COMPUTER TECHNOLOGY

A FORTRAN program for lag sequential analysis of contingency and cyclicity in behavioral interaction data

GENE P. SACKETT, RICHARD HOLM, CHARLES CROWLEY, and ALLEN HENKINS
*University of Washington, Child Development and Mental Retardation Center,
Regional Primate Research Center, Seattle, Washington 98195*

One method for identifying contingency relationships in a sequence of observed behaviors studies occurrences of matching behaviors at successive event or time steps from a criterion event. This paper briefly summarizes the method and presents a computer program for generating lag conditional probabilities and associated measures for assessing statistical significance.

Until recently, most sequential analyses of observational behavior data were performed using simple one-step conditional probabilities (e.g., Bakeman & Brown, 1977). In addition, Markov analyses have been used for studying sequential patterning among a number of behaviors (e.g., Altmann, 1965). A nonparametric variant of auto- and cross-correlation time series methods has also been described for studying behavioral contingencies. Although this lag sequential technique does not directly identify exact patterns of occurrence among behaviors, it has several advantages over Markovian methods and provides sequential information unavailable from simple one-step contingency measures. In brief, these advantages include (1) obtaining measures of contingency among behaviors far apart in time or sequentially occurring events, (2) obtaining direct measures of cyclicity for a single behavior (autocontingency) or phase relationships between a number of behaviors (cross-contingency), and (3) yielding much of the information available in Markov analysis with far less output. Sackett (in press) has presented a complete description of the lag technique. Here, we present a computer program for calculating lag sequential probabilities, preceded by a brief description of lag methods and terminology.

THE LAG METHOD

Suppose that four discrete behaviors, A-B-C-D, are measured during an observational test session. Suppose further that only one behavior can occur at any time,

Various aspects of the work leading to development of this lag analysis method and the program presented here were supported by Grant HD-02274 to the University of Washington Child Development and Mental Retardation Center and by Grant HD-08633 to the senior author, both from the Mental Retardation Branch of NICHD, and by Grant FR-00166 from the Animal Resources branch of NIH to the Regional Primate Research Center at the University of Washington.

and no time can pass without observing one of these behaviors (i.e., a mutually exclusive, exhaustive behavior categorization). A representative test session is shown in Table 1. Ten behavior changes are observed (sequential events), each lasting for some number of seconds. In analyzing these data, the basic concept is that of an event or time lag. The analytic procedure illustrates this concept. (1) Choose one of the behaviors as a criterion behavior. (2) Count the number of times that every behavior (including the criterion itself) follows the criterion as (a) the next behavior (Lag 1), (b) the second behavior after the criterion (Lag 2), (c) the third behavior after the criterion, (d) the MAX LAG behavior after the criterion. MAX LAG is the largest sequential step to be analyzed. This procedure can be done with one or more behaviors as criteria.

Event lags measure the number of times each instance of the criterion is followed by every behavior in the event stream up to the MAX LAG event in the series. Event sequences ignore the duration of each behavior. Time lags are counted in the same way, except the

Table 1
Data Used to Illustrate the Event and Time Lag Concepts

Real-Time Sequence			1-Sec Time Intervals			
Sequen- tial Event	Behav- ior	Dura- tion*	Time	Behav- ior	Time	Behav- ior
1	A	3	1	A	13	B
2	B	2	2	A	14	C
3	D	4	3	A	15	D
4	A	1	4	B	16	D
5	B	3	5	B	17	A
6	C	1	6	D	18	A
7	D	2	7	D	19	A
8	A	4	8	D	20	A
9	B	3	9	D	21	B
10	C	1	10	A	22	B
			11	B	23	B
			12	B	24	C

*In seconds.

events are lagged in successive real-time units. This is illustrated on the right side of Table 1, listing the real-time series in 1-sec units.

Event Lags

The upper portion of Table 2 presents the event-lag analysis of the Table 1 data to MAX LAG of 5. The frequency of occurrence (n) and the unconditional probability (n/N_{total}) for each behavior in the data as a whole are shown in Row 1 (overall). The next five rows give lag statistics for Behavior A as the criterion. Referring to the right half of Table 1, A occurs three times in the total event series. Behavior B matches A at Lag 1 after each of these occurrences. The matching frequency for B at Lag 1 is thus 3, with no Lag 1 matches for Behaviors C and D. In an event series with mutually exclusive behaviors, a particular event cannot follow itself at the next lag. Thus, Behavior A cannot occur at Lag 1 when it is the criterion (blanks in Table 2).

The number 3 in the total column of Table 2 indicates that the criterion could be followed by other behaviors at Lag 1 a total of three times. The lag conditional probability for each behavior matching the criterion at any lag is $n_{\text{match}}/N_{\text{total}}$. At Lag 1, these

values are $3/3 = 1$ for Behavior B and 0 for the other behaviors.

At Lag 2 (two steps away from each criterion occurrence), C follows A twice and D follows once, yielding probabilities of .67 and .33. At Lag 3, the total number of behaviors that can match the criterion is reduced by 1 because the criterion can only be followed by two more events after its occurrence at Event 8. This "falling off at the end" reduces the N total by 1 whenever the criterion occurs at an event equal to the current lag size away from the end of the series. The remainder of the Table 2 event summary gives frequencies and probabilities for Lags 3-5.

Time Lags

The lower portion of Table 2 presents three types of time-lag analyses. The "overall" row gives total duration and proportion of total test time for each behavior. In computing time lags, two decisions must be made. First, a time unit defining each lag must be chosen. A 1-sec unit was picked for this example. Second, one of three possibilities for initiating the lagging process (a lagging trigger) must be chosen.

The first possibility is a level trigger, with lagging initiated after every instance of the criterion. Referring

Table 2
Event and Time Lag Matching Frequencies and Probabilities for the Example Data in Table 1 with Behavior A as the Criterion

Lag	Number of Matched Occurrences					Probability			
	A	B	C	D	Total	A	B	C	D
Event Lags									
Overall*	3	3	2	2	10	.33	.33	.20	.20
1		3	0	0	3	.00	1.00	.00	.00
2	0	0	2	1	3	.00	.00	.67	.33
3	1	0	0	1	2	.50	.00	.00	.50
4	1	1	0	0	2	.50	.50	.00	.00
5	0	1	1	0	2	.00	.50	.50	.00
Time Lags									
Overall**	8	8	2	6	24	.33	.33	.08	.25
Level Triggered									
1	5	3	0	0	8	.63	.38	.00	.00
2	3	5	0	0	8	.38	.63	.00	.00
3	1	6	0	1	8	.13	.75	.00	.13
4	0	4	2	2	8	.00	.50	.25	.25
5	0	2	1	4	7	.00	.29	.14	.57
Trailing Edge Triggered									
1	0	3	0	0	3	.00	1.00	.00	.00
2	0	3	0	0	3	.00	1.00	.00	.00
3	0	2	0	1	3	.00	.67	.00	.33
4	0	0	2	1	3	.00	.00	.67	.33
5	0	0	0	2	2	.00	.00	.00	1.00
Leading Edge Triggered									
1	2	1	0	0	3	.67	.33	.00	.00
2	2	1	0	0	3	.67	.33	.00	.00
3	1	2	0	0	3	.33	.67	.00	.00
4	0	2	1	0	3	.00	.67	.33	.00
5	0	1	0	2	3	.00	.33	.00	.67

*Event lag: n = overall frequency in data; p = proportion of total column behavior changes.

