A Convenient Algorithm for Drawing a Simple Random Sample

A. I. McLeod, D. R. Bellhouse

Applied Statistics, Volume 32, Issue 2 (1983), 182-184.

Stable URL:
http://links.jstor.org/sici?sici=0035-9254%281983%2932%3A2%3C182%3AACAFDA%3E2.0.CO%3B2-U

Appl. Statist. (1983),
32, *No.* 2, *pp.* 182–184

Miscellanea

A Convenient Algorithm for Drawing a Simple Random Sample

By A. I. MCLEOD and D. R. BELLHOUSE

The University of Western Ontario, Canada

[Received November 1982]

SUMMARY

A convenient one-pass algorithm for drawing a simple random sample without replacement of size n from a population of N members, when N is initially unknown, is presented. Moreover, even when N is known, this algorithm appears to be more efficient than previously suggested algorithms when the entire population is stored in the fast memory of the computer. Applications to sampling from a computer file and to linear programming are briefly indicated.

Keywords: SAMPLING; WITHOUT REPLACEMENT; SIMPLEX METHOD FOR LINEAR PROGRAMMING

1. INTRODUCTION

In many applications, such as sampling from a large computer file and computer simulation studies, it is useful to be able to obtain a simple random sample without replacement in one pass through the computer file. For the case in which the population size N is known, algorithms have been given by Fan *et al.*, (1962), Bebbington (1975) and Ryan *et al.*, (1981). The case in which N is unknown is also of practical importance. For example, the population values of interest may be some subset of the items in the computer file. To use the previous algorithms, it would be necessary to make a preliminary pass through the file to obtain the appropriate population. We provide a one-pass algorithm for this latter situation. Furthermore, even when N is known, the algorithm is shown empirically to be more efficient than the Fan-Bebbington and Minitab (Ryan *et al.*, 1962) algorithms with respect to computer time. Fan *et al.*, (1962) also gave a one-pass algorithm for obtaining a simple random sample without replacement when N is unknown, but the algorithm presented here is much simpler.

Another application of our algorithm in the simplex method for linear programming is described in Section 4.

2. THE SAMPLING METHOD AND ALGORITHM

The sampling method is very simply described. Suppose a sample of size n is desired from a sequentially ordered population. Select the first n members of the population as the initial sample. Then proceed sequentially through the rest of the population. Each time a new population unit is encountered, the sample is updated, either by not including the new unit, or by including the unit and deleting one of the current sample units at random. When the kth population unit is encountered, $k = n+1, \ldots, N$, the updated sample remains the same with probability $1-(n/k)$ and the unit k replaces one of the current sample members with probability n/k. Again, when a

Present address: Department of Statistical and Actuarial Sciences, The University of Western Ontario, London, Ontario, Canada NGA 5B9

 0035–9254/83/32182 $2.00

replacement is made, the sample member deleted is chosen at random from among the n members of the current sample.

It can be shown by induction that at any step in the sampling process, the updated sample is a simple random sample without replacement. Hence, the final sample is as required. Let $P_{k,n}$ be the probability that any particular sample is chosen at the kth step of the sampling process, $k = n, \ldots, N$. The first n steps correspond to obtaining the initial sample. The initial step $k = n$ corresponds to selecting the first n members of the population so that $P_{n,n} = 1$. Now assume $P_{k,n} = 1/\binom{k}{n}$ for $k > n$. Then when the $(k+1)$th member of the population is encountered, either:

(a) the updated sample does not contain the $(k+1)$th member, or

(b) the lth population member (for some $l < k+1$) which was present in the current sample is replaced by the $(k+1)$th member.

When (a) occurs, the probability of the updated sample is

$$P_{k+1,n} = P_{k,n}\,[1 - n/(k+1)] = 1 \Big/ \binom{k+1}{n}.$$

When (b) occurs, the probability that the lth member was chosen from among the n sample members is $1/n$. The updated sample could also be obtained if the lth member was replaced by any of the other $k-n$ population members not in the current sample. Since each of these possible samples has probability $P_{k,n}$, the unconditional probability of the updated sample is

$$P_{k+1,n} = P_{k,n}\,[n/(k+1)]\,[(1/n) + (k-n)/n] = 1 \Big/ \binom{k+1}{n}.$$

The result follows by induction.

