PROGRAM ABSTRACTS/ALGORITHMS

A BASIC program to produce random samples without replacement

S. A. ROBERTSON

Manchester Polytechnic, Manchester, England M15 5FG

A recent paper by Scheirer (1976) presents a subroutine designed to solve the problem of randomly selecting from a set of objects (usually letters or digits) without replacement, that is, such that each object appears at most once. Such a problem is precipitated, for instance, when a specific number of trials must be block randomized in an experiment. Winer (1971) presents random orderings of the numbers 1 to 16, although usually random number tables are not easily adapted to the resolution of this kind of difficulty.

The subroutine presented by Scheirer is written in INTERACTIVE BASIC and randomly orders the integers 1 to T in the Dimension X(I). To define the nth member of the array, the program randomly generates an integer P such that $1 \leq P \leq T$, and P then becomes the nth array member if and only if it is not included in the first n - 1 numbers. Although effective, this procedure can be expensive in terms of processing time if T is large, since random numbers that are repeated must be rejected by the routine as they refer to established members of the new array. As n approaches T this problem tends to increase, as the number of such repetitions rises rapidly.

A slightly different procedure is more efficient. To illustrate, suppose we wish to randomly sample (without replacement) five numbers from the set 1 to 8. These numbers might be stored in Array X(I) as follows:

Storage Location:	X(0)	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)
Contents:	1	2	3	4	5	6	7	8

If a random number is generated between 0 and 7 (say 3), we then assign the contents of X(3) to A(0) in Dimension A(J). Instead of now generating a fresh random number (possibly duplicating 3), we shift the contents of X(4), X(5), X(6), and X(7) down Array X(I) by one step, so as to produce the following situation:

Storage Location:	X(0)	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)
Contents:	1	2	3	5	6	7	8	8

The value of X(7) remains unchanged in the assignment of its value to X(6). A random number now generated between 0 and 6 can be used to define the second member of the final Array A(J). If this, for instance, is 4,

the value 6 will be assigned to A(1) and X(5) and X(6) may then be moved one further step down Array X(I). Note also that we can ignore the contents of X(7) at this stage.

Repetition of this procedure (reducing the range of the random number generation by one each time) produces an appropriate set of numbers in Array A(J), and, when five have been selected, the procedure may be terminated. In this manner, a stepwise condensation of the population of objects sampled is effected such that repetition of random numbers still allows members to be selected for the final Array A(J).

A BASIC Subroutine. The program presented is able to perform tasks of the sort described above. The routine randomly samples (without replacement) L2 integers from the set 1 to L1, stores these in Array Locations A(0) to A(L2 - 1) and prints their values. The process is repeated N times. As a result, $L1 \geq L2$. To solve the problem described above, it is only necessary to input the values 8, 5, and 1 when Line 20 is executed. More generally, to randomly order the integers 1 to M 10 times, the values M, M, and 10 are input. As M and N grow larger, the advantages of the routine in terms of processing time become much more pronounced. In the situation where $L1 \gg L2$, some other procedure might well prove more efficient.

Table 1
Program to Sample Integers Without Replacement

```

10 DIM X(100), A(100)
20 INPUT L1, L2, N
30 FOR G = 1 TO N
40 FOR I = 0 TO L1 - 1
50 X(I) = I + 1
60 NEXT I
70 C = 0
80 D = L1 - 1
90 F = INT((D + 1)*RND(0))
100 A(C) = X(F)
110 C = C + 1
120 IF F = D THEN 160
130 X(F) = X(F + 1)
140 F = F + 1
150 GOTO 120
160 D = D - 1
170 IF C = L2 THEN 190
180 GOTO 90
190 FOR P = 0 TO L2 - 1
200 PRINT A(P);
210 NEXT P
220 PRINT
230 NEXT G
240 STOP
250 END

```

The program was developed on an ICL 1902S computer and would be relatively simple to convert to FORTRAN. More interesting applications of the subroutine to the randomizing of nonnumeric materials have been attempted successfully, for instance, generating random blocks of letters for iconic memory experimentation similar to those used by Holding (1973). Details of these are available from the author.

Availability. Requests for sample output and listings of the subroutine applications programs should be sent to S. A. Robertson, School of Psychology, Manchester Polytechnic, Gaythorn Annexe, River Street, Manchester, England M15 5FG.

REFERENCES

- HOLDING, D. H. Recognition tests of visual information storage. *British Journal of Psychology*, 1973, **64**, 9-16.
- SCHEIRER, C. J. Routine to provide a random order for counter-balanced variables. *Behavior Research Methods & Instrumentation*, 1976, **8**, 468.
- WINER, B. J. *Statistical principles in experimental design*. New York: McGraw-Hill, 1971.

(Received for publication April 12, 1977.)