**Time lag: n = total seconds in data; p = proportion of total column test duration.

to the right half of Table 1, Behavior A as criterion occurs at Seconds 1, 2, 3, 10, 17, 18, 19, and 20. Each of these is a level trigger (Lag 0) for lag initiation. The A at Second 1 is followed by A at Lag 1 and Lag 2, B at Lags 3 and 4, and D at Lag 5. At Second 2, A follows itself at Lag 1, is followed by B at Lags 2 and 3, and by D at Lags 4 and 5. This procedure continues for the remaining six level triggers, yielding the matching frequencies and probabilities in the first "time lags" section of Table 2. These probabilities measure the chance that each behavior will match the criterion up to MAX LAG of 5 sec after every 1-sec criterion occurrence.

A second trigger is the trailing edge (offset) of each continuous set of criterion behavior occurrences. The final time unit of each set serves as Lag 0. Referring again to Table 1, Behavior A has offsets at Seconds 3, 10, and 20. The trigger at Second 20 can only be followed for four lags due to "falling off at the end" of the data. Behavior B follows each trailing-edge trigger at Lags 1 and 2. At Lag 3, B follows twice and D once. Lag 4 has two B matches and one D, while Lag 5 has two D matches. These matching frequencies and associated probabilities, measuring the chances of each behavior following the offset of A, are shown in the second set of time-lag values in Table 2.

The third trigger choice is the leading edge (onset) of the criterion. Lags are triggered from these occurrences, but not from any other criterion occurrences within a continuous series of the criterion behavior. For the data in Table 1, A has onsets at Seconds 1, 10, and 17. A follows itself twice and is followed by B once at Lags 1 and 2. At Lag 3, A follows once and B twice. At Lag 4, B occurs twice and C once, with B occurring once and D twice at Lag 5. The bottom time-lag section of Table 2 presents these matching frequencies and probabilities, which measure the chance that each behavior will follow A onsets up to a MAX LAG of 5 sec.

In actual experiments, data usually consist of several tests sessions on a particular subject or group. The lag process is performed without lagging across the individual test sessions. Matching frequencies for all individual sessions are summed, and the total is used to compute lag conditional probabilities.

Hypothesis testing is a complex issue addressed in detail by Bakeman (1978), Gottman and Notarius (1976), and Sackett (in press). Most methods currently used to assess statistical significance employ binomial statistics estimated from the large sample z method, or a chi-square approach. The program presented next calculates appropriate z values under the null hypothesis that lag probabilities occur by chance at the unconditional probability of the matching behavior. Values necessary for making chi-square tests are also generated by the program.

THE FORTRAN PROGRAM

Using the Program: Control and Data Cards

The program computes lag statistics for event and time sequential data. It runs on the CDC6400 using CM 62000 words. Data are read as a time series. Each item in the series is called an observation. Each observation must specify three or four items as integer number codes read in each data field. The first three items are (1) a behavior code for the event that began or ended at the time of the observation, (2) a criterion code for the special behavior from which lagging will occur (the criterion and behavior code fields can be the same or different), and (3) an end code signaling either the end of a trial or the end of a session.

A trial is ordinarily a continuous observation session. No lags are computed across an end-of-trial code, but statistics continue to be accumulated together for all trials of a session. The tables with accumulated statistics are printed and the statistics cleared at the end of each session. End-of-trial codes are used only to inhibit lagging across a boundary. A session may have one or several trials. The end-of-trial or -session codes can be intermixed with the behavior (or criterion) codes by specifying the same card columns for both fields. Alternatively, the end codes can appear in a field of their own, which will be blank for all real observation lines.

The fourth item is required only for time sequential analyses that use the observation duration as well as the event codes. Duration is an integer number of time units, with all observations in the same units.

These three or four data fields form an observation. They occupy an observation field in the input data. Data can be on IBM cards or another source, such as tape or a disk file. Data are referred to here as 80-column cards, although they may actually be card images. An observation field must be some fixed number of continuous columns on one or more card images. Observation fields can be on three whole cards, two whole cards, one whole card, or part of one card. If an observation is only part of one card, then several observation fields per card are read. However, no field will be read across a card boundary, and the observation fields must all be the same length and must be evenly spaced on the card.

The observation fields are found on the data cards according to the FIELD CARDS specified by the user. A data field is a sequence of up to seven card columns (digits) that are concatenated to form a single integer. These columns can be concatenated in a different order than the way they appear on the data cards. That is, data fields can be formed by picking columns from anywhere in the observation field and in any order. If there are several observation fields per card, the column numbers for the data fields are all relative to the

Table 3

TITLE CARD: The first card contains any arbitrary labeling information printed on the top of every output page.

PARAMETER CARD: The second card contains most of the control information in 11 fields.

Columns Field Name and Meaning of Options

1	Type of analysis (two possible values).
1—event sequential
2—time sequential
2	Form of input. If zero or blank, a record containing end-of-trial or end-of-session code will be counted as an observation. This mode is used when the end-of-trial or -session codes are intermixed in the same columns as the criterion or behavior codes. If not zero or blank, records that contain end-of-trial or -session codes will be counted and considered the last observation of the trial.
3	Lagging trigger (three possible values).
1—lag from the onset of the criterion only
2—lag from all occurrences of the criterion
3—lag from the offset of the criterion only

For event sequential analyses, all three lag types will produce the same results if an event cannot follow itself. If the criterion and behavior codes are in the same columns and the lag type is 3, then a criterion cannot follow itself as a behavior at Lag 1. The expected values of the other behaviors are adjusted to reflect this restriction on randomness of occurrences (see Sackett, in press, for details). |
| 4-6 | Step size. An output line is printed for every "step size" of lags. Normally, this will be 1, but it can be larger to assess longer lags without producing giant quantities of output. |
| 7 | Summarization of lags. This is only used with step size greater than 1. Two options are allowed.
0—each output reflects the actual statistics at that lag with no averaging done. The lags in between the steps are ignored.
1—each output line is the sum of the present lag and all lags between it and the lag of the previous output line. For example, with a step size of 5, Lag 5 frequencies are the sum of the frequencies for Lags 1-5, and computations are based on these sums. |
8-11	The maximum lag (stepsize*maximum number of steps). The maximum lag value cannot be greater than 800.
12-18	The end-of-trial code. No lags can extend over this code. If zero is a legal behavior code, this value should be nonzero or blank even with only one trial per session.
19-25	The end-of-session code. The last value in the end-of-trial field for the session. This value terminates reading of data and causes the generation of output tables. With only one session, this code may be omitted. If the end-of-trial field is intermixed with the behavior field, this code must be different from any behavior coded.
26	Echo input. If this is not zero or blank, the data will be printed on the output as it is read.
27	Data source. If zero or blank, the data should follow the control cards on "INPUT." If not zero or blank, the data are read from logical file name (lfn) "DATA." Be sure that "DATA" is set up and rewound before the job is run.
28-29	Columns per observation. If this is 0, blank, or greater than 29, each observation (field) must be one, two, or three whole cards, and one card cannot contain

parts of two observations. A value between 1 and 79 is the number of columns in each observation field. Each card is assumed to contain as many observations as fit only one card, including a partial length observation at the end of the card if the number does not divide 80 evenly. For example, if this number is 60, the program assumes two observations per card, one in Columns 1-60 and one in Columns 61-80. Only the first 20 columns in 1-60 can be used, since the part card (61-80) only has 20 columns. The following is a list of the number of observations per card that will be assumed for various values: 40-79 = 2 fields, 27-39 = 3 fields, 20-26 = 4 fields, 16-19 = 5 fields, 14-15 = 6 fields, 12-13 = 7 fields, 10-11 = 8 fields, 9 = 9 fields, 8 = 10 fields, 7 = 12 fields, 6 = 14 fields, 5 = 16 fields, 4 = 20 fields, 3 = 27 fields, 2 = 40 fields, 1 = 80 fields.

