PROGRAM ABSTRACTS/ALGORITHMS

JOINT and SEQU: FORTRAN routines for the analysis of observational data

PETER W. D. DODD

Dalhousie University, Halifax, Nova Scotia, B3H 4J1 Canada

ROGER BAKEMAN

Georgia State University, Atlanta, Georgia 30303

ROLF LOEBER

Oregon Social Learning Center, Eugene, Oregon 97401

and

STEPHEN C. WILSON

Queen's University, Kingston, Ontario K7L 3N6, Canada

In a lag analysis, the frequencies of various events that follow a target or criterion event are tallied according to the ordinal position of each event observed following the target event. For behavior analysts, lag analysis is most useful for detecting regular patterns of contingent events, since reinforcers and punishers can be both identified and evaluated by their effects on antecedent and consequent behaviors. Once contingent events have been identified it is of considerable interest to detect extended sequences of particular events or classes of events. This article describes two FORTRAN programs that can facilitate the analysis of observational data: One, JOINT, computes the simple frequencies of events and provides a lag-sequential analysis; the other, SEQU, detects predefined sequences of events. Sackett, Holm, Crowley, and Henkins (1979) have recently described a FORTRAN program, LAGS, that performs lag-sequential analysis on behavioral interaction data, and they have addressed a number of theoretical issues involved in coding observational data and in applying inferential statistics to those data.

JOINT shares many common features with LAGS and is offered here only as an alternative. There are two major differences between the two programs, nonetheless. First, in JOINT, all behaviors serve as the target behaviors in the lag analysis, whereas in LAGS only one target can be specified at a time. Second, in JOINT, the maximum lag allowed is 10, whereas LAGS can handle lags of up to 800. These two differences seem to result because JOINT passes through the data only once, but LAGS passes through the data separately for each target behavior. Thus, one program requires no a priori decisions about target behaviors, and the other allows a more extended lag analysis.

Program Input. The input to both JOINT and SEQU is in two parts, or files. The first file, a control file,

provides a general title, the numerical codes that represent the behaviors and alphanumeric labels for each, the lags to be analyzed, and an optional format statement for reading the second file, the data file. If any of the numerical codes represent more than one behavior that is coded in the data, the recoding information is given here. The second file, the data file, begins with a line of identifying information for the session and, in sequence, the numerical codes for each behavior recorded during the session. A unique code marks the end of the session, and another code marks the end of the set of sessions to be analyzed.

JOINT accepts observational data that consist of up to 80 different codes, each code being a number that represents a particular behavior. At the user's option, codes may be assigned that represent the occurrence of any 1 of up to 80 other uniquely coded behaviors. Thus, the raw data may comprise many codes, but the program can analyze only 80 codes at a time. Memory limitations on some computers may require program modifications that will reduce the number of codes per analysis to 20 or 40. The data to be analyzed may be drawn from a single session or may be combined across as many as 50 different sessions. The number of behaviors recorded in a session is limited by only the largest integer value the computer can store, a value usually so large as to pose no practical problems.

Program Output: JOINT. JOINT computes simple frequencies and probabilities and rates for each of the codes defined in the control file. JOINT also computes joint frequencies, conditional probabilities, z scores, and rates for the codes with lags from 1 to 10. The lag analysis is printed in matrix form, with the entire set of codes comprising both the row (R) and column (C) behaviors. For a lag of 1, the joint frequency of R and C, $f(R \& C)$, is the number of times Behavior C occurred immediately after Behavior R occurred. For a lag of 2, of course, $f(R \& C)$ is the number of times C was the second behavior recorded after R occurred, and so on. The conditional probability, $p(C/R)$, is the probability of the C behavior, given that the R behavior was occurring in a preceding time frame. The conditional probability of a particular R and C is obtained by dividing $f(R \& C)$ by $f(R)$. The rate of each behavior is calculated simply from the duration of the observation periods. Likewise, the rate of each joint frequency is given, that is, the rate at which a C behavior followed an R behavior at each lag. The z scores indicate the extent to which an observed joint frequency deviated from the joint frequency that would occur if Behavior R and Behavior C were independent of each other. Assuming independence (or only chance association), the expected

frequency for a joint occurrence would be $f(R)$ multiplied by $p(C)$. If the observed joint frequency is greater than expected, the z score is positive; if less than expected, it is negative.

With JOINT, the printed output can be voluminous, since an analysis of 80 codes at Lags 1-10 generates 64,000 cells. However, to facilitate visual inspection of the data, cells with z scores having probabilities less than .05 are underlined. In addition, a one-page matrix of all codes and lags displays symbols to identify cells with z scores greater or smaller than those expected by chance.

Program Output: SEQU. SEQU searches the data file for up to 40 different sequences of codes. Each sequence may contain up to 10 different codes for behaviors or classes of behaviors. Because a sequence of narrowly defined behaviors may occur only rarely in a session, grouping the observed behaviors into classes would seem a useful strategy. In order that the particular elements that compose each sequence can be identified, the program "tags" each sequence, printing the ordinal position in the input data at which a sequence was detected. The tagging provides a ready check that appropriate codes comprise each class of behaviors. SEQU computes simple frequencies and probabilities of the various sequences. A binomial test compares the difference between the probability of successive pairs of sequences as they are listed in the control file. These tests, however, have inferential validity only when the sequences that are compared differ by one behavior

or class of behaviors. SEQU was an adaptation of JOINT; the routine that obtains the next sequential behavior code in JOINT was modified to return a code only when a sequence of codes is detected. Consequently, the lag analysis available in JOINT is also available in SEQU, although such an analysis may be of dubious value and validity.

Language and Computer. JOINT is written in FORTRAN IV. It was developed on a UNIVAC 70/0 TSOS system. Extensive modifications, including the development of SEQU, were accomplished on a BURROUGHS B6700 computer. Both programs should run with minimal modifications on any computer that supports a FORTRAN system. Presently, the programs are being modified to run on an LSI-11/2 computer using the RT-11 operating system.

Availability. Listings and specific instructions on the formats of control and data files of JOINT and SEQU may be obtained for \$2 (to cover duplication and mailing charges) from Peter W. D. Dodd, Department of Psychology, Dalhousie University, Halifax, Nova Scotia, B3H 4J1 Canada.

REFERENCE

- SACKETT, G. P., HOLM, R., CROWLEY, C., & HENKINS, A. A FORTRAN program for lag sequential analysis of contingency and cyclicity in behavioral interaction data. *Behavior Research Methods & Instrumentation*, 1979, 11, 366-378.

(Accepted for publication May 29, 1981.)