DOCUMENT RESUME

ED 083 820 EM 011 543

AUTHOR	Christ, David E.
TITLE	The CADA Monitor.
INSTITUTION	American Coll. Testing Program, Iowa City, Iowa. Research and Development Div.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C.
REPORT NO	ACT-TB-12
PUB DATE	Jan 73
NOTE	11p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Bayesian Statistics; *Computer Programs; Computer Science; Man Machine Systems; Program Descriptions; *Programing; *Programing Languages; *Statistical Analysis; *Translation
IDENTIFIERS	BASIC; CADA; CADA Monitor; Chaining; *Computer Assisted Data Analysis; Formatted Print Statement; Interdialect Translation; University of Iowa

ABSTRACT

A system of computer-assisted data analysis (CADA) was developed at the University of Iowa to handle the mechanical, arithmetical and computational tasks associated with Bayesian statistical analysis. The original programs were subsequently expanded and a CADA monitor constructed to aid their use; a search was then undertaken to find a means of facilitating the monitor's application on diverse computer system. Since no entirely transportable language for all interactive systems existed, a strategy of interdialect translatability was pursued and BASIC was selected for use. A BASIC version of CADA was written and proved to be easily translatable. The monitor itself was designed so that no user programing skills were required and so that it was self-documenting and readily modifiable. In addition, the system was built so as not to leave its unsophisticated users hanging suspended when errors occurred, but to direct them ahead on the basis of available information. The two system features which most limited translatability across several systems were chaining and formatted print statements, but these were so desirable that they were retained. (LB)

ERIC

ED 083820

ACT TECHNICAL BULLETIN NO. 12

THE CADA MONITOR

by

David E. Christ

The University of Iowa

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

The Research and Development Division

The American College Testing Program

P. O. Box 168, Iowa City, Iowa 52240

January, 1973

EM 011 543

FILMED FROM BEST AVAILABLE COPY

THE CADA MONITOR

David E. Christ

The University of Iowa

Several elements go into a Bayesian statistical analysis. Some are skilled tasks requiring the expertise of a professional and others are purely mechanical. The former include such tasks as choice of model, specification of the prior, and interpretation of the posterior distribution; whereas the latter include such things as the arithmetic necessary to take statements about the prior and combine them with the data to produce the posterior distribution and to produce probability statements about parameters using the posterior distribution. Unfortunately, it is all too often the case that the arithmetic gets in the way of the professional's decision-making task 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. For these and other reasons, a system of Computer-Assisted Data Analysis (Novick, 1971) was developed at The University of Iowa. Further investigation into available computer technology coupled with expansion of the theoretical base on which the original system rested has resulted in the refinement and expansion of the available programs and the construction of a monitor to facilitate their use.

Since CADA (Computer-Assisted Data Analysis) was meant 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 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 (1972) 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.

The detailed outline of 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 be required to have no 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 to 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 a numerical integration fails to converge, an error message followed by the stopping of the program is not enough. Control must branch to a point where the unsophisticated user can proceed on the information available 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 implementation limit what can be programmed in that language. When programming for translatability across several systems, the constraints become somewhat more demanding 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, with the three design rules mentioned above, has governed most of the design of the monitor and the programs.

While the monitor is currently available for operation on only three systems, an attempt has been made to minimize the dependence on features not available in BASIC dialects for other computers. The two features used which might be the most limiting are chaining and formatted print statements. However, the systems in which we are <u>most</u> interested have these features available. The formatted print statements were used to present the output and textual material in a visually pleasing way. This is not necessary, per se, but is desirable to facilitate the man-machine interaction since the intended user is not presumed to be a computer expert. The formatted print statements 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.

ERIC

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 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 are values known to the user, usually understood by him, and normally recorded, to be used in any published record of the analysis; thus, it is reasonable to ask the user to reenter the parameters when necessary. This also allows 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 for the system is 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 the choice of choosing 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. A third consideration is 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. All user input is screened for validity. 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.

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 5000. 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 2000. As noted above, the monitor system used enables new programs to be added with little programming effort.

The accompanying appendices show a sample of the monitor output, give a listing of the current package contents, and outline the chaining sequence.

APPENDIX I

Monitor Output

```
RUN CBCADA

                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 ACQUAINTENCE WITH BAYESIAN METHODS, STEP-BY-
STEP THROUGH A COMPLETE BAYESIAN ANALYSIS.  A LIST OF THE
ROUTINES FOLLOWS:

1.  PRIOR BETA-BINOMIAL MODEL
2.  POSTERIOR BETA-BINOMIAL MODEL
3.  PRIOR TWO PARAMETER NORMAL--MARGINAL DIST FOR STANDARD DEV
4.  PRIOR TWO PARAMETER NORMAL--CONDITIONAL DIST FOR MEAN
5.  POSTERIOR TWO PARAMETER NORMAL
6.  PRIOR M-GROUP PROPORTIONS
7.  POSTERIOR M-GROUP PROPORTIONS
8.  EVALUATE STUDENT-DISTRIBUTION
9.  EVALUATE BETA-DISTRIBUTION
10. EVALUATE INVERSE CHI-DISTRIBUTION
11. EVALUATE NORMAL DISTRIBUTION
14. CALCULATE MEANS,STANDARD DEV., SUMS OF SQUARES

IF YOU WANT TO RUN ONE OF THE ABOVE ROUTINES,TYPE ITS NUMBER
OTHERWISE TYPE A ZERO.
? 1
```