CRITERION CODE CARDS: The next one or two control cards contain criterion codes: (1) 11 codes to a card, (2) in seven-column wide fields, (3) with a maximum of 20 criteria, (4) the last criterion being followed by a -1 field, which must be present. A separate analysis is done, and a separate output table is produced for each criterion specified. Data from the same session are used for analyzing each criterion. The data will be written in binary to a scratch file and reread from that, so the first criterion run time takes longer than the others, since it decodes the input.

BEHAVIOR CODE CARDS: The next set of one or two control cards contains codes for the matching behaviors to be studied. They are specified like the criterion codes, 11 to a card in seven-column wide fields starting with Column 1. Up to 20 matching behaviors can be specified. The last field must be seven columns wide and contain a -1. This marker must be present even if it is the only field on the card, because it is used as a flag to stop the scan for more codes.

CRITERION FIELD CODES: This card specifies the input card columns that contain the criterion codes. From one to seven columns can be specified in up to seven three-column fields. That is, first input card column in 1-3, second in 4-6, and so on, to seventh in 19-21. If the criterion code is less than seven columns, leave the last fields blank. The criterion code will be constructed by concatenating these columns in the order specified. For example, suppose the criterion is to be read from Columns 2, 7, 23, and 5 in that order. Starting from Column 1, the criterion field card would then be

2 7 23 5

and in the computer, these criteria would be represented as $1000 \times \text{column} 2 + 100 \times \text{column} 7 + 10 \times \text{column} 23 + \text{column} 5$. Normally, there will be only one data card per observation, but there can be up to three cards per observation. The second card is then represented as Columns 81-160, and the third card as columns 161-240. If all the columns are less than 80, one card per observation will be read. If all the columns are less than 160 and some are greater than 80, two cards per observation will be read.

BEHAVIOR CODE FIELD CARD: This card specifies the card columns containing the behavior codes. The card form is exactly the same as the criterion field card.

END-OF-TRIAL FIELD CARD: This card specifies the cards columns containing the end-of-trial codes. The form is exactly like the criterion field card.

DURATION FIELD CARD: This card is to be included only if a time sequential analysis is specified (analysis type = 2). It specifies the card columns where duration is found. The form is exactly like the criterion field card.

Table 4
Example of Control Cards for Running the Event and Time Lag Analyses in Table 2 and From the Data in Table 1

		Card Columns																											
Card		1	2	3	4	5	6	7																					
1	EVENT LAGS, 1 TRIAL, 1 SESSION, 10 OBSERVATIONS PER CARD, MAX LAG 5																											(Title)	
2	102 10 5 9 00 1																											(Parameter)	
3	1 -1																											(Criterion Code)	
4	1 2 3 4 -1																											(Behavior Code)	
5	1																											(Criterion Field)	
6	1																											(Behavior Field)	
7	1																											(End-of-Trial Code)	
8	12412341239																											(Data)*	
1	TIME LAGS, 1 TRIAL, 1 SESSION, 10 OBSERVATIONS PER CARD, MAX LAG 5																											(Title)	
2	202 10 5 9 00 2																											(Parameter)	
3	1 -1																											(Criterion Code)	
4	1 2 3 -1																											(Behavior Code)	
5	1																											(Criterion Field)	
6	1																											(Behavior Field)	
7	1																											(End-of-Trial Code)	
8	2																											(Duration Field)	
9	132244112331421423319																											(Data)*	

Note—Parenthetical words are comments and would not be punched on actual cards. *Behaviors A, B, C, D are coded as 1, 2, 3, 4 in the input data. In event lags the 10 behavior changes are in 10 successive one-column fields. In time lags the behavior changes and associated durations are in 10 successive two-column fields. The first column of each field gives the event; the second gives the duration.

beginning of the observation field, not to the beginning of the card. For example, if each observation field is 20 columns wide, there will be four observations per card and all FIELD CARD (date field) column markers must be between 1 and 20.

To run a series of analyses on one session of sequential data, it is necessary to apply one set of control information cards (7 to 10 cards), as shown in Table 3.

If several data sets are to be analyzed in one run, their control card sets must follow each other sequentially on "INPUT." The data cards can follow their control card sets on "INPUT," or they can be sequentially placed in "DATA," or some can be on "DATA" and some can follow the control cards. If the data cards follow the control cards, they are placed between the control cards for their analysis and the control cards for the next analysis. The program will read the data cards until the end-of-session code is read. The data sets on "DATA" must not contain any internal end-of-record (7-8-9) cards or any internal end-of-file (6-7-8-9) cards.

The last set of control cards must be followed by two cards, each having a 9 in Column 1. If these are omitted, everything will still work, except the program will abort with an EOF after it has finished. The page before each output table will be a record of the parameters chosen. The number of records read is also on this page.

Example Control and Data Cards

Table 4 gives example control and data cards for an event and a time-lag analysis. The parameters are set up to analyze the data from Table 1, producing the summary values in Table 2. Running this data set should provide a test data set for checking the program once it has been compiled on an appropriate computer.

For the convenience of users, the FORTRAN program for the lag sequential analysis is presented in the Appendix of this paper.

REFERENCES

- ALTMANN, S. A. Sociobiology of rhesus monkeys. II: Stochastics of social communication. *Journal of Theoretical Biology*, 1965, 8, 490-522.
- BAKEMAN, R. Untangling streams of behavior. In G. P. Sackett (Ed.), *Observing behavior. Data collection and analysis methods* (Vol. 2). Baltimore: University Park Press, 1978.
- BAKEMAN, R., & BROWN, J. V. Behavioral dialogues: An approach to the assessment of mother-infant interaction. *Child Development*, 1977, 48, 195-203.
- GOTTMAN, J. M., & NOTARIUS, C. Sequential analysis of observational data using Markov chains. In T. Kratochwill (Ed.), *Strategies to evaluate change in single subject research*. New York: Academic Press, 1976.
- SACKETT, G. P. The lag sequential analysis of contingency and cyclicity in behavioral interaction research. In J. D. Osofsky (Ed.), *Handbook of infant development*. New York: Wiley, in press.

