellaneous modules, STAT and NDIST.

STAT is the module which calculates a mean, standard deviation and sums of squared deviations from the investigator's input data. Data is input in groups of 10. This is then printed out for the investigator to make sure it is correct before it is stored. The mean, standard deviation and sums of squares are then reported after all data has been input and checked.

NDIST is the module which is used to evaluate a normal distribution. The investigator must input the mean and standard deviation. The investigator may see any of the following: any P% highest density region; the probability that is less than any point X; or the probability that is between any two points X1 and X2.

Appendix 1

Monitor Contents

I. Supervisory Routines

A. START: Initializes Monitor, control passes automatically to CBCADA.

B. CBCADA: Monitors all regular statistical routines.

C. CERROR: Gives instructions when a program fails.

II. Beta-Binomial Model Routines

A. FASP: Assists in fitting prior knowledge to the beta class using the fractile assessment and other methods.

B. POSTB: Combines a beta class prior with binary data to give a beta posterior and provides some detailed information concerning this distribution.

C. BDIST: Evaluates the probability integral of a beta distribution.

III. Two Parameter Normal Model

A. PRIORS: Fits prior knowledge (marginally) on the standard deviation to an inverse chi distribution.

B. PRIORM: Fits prior knowledge (conditionally) on the mean to a normal distribution.

C. POSTN: Combines the inverse chi and normal priors with normal data to give a posterior distribution and provides some detailed information on that distribution.

D. ICDIST: Evaluates the probability integral of a nonstandard inverse chi distribution.

E. TDIST: Evaluates the probability integral of a nonstandard student t-distribution.

IV. Comparison of Normal Means

A. NORCOM: Entry of posterior distribution on two populations.

B. BERFIS: Evaluates Behrens-Fisher distribution.

V. m-Group Proportions

A. PRIORP: Evaluates exchangeable prior information on any of a set of proportions for use in an m-group proportion routine.

B. PROPOR: Solves the Lindley equations for a set of binary data and thus provides modal estimates for the estimation of proportions in m-groups.