ERIC

APPENDIX II

Package Contents

I. Supervisory Routines

   A. CADA - Monitor

   B. ERROR - Gives instructions when a program fails

II. BETA - Binomial Model Routines

   A. PRIORB - Assists in fitting prior knowledge to the beta class

   B. POSTB - Combines a beta class prior with binary data to give a beta posterior

III. Two Parameter Normal Model

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

   B. PRIORM - Fits prior knowledge (conditional) on the mean to a normal distribution

   C. POSTN - Combines the inverse chi and normal priors with normal data to give posterior distribution

IV. 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

V. Evaluation Routines

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

   B. BDIST - Evaluates the probability integral of a beta distribution

ERIC

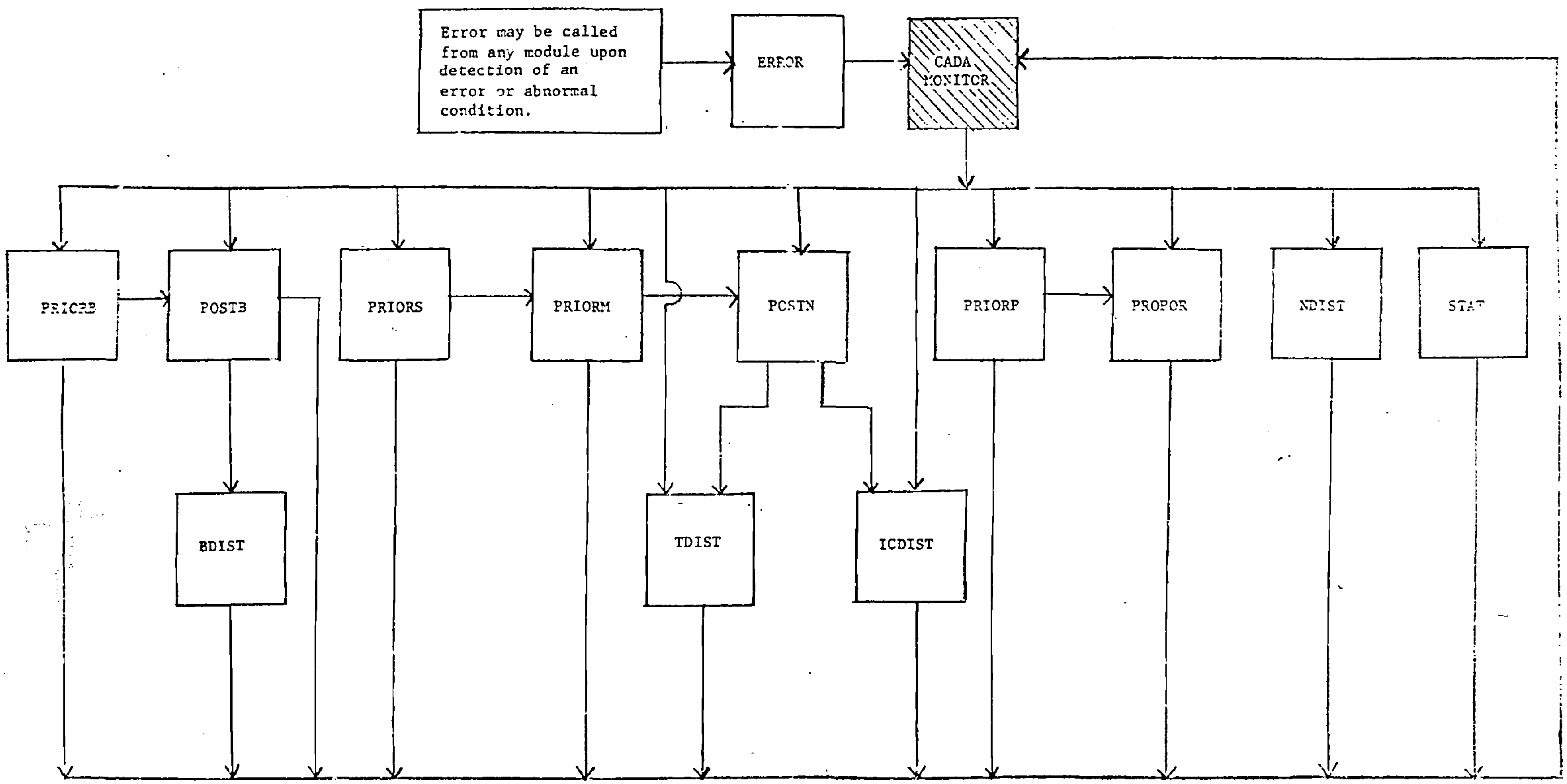
C. ICDIST - Evaluates the probability integral of a nonstandard inverse chi distribution

D. NDIST - Evaluates the probability integral of a nonstandard normal distribution

VI. Service routine STAT calculates the mean, standard deviation, and sum of squared deviations from the mean for a set of data

APPENDIX III

Chaining Sequence



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

Bibliography

Isaacs, G. L. Interdialect translatability of the BASIC programming language. *ACT Technical Bulletin No. 11*. Iowa City, Iowa: The American College Testing Program, 1972.

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