Appendix FORTRAN Program for Lag Sequential Analysis

```

PROGRAM LAGS (INPUT, OUTPUT, DATA, SCRATCH, TAPE3 = SCRATCH,
$  TAPE4 = DATA, TAPE5 = INPUT, TAPE6 = OUTPUT)
C  ****SACKETT****  CDMR CENTER UNIVERSITY OF WASHINGTON
C

```

```

C THIS PROGRAM WAS WRITTEN BY CHARLES CROWLEY, COMPUTER SCIENCE DEPT.,
C UNIVERSITY OF WASHINGTON, SEATTLE, WA. 98195 (206) 543-7798 OR (206)
C 543-2697.
C
C THIS IS REVISION B OF 14 FEB 1975
C REVISION A WAS OF NOV 1974
C THE ORIGINAL PROGRAM WAS OF OCT 1974
C
C THIS REVISION B INCLUDES CHANGES TO ALLOW TWO OR MORE OBSERVATIONS
C PER CARD AND TO WRITE THE DATA ON A BINARY SCRATCH FILE SO THAT RUNS
C WITH TWO OR MORE CRITERIA WILL BE CONSIDERABLY FASTER
C
C THIS PROGRAM COMPUTES LAG STATISTICS FOR TIME OR EVENT LAG
C ANALYSIS. THE SUBROUTINE *NEXT* READS THE INPUT. ANALYSES OF
C BEHAVIORS WITH DURATIONS ARE TRANSFORMED TO UNIT TIME OBSERVATION
C TYPES BY GENERATING *N* UNIT TIME DELAY OBSERVATIONS FOR A BEHAVIOR OF
C DURATION *N* TIME UNITS. *CRHIST* WILL RECORD THE HISTORY OF THE LAST
C *MAXLAG* TIME DELAYS AND WILL INDICATE WHEN DURING THAT PERIOD THAT
C THE CRITERION BEHAVIOR OCCURRED. AS WE ENCOUNTER EACH BEHAVIOR
C OBSERVATION WE GO OVER THE PAST *MAXLAG* TIME UNITS AND RECORD LAGS
C FOR ALL OCCURANCES OF THE CRITERION BEHAVIOR. THE SUBROUTINE *INIT*
C WILL READ THE CONTROL INFORMATION, SET UP THE CONTROL VARIABLES AND
C INITIALIZE THE OTHER VARIABLES
C
      REAL EXPECT, OBSERV, SD, Z, HICI, LOWCI
      INTEGER LASTCR, MAXLAG, TRIALS, ENDCOD, TARGET, NCODES, LAGTYP,
      $ ANALTY, INTYPE, CODES(21), FREQ( 800, 21), TIME,
      $ STEPSZ, NSTEPS, NEXTCR, NEXTBE, CRIT, BEH, TOTCR, TOTIME,
      $ FREQUENCY, CODE, LASTBE, TMLEFT, CUMTYP, LOWBD, HIBD, EOFMRK,
      $ NUMREC, ECHO, UNITNO, TITLE(20), COLS(32), NCARDS, PARTCD,
      $ CRDCOL
      LOGICAL ENDFIL, CRHIST(1000), ADJLG1, ENDTRL, TIME1
      COMMON LASTCR, MAXLAG, TRIALS, ENDCOD, TARGET, NCODES, LAGTYP,
      $ ANALTY, INTYPE, CODES, FREQ, ENDFIL, TIME, STEPSZ,
      $ NSTEPS, TMLEFT, CUMTYP, EOFMRK, NUMREC, ECHO, UNITNO, TITLE,
      $ COLS, NCARDS, ENDTRL, PARTCD, CRDCOL, TIME1
C
C INIT WILL SET UP MAXLAG, ANALTY, INTYPE, LAGTYP, STEPSZ, NSTEPS,
C TARGET (THE CODE OF THE CRITERION BEHAVIOR), NCODES (THE NUMBER OF
C CODES IN THE ARRAY CODES) AND CODES.
C
      5  CALL INIT
C
C WE WILL ALWAYS BE READING ONE TIME UNIT AHEAD SO THAT THE LEADING EDGE
C AND TRAILING EDGE LAGS CAN BE DONE CORRECTLY
C
      CALL NEXT(NEXTCR, NEXTBE)
C
C NOW FALSE OUT THE HISTORY ARRAY (CRHIST) IMPLYING THAT NO CRITERION
C BEHAVIORS HAVE BEEN FOUND IN THE LAST *MAXLAG* TIME UNITS (THESE ARE
C FICTITIOUS TIMES PUT IN TO SIMPLY THE PROGRAM, THEY WILL NOT AFFECT
C THE STATISTICS IN ANY WAY)
C
      10  DO 20 I = 1, MAXLAG
           CRHIST(I) = .FALSE.
      20  CONTINUE
           IHIT = 1
           LASTCR = -1
C
C NOW GO THROUGH THE BEHAVIORS ONE TIME UNIT AT A TIME
C THIS IS THE MAIN COMPUTING LOOP
C
      30  CONTINUE
           CRIT = NEXTCR
           BEH = NEXTBE
           CALL NEXT(NEXTCR, NEXTBE)
C
C IF CRIT = -2 THEN THE END OF THE DATA HAS BEEN REACHED SO WE GO TO
C PRINT THE RESULTS
C
           IF( CRIT .EQ. -2 ) GO TO 500
C
C CHECK IF THIS IS AN END OF TRIAL AND IF IT IS RESET THE CRITHIST ARRAY
C AND START OVER FRESH WITH THE NEXT TRIAL
C
           IF( CRIT .EQ. -1 ) GO TO 10
C
C COUNT UP THE RAW FREQUENCY FOR THIS BEHAVIOR
C
           TIME = TIME + 1
           FREQ(1, BEH) = FREQ(1, BEH) + 1

