

Generating Random Variables With a t -Distribution

By George Marsaglia

Abstract. Let RNOR and REXP represent normal and exponential random variables produced by computer subroutines. Then this simple algorithm may be used to generate a random variable T with t_n density $c(1 + t^2/n)^{-1/2}n^{-1/2}$, for any $n > 2$:

Generate $A = \text{RNOR}$, $B = A^2/(n-2)$ and $C = \text{REXP}/(\frac{1}{2}n-1)$ until $e^{-B-C} \leq 1 - B$, then exit with $T = A[(1-2/n)(1-B)]^{-1/2}$.

Discussion. This note gives a simple algorithm for computer generation of random variables from the t_n distribution—Student's t -distribution with n degrees of freedom. With an efficiency greater than $1 - 1/n$, the above rejection technique provides a method for t_n variates that is simpler and faster than those of Kinderman, Monahan and Ramage [1], recently published in this journal. It is simpler but not as fast as methods based on the exact-approximation method [4]. Comparison of methods depends on the computer used, the random variables and subroutines required, the set-up time and so on. The above procedure is so simple that potential users can readily try it themselves, and it has the additional advantage that the time for each t_n variate is virtually constant, even with n changing from call to call.

The claim that such a simple algorithm is faster than published ones is premised on the availability of very fast subroutines for RNOR and REXP, which the algorithm is designed to exploit. Most large computer installations have, or should have, such very fast RNOR and REXP. Users who do not have such might want to consider getting them and using the above algorithm rather than implementing more elaborate and slower published methods for the t -distribution. References [2] and [3] describe fast procedures for RNOR and REXP, and a random number package "Super-Duper" containing them is available from the author.

Proof That the Method Works. The idea behind the method is to use an intermediate density function, $c'(1-x^2)^{1/2}n^{-1}$, $-1 < x < 1$, chosen because it is close to a normal density and because a variate X with that density is readily converted to the required variate T : If X has density $c'(1-x^2)^{1/2}n^{-1}$, then $T = n^{1/2}X(1-X^2)^{-1/2}$ will have the t_n density $c(1+t^2/n)^{-1/2}n^{-1/2}$.

The problem then is to generate such an X . Using the elementary inequality $1-x^2 \leq e^{-x^2}$, we see that

$$(1-x^2)^{1/2}n^{-1} \leq e^{-x^2}(\frac{1}{2}n-1).$$

Received February 9, 1979; revised April 24, 1979.

AMS (MOS) subject classifications (1970). Primary 65C10, 62E25.

Except for normalizing constants, the right side is the density of $\text{RNOR}/(n-2)^{1/2}$ and the left side is our intermediate density. This provides the basis for the rejection technique: generate $X = \text{RNOR}/(n-2)^{1/2}$ until

$$(1) \quad Ue^{-X^2(\frac{1}{2}n-1)} \leq (1-X^2)^{\frac{1}{2}n-1},$$

where U is a uniform variate. This amounts to choosing a point (X, Y) uniformly under the curve $y = e^{-x^2(\frac{1}{2}n-1)}$ until we get one under the curve $y = (1-x^2)^{\frac{1}{2}n-1}$.

To avoid taking the $\frac{1}{2}n-1$ power of the right side of (1), we use the fact that $e^{-\text{REXP}}$ is a uniform variate, and thus inequality (1) is equivalent to the one given in the algorithm. A small calculation shows that the efficiency of the rejection technique is $(\frac{1}{2}n-1)^{1/2}\Gamma(\frac{1}{2}n)/\Gamma(\frac{1}{2}n+1/2)$, which is slightly greater than $1-1/n$.

The parameter n of the algorithm need not be an integer, and the average running time may be improved for small n , say $2 < n < 6$, by using this version:

Generate $A = \text{RNOR}$, $B = A^2/(n-2)$ and $C = \text{REXP}/(\frac{1}{2}n-1)$ until $1-B > 0$ and $e^{-B-C} < 1-B$, then exit with $T = A[(1-2/n)(1-B)]^{-1/2}$.

Rejecting for $1-B < 0$ will save the need to generate and test with an REXP . For small n the cost of the additional test is justified.

Computer Science Department
Washington State University
Pullman, Washington 99164

1. A. J. KINDERMAN, J. F. MONAHAN & J. G. RAMAGE, "Computer methods for sampling from Student's t -distribution," *Math. Comp.*, v. 31, 1977, pp. 1009-1018.
2. M. D. MACLAREN, G. MARSAGLIA & T. A. BRAY, "A fast procedure for generating exponential random variables," *Comm. ACM*, v. 7, 1964, pp. 298-300.
3. G. MARSAGLIA, K. ANANTHANARAYANAN & N. J. PAUL, "Improvements on fast methods for generating normal random variables," *Information Processing Lett.*, v. 5, 1976, pp. 27-30.
4. G. MARSAGLIA, "The exact-approximation method for generating random variables in a computer." (To appear.)