The algorithm given below is derived from this sampling method. At the beginning of this algorithm, the population size N is unknown. At the conclusion of the algorithm, N is obtained and a simple random sample without replacement of size n has been obtained, provided that $n \leqslant N$.

1. Set $k \leftarrow 0$.
2. Are there any remaining members in the population? If no, terminate. Otherwise, obtain the next member and set $k \leftarrow k + 1$.
3. (a) If $k \leqslant n$, the kth population member becomes the kth member of the sample. Go to Step 2.
   (b) If $k > n$, generate a uniform $(0, 1)$ random variable U and set $j \leftarrow 1 + [Uk]$ where $[\cdot]$ denotes the integer part. If $j \leqslant n$, the jth member of the current sample is replaced by the kth population member. Go to Step 2.

At the conclusion of the algorithm, k contains the value of N.

3. COMPUTER SIMULATION EXPERIMENTS

The algorithm described in Section 2 as well as the Fan-Bebbington and Minitab algorithms were coded using the random number generator of Wichmann and Hill (1982). These algorithms were then run on both the CYBER 835(NOS) and PRIME 400 Computers. On the CYBER machine, the algorithms were run under all levels of optimization with the FTN compiler. One hundred simple random samples of size 25, 50 and 75 were generated from populations of size 100, 1000 and 10 000. Table 1 shows the CPU time on the CYBER machine under the highest level of optimization. Many other tests on both the PRIME and CYBER showed exactly the same pattern in the results.

From Table 1, it is apparent that the algorithm of Section 2 is consistently better than the other two. The difference between the algorithm of Section 2 and the Fan-Bebbington algorithm is always small and decreases as the sampling fraction decreases. The Minitab algorithm is always much worse than the other two. In these experiments, the entire population was stored in the fast memory of the computer. In other situations, it might be expected that the Fan-Bebbington

algorithm would be more efficient since with this algorithm it is not necessary to enumerate the entire population.

TABLE 1
CPU Time in Seconds (MB: McLeod-Bellhouse; FB: Fan-Bebbington; M: Minitab) for 100 Samples

		N		
n	*Algorithm*	100	1000	10 000
25	MB	0.52	6.51	66.57
	FB	0.69	6.71	66.59
	M	1.37	13.01	132.87
50	MB	0.37	6.38	66.42
	FB	0.73	6.82	68.07
	M	2.10	19.84	204.61
75	MB	0.20	6.21	66.05
	FB	0.74	6.91	68.65
	M	2.90	26.60	275.86

4. THE RANDOM CHOICE RULE

The problem of cycling or circling in linear programming computer codes may occur when there is a tie for the leaving basic variable. In this situation, it is possible for the simplex method to cycle continuously through a fixed set of basic feasible solutions. Kotiah and Steinberg (1977) have reported that this difficulty is of some importance in actual practical linear programming problems.

The perturbation method (Dantzig, Ch. 10) can be used to guarantee convergence in this case but it entails considerably more calculation than the normal simplex iteration. Dantzig (1963, pp. 120–123) pointed out that the random choice rule is more expedient than the perturbation method and also guarantees convergence of the simplex algorithm. Under the random choice rule, the leaving variable is chosen at random from among all tied candidates. A one-pass algorithm for the random choice rule can be obtained from the algorithm of Section 2 with $n = 1$. Note that in this situation, the number of ties, N, is unknown.

ACKNOWLEDGEMENTS

This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Bebbington, A. C. (1975) A simple method of drawing a sample without replacement. *Appl. Statist.*, **24**, 136.

Dantzig, G. B. (1963) *Linear Programming and Extensions*. Princeton: Princeton University Press.

Fan, C. T., Muller, M. E. and Rezucha, I. (1962) Development of sampling plans by using sequential (item by item) selection techniques and digital computers. *J. Amer. Statist. Ass.*, **57**, 387–402.

Kotiah, T. C. T. and Steinberg, D. I. (1978) On the possibility of cycling with the simplex method. *Oper. Res.* **26**, 374–376.

Ryan, T. A., Joiner, B. L. and Ryan, B. F. (1981) *Minitab Reference Manual*. Pennsylvania State University.

Wichman, B. A. and Hill, I. D. (1982) An efficient and portable pseudo-random number generator. *Appl. Statist.*, **31**, 188–190.