```

```

C
C GO THROUGH THE LAST *MAXLAG* TIME DELAYS LOOKING FOR AN OCCURRENCE
C THE TARGET CRITERION (THE COUNTING IS BACKWARDS FROM IHIT)
C
      J = IHIT
      DO 40 I = 1, MAXLAG
        J = J - 1
        IF( J .EQ. 0 ) J = MAXLAG
        IF( CRHIST(J) ) FREQ(I+1, BEH) = FREQ(I+1, BEH) + 1
40      CONTINUE
C
C NOW DECIDE WHETHER THIS IS THE TARGET CRITERION
C
      CRHIST(IHIT) = .FALSE.
      IF( CRIT .NE. TARGET ) GO TO 100
      GO TO (50, 70, 60), LAGTYP
C
C LAGTYPE = 1 - LAG FROM THE BEGINNING OF A BEHAVIOR ONLY
C
      50      IF( CRIT .EQ. LASTCR ) GO TO 100
      GO TO 70
C
C LAGTYPE = 3 - LAG FROM THE TERMINATION OF A BEHAVIOR ONLY
C
      60      IF( CRIT .EQ. NEXTCR ) GO TO 100
      GO TO 70
C
C LAGTYPE = 2 - LAG FROM EVERY OCCURRENCE OF A BEHAVIOR (I.E. FROM EVERY
C TIME UNIT WITHIN A DURATION OF THE BEHAVIOR)
C
      70      CRHIST(IHIT) = .TRUE.
C
      100     LASTCR = CRIT
C
C IHIT CYCLES CIRCULARLY FROM 1 TO MAXLAG
C
      IHIT = MOD(IHIT, MAXLAG) + 1
      GO TO 30
C
C OUTPUT RESULTS
C
500    LINES = 60
      IF( TIME1 ) WRITE(6, 505) NUMREC
505    FORMAT(1H , 17, 18H DATA RECORDS READ)
510    FORMAT(1H1, 20A4, /, 36H LAG CODE FREQ NCRIT TOTAL OBSERVED,
      $ 53H EXPECTED STANDEV 2 95 PCT CONFIDENCE INTERVAL, /)
      MLPL1 = MAXLAG + 1
      NCPL1 = NCODES + 1
      IF( FREQ(1, NCPL1) .EQ. 0 ) NCPL1 = NCODES
C
C IN THIS LOOP *LAG* IS REALLY ONE MORE THAN THE REAL LAG. THE FIRST
C ROW OF THE *FREQ* ARRAY IS THE FREQUENCY FOR THE BEHAVIOR CODE
C
      DO 550 LAG = 1, MLPL1, STEPSZ
C
C CUMULATE FREQUENCIES IN STEPS OF STEPSZ IF INDICATED
C
      IF( STEPSZ .EQ. 1 .OR. LAG .EQ. 1 .OR. CUMTYP .EQ. 0 )
      $      GO TO 518
      LOWBD = LAG - STEPSZ + 1
      HIBD = LAG - 1
      DO 516 I = 1, NCPL1
        DO 514 J = LOWBD, HIBD
          FREQ(LAG, I) = FREQ(LAG, I) + FREQ(J, I)
514      CONTINUE
516    CONTINUE
C
C ADJUST THE LAG 1 EXPECTED VALUES IF A BEHAVIOR CANNOT FOLLOW ITSELF
C
518    ADJLG1 = .FALSE.
      IF( LAG .NE. 2 ) GO TO 519
      IF( LAGTYP .NE. 3 .OR. INTYPE .GT. 2 ) GO TO 519
      ADJLG1 = .TRUE.
      NCR = 0
      DO 5181 I = 1, NCODES
        IF( TARGET .NE. CODES(I) ) GO TO 5181
        NCR = I
        GO TO 519
5181   CONTINUE
      IF( NCR .EQ. 0 ) ADJLG1 = .FALSE.

```

```

519      TOTCR = 0
      DO 520 I = 1, NCPL1
          TOTCR = TOTCR + FREQ(LAG, I)
520      CONTINUE
      DO 540 CODE = 1, NCPL1
          TOTIME = TIME - (LAG - 1) * TRIALS
          IF( TOTIME .GT. 0 ) GO TO 524
          WRITE(6, 526)
          FORMAT(37H LAGS TOO LARGE FOR THE DATA SAMPLE, ,
$          22H THIS ANALYSIS ABORTED)
          GO TO 560
524      FREQCY = FREQ(LAG, CODE)
          EXPECT = FLOAT(FREQ(1, CODE)) / FLOAT(TIME)
          IF( .NOT. ADJLG1 ) GO TO 527
          EXPECT = FLOAT(FREQ(1, CODE))
$          / FLOAT(TIME - FREQ(1, NCR))
          IF( NCR .EQ. CODE ) EXPECT = 0.
527      SD = SQRT((EXPECT * (1 - EXPECT)) / FLOAT(TOTIME))
          OBSERV = 0.
          IF( TOTCR .NE. 0 ) OBSERV = FLOAT(FREQCY) /
$          FLOAT(TOTCR)
          LINES = LINES + 1
          IF( LINES .LE. 55 ) GO TO 528
          LINES = 1
          WRITE(6, 510) (TITLE(I), I = 1, 20)
528      LAGM1 = LAG - 1
          Z = 0.
          IF( SD .NE. 0. ) Z = (OBSERV - EXPECT) / SD
          HICI = EXPECT + 1.96 * SD
          LOWCI = EXPECT - 1.96 * SD
          WRITE(6, 530) LAGM1, CODES(CODE), FREQCY, TOTCR,
$          TOTIME, OBSERV, EXPECT, SD, Z, HICI, LOWCI
530      FORMAT(315,I6,I6,F7.3,2F9.3,2F9.3 ,3H TO,F6.3)
540      CONTINUE
          WRITE(6, 545)
          FORMAT(1H )
545      LINES = LINES + 1
550      CONTINUE
560      GO TO 5
      END
      SUBROUTINE NEXT(CRIT, BEH)
C      ****SACKETT**** CDMR CENTER UNIVERSITY OF WASHINGTON
C
C THIS ROUTINE READS THE INPUT DATA. FOUR VARIABLES MUST BE SPECIFIED
C BY EACH READ.. THE CRITERION CODE (CRIT), THE BEHAVIOR CODE (BEH), THE
C END-OF-TRIAL CODE (EOT) AND THE DURATION (DUR) OF THE OBSERVATION)
C IF EVENT SEQUENTIAL ANALYSIS IS SPECIFIED NO DURATIONS ARE GIVEN.
C THESE VARIABLES ARE READ FROM CARD COLUMNS (ONE TO SEVEN) WHICH MAY BE
C ANYWHERE ON UP TO THREE CARDS. THE USER SPECIFIES THE COLUMNS WHICH
C MAKE UP HIS FIELDS. FOR EXAMPLE, IF CRIT IS IN COLUMNS 2, 5, 55 AND
C 7 THEN CRIT = 1000*C2+100*C5+10*C55+C7.
C
      INTEGER LASTCR, MAXLAG, TRIALS, ENDCOD, TARGET, NCODES, LACTYP,
$      ANALTY, INTYPE, CODES(21), FREQ( 800, 21), TIME,
$      STEPSZ, NSTEPS, CRIT, CRITER, TMLEFT, BEH, BEHAV, DELAY, EOT,
$      CODE, PREVCR, CUMTYP, EOFMRK, NUMREC, ECHO, UNITNO, TITLE(20),
$      COLS(32), NCARDS, ZERO, INCOLS(80, 3), KIN(4), LIN(4),
$      PARTCD, CRDCOL
      LOGICAL ENDFIL, ENDTRL, TIME1
      COMMON LASTCR, MAXLAG, TRIALS, ENDCOD, TARGET, NCODES, LACTYP,
$      ANALTY, INTYPE, CODES, FREQ, ENDFIL, TIME, STEPSZ,
$      NSTEPS, TMLEFT, CUMTYP, EOFMRK, NUMREC, ECHO, UNITNO, TITLE,
$      COLS, NCARDS, ENDTRL, PARTCD, CRDCOL, TIME1
      DATA LIN / 1, 9, 17, 25 /
      DATA ZERO / IRO /
C
      IF( .NOT. ENDFIL ) GO TO 5
      CRIT = -2
      RETURN
C
C IF TIME SEQUENTIAL ANALYSIS IS SPECIFIED (I.E. DURATIONS ARE PROVIDED)
C THE SAME CODES WILL BE RETURNED FOR THE DURATION OF THE BEHAVIOR.
C THE VARIABLE *TMLEFT* COUNTS DOWN THE DURATIONS.
C
      5 IF( TMLEFT .NE. 0 ) GO TO 200
C
C A NEW RECORD MUST BE READ
C
      IF( .NOT. ENDTRL ) GO TO 8.
      ENDTRL = .FALSE.