C. MARPRO: Computes marginal probabilities from the estimation of proportions in m-groups.

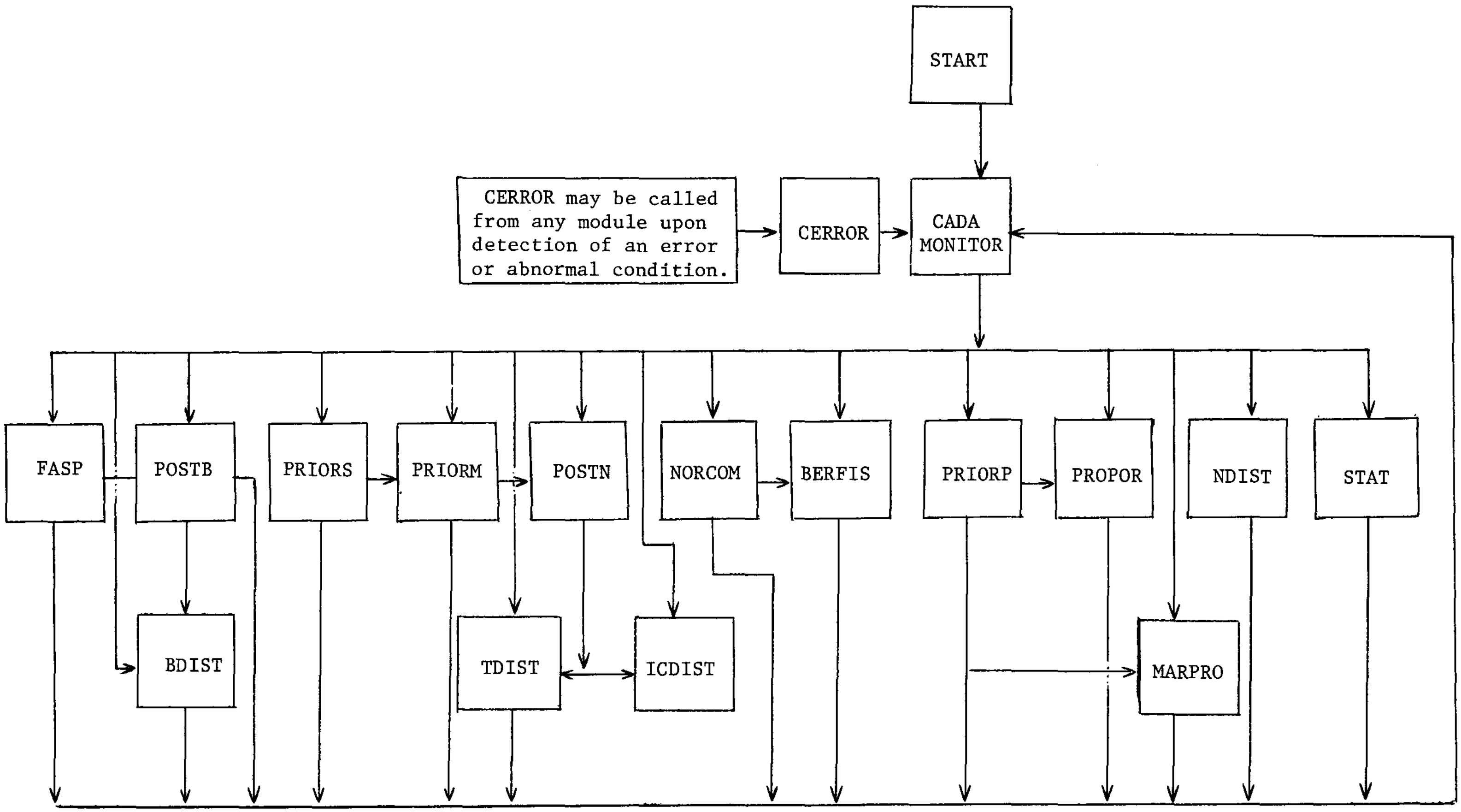
VI. Service Routines

A. STAT: Calculates the mean, standard deviation, and sum of squared deviations from the mean for a set of data.

B. NDIST: Evaluates the probability integral of a nonstandard normal distribution.

Appendix 2

Chaining Sequence



Note: Any program can chain to CERROR upon detection of an abnormal condition.

