PROGRAM ABSTRACTS/ALGORITHMS

BASIC rank-order program

KATHLEEN A. McNALLY and D. LOUIS WOOD
University of Arkansas, Little Rock, Arkansas 72204

Psychologists doing rank-order statistics face the problem of assigning ranks to data in a manner that computes ties and maintains the ordinal sequence of the data for entrance into Spearman's rho and similar statistics. For the microcomputer owner who is seeking a sorting routine to write a statistics program, most available programs perform a "bubble sort," which reorganizes input data into a logical sequence. This sorting routine is primarily used for alphabetizing, but it can also be used in other applications (such as arranging scores in numerical order).

Using bubble-sort routines in rank-order statistics is time-consuming, since each score is repeatedly compared to each other score, and long and cumbersome. Additional disadvantages are that these routines end with the scores in an order different from that required, and ties are not resolved. (When the sort is completed, it is left to the user to assign ranks to the sorted scores and, presumably, to deal with any ties.)

Table 1 shows a rank-order statistics program written for a TRS-80 Model III microcomputer in Level II BASIC; the program can be easily modified for any BASIC that uses arrays. This program, unlike a bubble-sort routine, will (1) maintain the relative position of each score in a list, (2) assign ranks to each score, assigning the appropriate rank to any number of tie scores, and (3) print the frequency with which each score occurs. It is a short program, utilizing very little core, and it is relatively fast. It utilizes a two-dimensional (N by 3) array. The first column of the array is used to store the scores to be ranked. The second column is used to store the rank of each score, and the third column stores the frequency of occurrence of each score and is used to resolve ties.

When the program is run, it asks for the number of scores to be ranked (N) and uses N to initialize the N by 3 array. The maximum size of N is limited only by the available random-access memory (RAM). The program then asks for each score to be input and enters the scores in the first column of the array. Line 40 determines the ranks, and Lines 50 and 60 resolve ties. Lines 70 and 80 simply print each score, its corresponding rank, and its frequency of occurrence.

Table 1
BASIC Rank-Order Program Listing

```

10 CLS : INPUT "ENTER N"; N
15 DIM X(N,3)
20 FOR I = 1 TO N : INPUT "ENTER X"; A : X(I,1) = A : NEXT
30 FOR J = 1 TO N : X(J,2) = 1 : X(J,3) = 0 : NEXT
40 FOR I = 1 TO N : FOR J = 1 TO N : IF X(I,1) > X(J,1) THEN
    X(I,2) = X(I,2)+1
45 NEXT J,I
50 FOR I = 1 TO N : FOR J = 1 TO N : IF X(I,1) = X(J,1) THEN
    X(I,3) = X(I,3)+1
55 NEXT J,I
60 FOR J = 1 TO N : X(J,2) = X(J,2) + (X(J,3)-1)/2 : NEXT
70 PRINT "SCORE","RANK","FREQUENCY",
80 FOR J = 1 TO N : PRINT X(J,1),X(J,2),X(J,3) : NEXT : END

```

(Accepted for publication September 24, 1982.)