```



```

      GO TO 15
8      IF( TIME1 ) GOTO 9
C      READ FROM SCRATCH FILE ON SECOND AND LATER PASSES
      READ(3) CRIT, BEHAV, EOT, DELAY
      GOTO 70
9      IF( CRDCOL .LE. 80 ) GOTO 16
C      ELSE THIS CARD IS FINISHED SO READ THE NEXT
      CRDCOL = 1
      NUMREC = NUMREC + 1
      READ(UNITNO, 10) ((INCOLS(I, J), I = 1, 80), J = 1, NCARDS)
10     FORMAT(80I1)
      IF( EOF, UNITNO ) 14, 16
14     ENDFIL = .TRUE.
      IF( PREVCR .NE. -1 ) TRIALS = TRIALS + 1
15     CRIT = -1
      PREVCR = -1
      RETURN
C
C READ THE CARD AND ASSEMBLE THE THREE OR FOUR FIELDS FROM THE COLUMN
C NUMBERS LISTED IN *COLS*. *LIN* IS USED TO SET THE INICES FOR GOING
C THROUGH *COLS* 8 AT A TIME STARTING WITH 1, 9, 17 AND 25. *ILOWBD*
C AND *IHIBD* INDEX THROUGH *COLS*. NOTE ALL COLUMN NUMBERS ARE
C RELATIVE TO *CRDCOL*. IF MORE THAN ONE OBSERVATION PER CARD IS
C SPECIFIED. BLANKS AND ILLEGAL CHARACTERS IN A COLUMN ARE READ AS
C ZEROS. *KSIGN* KEEPS TRACK OF THE SIGN OF THE INPUT NUMBER
C
16     IUPPER = 4
      IF( ANALTY .EQ. 1 ) IUPPER = 3
      DO 60 I = 1, IUPPER
        L = LIN(I)
        KIN(I) = 0
        ILOWBD = L + 1
        IHIBD = L + COLS(L)
        KSIGN = 1
        DO 40 J = ILOWBD, IHIBD
          ISUB = COLS(J) + CRDCOL - 1
          IDIGIT = INCOLS(ISUB) - ZERO
C          CHECK FOR BLANKS
          IF( IDIGIT .EQ. 18 ) IDIGIT = 0
C          CHECK FOR MINUS SIGNS
          IF( IDIGIT .NE. 11 ) GOTO 18
          KSIGN = - KSIGN
          GOTO 40
18          IF( IDIGIT.GE.0 .AND. IDIGIT.LE.9 ) GO TO 30
          WRITE(6, 20) INCOLS(ISUB), ISUB, NUMREC
C
20     FORMAT(20H ILLEGAL CHARACTER (, R1, 12H) IN COLUMN , 13,
      $ 11H IN RECORD , 17, 18H - CHANGED TO ZERO)
C
          IDIGIT = 0
          KIN(I) = 10 * KIN(I) + IDIGIT
40     CONTINUE
      KIN(I) = KSIGN * KIN(I)
60     CONTINUE
      CRDCOL = CRDCOL + PARTCD
      CRIT = KIN(1)
      BEHAV = KIN(2)
      EOT = KIN(3)
      DELAY = KIN(4)
      IF( ANALTY .EQ. 1 ) DELAY = 1
      WRITE(3) CRIT, BEHAV, EOT, DELAY
70     IF( ECHO .NE. 0 ) WRITE(6, 75) CRIT, BEHAV, EOT, DELAY
75     FORMAT(1H , 19, 318)
      IF( EOT .EQ. EOFMRK ) GO TO 14
      IF( EOT .NE. ENDCOD ) GO TO 80
      IF( INTYPE .EQ. 0 ) GO TO 15
      ENDTRL = .TRUE.
80     TMLEFT = DELAY
C
200    TMLEFT = TMLEFT - 1
C
C LOOK FOR THE BEHAVIOR CODE IN THE CODE TABLE, IF IT IS NOT THERE
C ASSIGN IT A CODE OF *NCODES+1* WHICH IS THE ALL-THE-REST CODE
C
      BEH = NCODES + 1
      DO 210 CODE = 1, NCODES
        IF( BEHAV .NE. CODES(CODE) ) GO TO 210
        BEH = CODE
        GO TO 220
210    CONTINUE