Appendix 3

Example of CADA

```
GET-START
RUN
START

IF YOU ARE USING A BEEHIVE TERMINAL, PLEASE TYPE A 1.
IF YOU ARE USING AN ADDS TERMINAL, PLEASE TYPE A 2.
IF YOU ARE USING A HAZELTINE TERMINAL, PLEASE TYPE A 2.
IF YOU ARE USING A SUPER BEEHIVE, PLEASE TYPE A 3.
OTHERWISE TYPE A 0.
?0

                   COMPUTER ASSISTED DATA ANALYSIS
IF YOU WISH AN EXPLANATION TYPE 1, ELSE TYPE 0
?1



    THIS PACKET OF PROGRAMS PROVIDES A GROUNDING IN THE
FUNDAMENTALS OF BAYESIAN METHODS OF STATISTICAL INFERENCE.
THESE ROUTINES ARE DESIGNED TO GUIDE THE RESEARCHER WHO HAS
ONLY A MINIMAL ACQUAINTANCE WITH BAYESIAN METHODS, STEP-BY-
STEP THROUGH A COMPLETE BAYESIAN ANALYSIS.  A LIST OF THE
ROUTINES FOLLOWS:
WHEN YOU WISH TO CONTINUE TYPE A 1.
?1

      ROUTINES AVAILABLE ON CADA MONITOR
1.  CALCULATE MEAN, STANDARD DEVIATION, SUMS OF SQUARES
2.  PRIOR--BETA BINOMIAL MODEL
3.  POSTERIOR--BETA BINOMIAL MODEL
4.  PRIOR--TWO PARAMETER NORMAL--MARGINAL DIST. FOR STANDARD DEV.
5.  PRIOR--TWO PARAMETER NORMAL--CONDITIONAL DIST. FOR MEAN
6.  POSTERIOR--TWO PARAMETER NORMAL
7.  COMPARISON OF TWO NORMAL MEANS
8.  PRIOR--M GROUP PROPORTIONS
9.  POSTERIOR--M GROUP PROPORTIONS
10. POSTERIOR--MARGINAL PROPORTIONS
11. EVALUATE BETA DISTRIBUTION
12. EVALUATE STUDENT DISTRIBUTION
13. EVALUATE INVERSE CHI DISTRIBUTION
14. EVALUATE NORMAL DISTRIBUTION
15. EVALUATE BEHRENS FISHER DISTRIBUTION
IF DURING THE RUNNING OF ANY OF THE ABOVE MODULES YOU WISH TO
ABANDON YOUR PRESENT COMPUTATIONS AND RESTART THAT MODULE,
YOU CAN DO SO BY TYPING -9999 WHENEVER YOU ARE ASKED FOR INPUT.
YOU WILL THEN RESTART THAT MODULE WITH ALL VALUES SET AS THEY
WERE WHEN YOU FIRST ENTERED THAT MODULE.

IF YOU WANT TO RUN ONE OF THE ABOVE ROUTINES, TYPE ITS NUMBER
OTHERWISE TYPE A 0.
?
```

References

Abramowitz, M. and Stegun, I. A. Handbook of Mathematical Functions. U.S. Department of Commerce, NBS Applied Mathematics Series, 1964.

Borgmann, R. E. and Ghosh, S. P. Statistical distribution programs for a computer language. Research Report, IBM Watson Research Center. Yorktown Heights, New York: RC-1094, 1963.

Christ, D. E. The CADA Monitor. ACT Technical Bulletin No. 12. Iowa City, Iowa: The American College Testing Program, 1973.

Fröberg, C. E. Introduction to Numerical Analysis. Reading, Massachusetts: Addison Wesley Publishing Company, Inc., 1965.

Hastings, C. Approximations for Digital Computers. Princeton, New Jersey: Princeton University Press, 1955.

IBM System 136 Scientific Subroutine Package Version III (360A-CM - 03X). Programmer's Manual, Fifth Edition, Gh20-0205-4. White Plains, New York, 1968.

Isaacs, G. L. Interdialect translatability of the BASIC programming language. Sigcue Bulletin, 1974, 8, 11-22.

Isaacs, G. L., Christ, D. E., Novick, M. R., and Jackson, P. H. Tables for Bayesian Statisticians. Iowa City, Iowa: The University of Iowa, 1974.

Jackson, P. H. Formulae for generating highest density credibility intervals. ACT Technical Bulletin No. 20. Iowa City, Iowa: The American College Testing Program, 1974.

Novick, M. R. Bayesian computer-assisted data analysis. ACT Technical Bulletin No. 3. Iowa City, Iowa: The American College Testing Program, 1971.

Novick, M. R. High school attainment: An example of a computer-assisted Bayesian approach to data analysis. International Statistical Review, 1973, 41, 264-271.

Novick, M. R. A course in Bayesian statistics. The American Statistician, 1975, 29, 94-97.

Novick, M. R. and Jackson, P. H. Statistical Methods for Educational and Psychological Research. New York: McGraw-Hill, 1974.

Novick, M. R. and Lindley, D. V. Assessing utilities (with applications to education). Unpublished manuscript, August 1975.

Stroud, A. H. and Secrest, D. Gaussian Quadrature Formulas. Englewood Cliffs, New Jersey: Prentice-Hall, 1966.

Sukhatme, P. V. On Fisher and Behrens' test of significance for the difference in means of two normal samples. Sankhya, 1938, 4, 39-48.