```

```

C
220  PREVCR = 0
      RETURN
      END
      SUBROUTINE INIT
C      ****SACKETT**** CDMR CENTER UNIVERSITY OF WASHINGTON
C
C THIS ROUTINE READS THE CONTROL INFORMATION AND THE INPUT FORMAT FOR
C THE DATA. THE FIRST CARD OF A SET OF CONTROL INFORMATION CARDS IS THE
C TITLE CARD AND CAN BE ANY 80 CHARACTERS. IT WILL BE PRINTED ON EVERY
C PAGE OF THE OUTPUT. THE NEXT CARD OF CONTROL INFORMATION IS
C ARRANGED AS FOLLOWS..
C
C COLUMNS FIELD
C 1 TYPE OF ANALYSIS
C     1 - EVENT
C     2 - TIME DURATION
C
C 2 TYPE OF INPUT - IF ZERO OR BLANK THEN A RECORD WITH AN EOT MARK
C IS NOT COUNTED AS AN OBSERVATION. IF THIS IS NOT ZERO OR BLANK
C THEN OBSERVATION WITH EOT MARKS WILL BE COUNTED AS OBSERVATIONS
C
C 3 TYPE OF LAG
C     1 - LAG FROM THE BEGINNING OF A BEHAVIOR ONLY
C     2 - LAG FROM EACH OCCURANCE OF A BEHAVIOR (IN TIME)
C     3 - LAG FROM THE TERMINATION OF A BEHAVIOR ONLY
C
C 4-6 STEPSIZE - THE NUMBER OF LAGS TO SKIP BETWEEN LINES OF THE
C OUTPUT TABLE
C
C 7 LAG SUMMARIZATION. THIS IS ONLY APPLICABLE IF STEPSIZE IS
C GREATER THAN ONE. IF THIS FIELD IS ZERO EACH OUTPUT LINE WILL
C REFLECT THE REAL STATISTICS FOR THAT LAG. IF THIS FIELD IS
C ONE (1) THEN EACH LINE WILL REFLECT THE ACCUMULATED STATISTICS
C FOR ALL THE LAGS AFTER THE LAST LINE PRINTED AND UP TO AND
C INCLUDING THIS LAG.
C
C 8-11 NUMBER OF LAGS YOU ARE INTERESTED IN (IN STEPS OF STEPSIZE)
C
C 12-18 END OF TRIAL CODE (MARKER)
C
C 19-25 END OF ONE DATA SET MARKER. THIS IS USED IF YOU WANT TO
C PUT SEVERAL DATA SETS CONSECUTIVELY IN A FILE AND RUN ANALYSES
C ON EACH OF THEM. THEY ARE SEPERATED ONLY BY THIS MARKER AND
C NOT BY A FILEMARK
C
C 26 ECHO CODE - IF THIS IS 1 THE DATA WILL BE ECHOED ON THE OUTPUT
C AS IT IS BEING READ
C
C 27 IF THIS FIELD IS 1 THEN THE DATA WILL BE READ FROM THE LOGICAL
C FILE *DATA*, OTHERWISE IT WILL BE READ FROM *INPUT*.
C
C 28-29 IF THIS FIELD IS BETWEEN 1 AND 79 TWO OR MORE OBSERVATIONS PER
C CARD WILL BE READ. THIS NUMBER IS THE WIDTH OF THE FIELD.
C AS MANY FIELDS AS WILL FIT ON A CARD WILL BE READ.
C
C NEXT COME ONE TO FOUR CARDS CONTAINING THE CRITERION BEHAVIOR
C CODES. A SEPARATE ANALYSIS WILL BE RUN FOR EACH CRITERION
C SPECIFIED AND A SEPARATE TABLE PRINTED FOR EACH ONE. THE LIST
C OF CODES ARE PUT (UP TO) ELEVEN (11) ON A CARD IN SEVEN
C COLUMN WIDE FIELDS. THAT IS, THE FIRST ONE IS COL. 1-7( THE
C SECOND IS COL. 8-14 AND SO ON TO THE LAST WHICH WILL BE
C COL. 71-77. UP TO FORTY CRITERIA CAN BE SPECIFIED. THE LAST
C ONE IS INDICATED BY A DUMMY CRITERION CODE OF -1. THIS DUMMY
C CODE MUST BE PRESENT TO MARK THE END EVEN IF IT IS THE ONLY
C CODE ON THE LAST CARD
C
C THE NEXT SET OF CARDS GIVES THE CODES OF THE OBJECT BEHAVIORS
C TO BE LOOKED FOR AT THE VARIOUS LAGS. THESE ARE ENTERED IN
C EXACTLY THE SAME MANNER AS THE CRITERION CODES WERE. THAT IS,
C UP TO ELEVEN CODES PER CARD IN SEVEN COLUMN FIELDS WITH THE
C LAST CODE FOLLOWED BY A DUMMY CODE OF -1 (WHICH MUST BE PRESENT
C ). BUT THESE CARDS MUST BE SEPARATE FROM (AND MUST FOLLOW) THE
C CARDS FOR THE CRITERION CODES. THAT IS, THEY MUST START ON A
C NEW CARD AND CANNOT BE PUT IN EXTRA FIELDS ON THE LAST
C CRITERION CARD
C
C THE NEXT SET OF THREE OR FOUR CARDS GIVES THE CARD COLUMNS IN WHICH TO
C FIND.. FIRST THE CRIT CODES, SECOND THE BEH CODES, THIRD THE
C EOT CODES AND FOURTH (IF REQUIRED) THE DURATIONS. THESE FIELDS
C ARE ENTERED AS SEVEN (7) THREE COLUMN FIELDS. ONE SUCH CARD IS

```

```

C      REQUIRED FOR EACH OF.. CRIT, BEH, AND EOT (ALSO DUR IF NEEDED)
C      THE CODE IS GENERATED BY CONCATENATING THE SPECIFIED COLUMNS.
C      IF CRIT IS IN COLS. 3 AND 68 AND 23 THEN CRIT=100*C3+10*C68+C23
C      IF COLUMNS OVER 160 ARE SPECIFIED THREE CARDS WILL BE READ PER
C      /BSERVATION. IF OVER 80 BUT ALL LESS THEN 161 THEN TWO CARDS
C      WILL BE READ AND IF ALL ARE UNDER 81 THEN ONE CARDS ONLY WILL
C      BE READ. ONLY THREE CARDS PER OBSERVATION ARE ALLOWED.
C
C IF SEVERAL RUNS ARE TO BE DONE ON SEQUENTIALLY ENTERED DATA SETS
C ON A FILE THEN THIS SAME SEQUENCE OF CARDS IS REPEATED FOR EACH
C RUN AND ANY OF THE CONTROL VARIABLES CAN BE CHANGED. THE LAST
C SUCH RUN SHOULD BE FOLLOWED BY TWO CARDS EACH CONTAINING A 9
C IN COLUMN 1.
C
      INTEGER LASTCR, MAXLAG, TRIALS, ENDCOD, TARGET, NCODES, LAGTYP,
$      ANALTY, INTYPE, CODES(21), FREQ( 800, 21), TIME,
$      STEPSZ, NSTEPS, IN(11), TMLEFT, CUMTYP, CRTBL(30), ICRTBL,
$      EOFMRK, NCRTBL, NUMREC, ECHO, UNTYPE, UNITNO, TITLE(20),
$      COLS(32), NCARDS, PARTCD, CRDCOL
      LOGICAL ENDFIL, ENDTRL, TIME1
      DATA ICRTBL, NCRTBL /0, 0/
      COMMON LASTCR, MAXLAG, TRIALS, ENDCOD, TARGET, NCODES, LAGTYP,
$      ANALTY, INTYPE, CODES, FREQ, ENDFIL, TIME, STEPSZ,
$      NSTEPS, TMLEFT, CUMTYP, EOFMRK, NUMREC, ECHO, UNITNO, TITLE,
$      COLS, NCARDS, ENDTRL, PARTCD, CRDCOL, TIME1
C
C ARE WE GOING THROUGH CRTBL AND DOING AN ANALYSIS FOR EACH CRITERION
C (NCRTBL IS ONE MORE THAN THE NUMBER OF CRITERIA READ IN)
C
      WRITE(6, 80)
80      FORMAT(1H1)
      IF( ICRTBL .LT. NCRTBL ) GO TO 205
      TIME1 = .TRUE.
      REWIND 3
      READ(5, 90) (TITLE(I), I = 1, 20)
90      FORMAT(20A4)
      READ(5, 100) ANALTY, INTYPE, LAGTYP, STEPSZ, CUMTYP, NSTEPS,
$      ENDCOD, EOFMRK, ECHO, UNTYPE, PARTCD
100     FORMAT(3I1, I3, I1, I4, 2I7, 2I1, I2)
      IF( ANALTY .EQ. 9 ) STOP
      UNITNO = 5
      IF( UNTYPE .NE. 0 ) UNITNO = 4
      IF( ANALTY .EQ. 1 .OR. ANALTY .EQ. 2 ) GO TO 104
      WRITE(6, 101)
101     FORMAT(42H ANALYSIS TYPE OUT OF RANGE - CHANGED TO 1)
      ANALTY = 1
104     IF( LAGTYP .GE. 1 .AND. LAGTYP .LE. 3 ) GO TO 106
      WRITE(6, 105)
105     FORMAT(37H LAG TYPE OUT OF RANGE - CHANGED TO 2)
      LAGTYP = 2
106     IF( STEPSZ .GT. 0 ) GO TO 108
      WRITE(6, 107)
107     FORMAT(34H STEPSIZE BELOW ONE - CHANGED TO 1)
      STEPSZ = 1
108     IF( PARTCD .LE. 0 .OR. PARTCD .GT. 80 ) PARTCD = 80
C
C READ THE CRITERION LIST
C
120     NCRTBL = 1
      ICRTBL = 1
130     READ(5, 140) (IN(I), I =1, 11)
140     FORMAT(11I7)
      DO 150 I = 1, 11
      IF( IN(I) .EQ. -1 ) GO TO 160
      CRTBL(NCRTBL) = IN(I)
      NCRTBL = NCRTBL + 1
      IF( NCRTBL .LE. 20 ) GO TO 150
      WRITE(6, 144)
144     FORMAT(40H TOO MANY CRITERIA - ALL OVER 20 IGNORED)
      GO TO 160
150     CONTINUE
      GO TO 130
C
C READ THE CODE LIST
C
160     NCODES = 0
170     READ(5, 140) (IN(I), I =1, 11)
      DO 180 I = 1, 11
      IF( IN(I) .EQ. -1 ) GO TO 190
      NCODES = NCODES + 1
      CODES(NCODES) = IN(I)

```

```

        IF( NCODES .LE. 20 ) GO TO 180
        WRITE(6, 174)
174      FORMAT(37H TOO MANY CODES - ALL OVER 20 IGNORED)
        GO TO 190
180    CONTINUE
        GO TO 170
190    CODES(NCODES + 1) = -1
        COLS(25) = 0
        NCARDS = 1
        IHIBD = 25
        IF( ANALTY .EQ. 1 ) IHIBD = 17
        DO 196 I = 1, IHIBD, 8
            READ(5, 192) (IN(K), K = 1, 7)
192      FORMAT(7I3)
            COLS(I) = 0
            DO 194 J = 1, 7
                N = IN(J)
                IF( N .EQ. 0 ) GO TO 196
                IF( N .GT. 80 .AND. NCARDS .EQ. 1 ) NCARDS = 2
                IF( N .GT. 160 ) NCARDS = 3
                COLS(I) = COLS(I) + 1
                ISUB = I + J
                COLS(ISUB) = N
194      CONTINUE
196    CONTINUE
        GO TO 210
C
C REWIND THE DATA SET TO ANALYZE IT WITH THE NEXT CRITERION
C
205    TIME1 = .FALSE.
        REWIND 3
        ECHO = 0
C
C INITIALIZE VARIOUS STUFF
C
210    TARGET = CRTBL(ICRTBL)
        ICRTBL = ICRTBL + 1
        CRDCOL = 81
        NUMREC = 0
        MAXLAG = STEPSZ * NSTEPS
        IF( MAXLAG .LE. 800 ) GO TO 230
        WRITE(6, 220)
220      FORMAT(39H MAXIMUM LAG TOO LARGE - CHANGED TO 800)
        MAXLAG = 800
        NSTEPS = 800
        STEPSZ = 1
230    ENDFIL = .FALSE.
        ENDTRL = .FALSE.
        TRIALS = 0
        TIME = 0
        TMLEFT = 0
        NCPL1 = NCODES + 1
        MLPL1 = MAXLAG + 1
        DO 310 I = 1, MLPL1
            DO 300 J = 1, NCPL1
                FREQ(I, J) = 0
300      CONTINUE
310    CONTINUE
C
C OUTPUT A RECORD OF THE OPTIONS SELECTED
C
        WRITE(6, 400) (TITLE(I), I = 1, 20)
400      FORMAT(1H , 20A4)
        IF( ANALTY .EQ. 1 ) WRITE(6, 401)
401      FORMAT(50H ***** EVENT SEQUENTIAL ANALYSIS - REVISION *B* OF,
$ 16H 14 FEB 75 *****)
        IF( ANALTY .EQ. 2 ) WRITE(6, 402)
402      FORMAT(49H ***** TIME SEQUENTIAL ANALYSIS - REVISION *B* OF,
$ 16H 14 FEB 75 *****)
        IF( INTYPE .EQ. 0 ) WRITE(6, 411)
411      FORMAT(38H EOT CARDS NOT COUNTED AS OBSERVATIONS)
        IF( INTYPE .NE. 0 ) WRITE(6, 412)
412      FORMAT(34H EOT CARDS COUNTED AS OBSERVATIONS)
        IF( LACTYP .EQ. 1 ) WRITE(6, 421)
421      FORMAT(33H LAG FROM ONSET OF CRITERION ONLY)
        IF( LACTYP .EQ. 2 ) WRITE(6, 422)
422      FORMAT(42H LAG FROM ALL OCCURRANCES OF THE CRITERION)
        IF( LACTYP .EQ. 3 ) WRITE(6, 423)
423      FORMAT(38H LAG FROM OFFSET OF THE CRITERION ONLY)
        WRITE(6, 430) NSTEPS, STEPSZ

```

```

430  FORMAT(5H LAG , I4, 16H STEPS OF SIZE , I3)
      IF( CUMTYP .NE. 0 ) WRITE(6, 440) STEPSZ
440  FORMAT(21H ACCUMULATE PREVIOUS , I3, 22H STEPS PER OUTPUT LINE)
      WRITE(6, 450) ENDCOD
450  FORMAT(23H END OF TRIAL CODE = , I7)
      WRITE(6, 460) EOFMRK
460  FORMAT(23H END OF SESSION CODE = , I7)
      WRITE(6, 470) TARGET
470  FORMAT(23H CRITERION SELECTED = , I7)
      WRITE(6, 478)
478  FORMAT(25H LIST OF CODES SELECTED..)
      WRITE(6, 480) (CODES(I), I = 1, NCODES)
480  FORMAT(1H , 11I7)
      IF( UNTYPE .EQ. 0 ) WRITE(6, 481)
481  FORMAT(51H INPUT FOLLOWS CONTROL INFORMATION CARDS ON *INPUT*)
      IF( UNTYPE .NE. 0 ) WRITE(6, 482)
482  FORMAT(27H INPUT FOUND ON FILE *DATA*)
      N = COLS(1) + 1
      WRITE(6, 200) (COLS(I), I = 2, N)
      N = COLS(9) + 9
      WRITE(6, 201) (COLS(I), I = 10, N)
      N = COLS(17) + 17
      WRITE(6, 202) (COLS(I), I = 18, N)
      N = COLS(25) + 25
      IF( COLS(25) .NE. 0 ) WRITE(6, 203) (COLS(I), I = 26, N)
      N = (PARTCD + 79) / PARTCD
      IF( PARTCD .LT. 80 ) WRITE(6, 250) N, PARTCD
250  FORMAT(I3, 28H OBSERVATIONS PER CARDS WITH, I3,
      $ 24H COLUMNS PER OBSERVATION)
200  FORMAT(24H CRITERION COLUMNS = , 7I4)
201  FORMAT(24H CODE COLUMNS = , 7I4)
202  FORMAT(24H END OF TRIAL COLUMNS = , 7I4)
203  FORMAT(24H DURATION COLUMNS = , 7I4)
      IF( ECHO .NE. 0 ) WRITE(6, 490)
490  FORMAT(/, 34H CRITERION CODE EOT DELAY)
      RETURN
      END

```

(Accepted for publication December 28, 1978.)