

ORNL/TM-10670

**OAK RIDGE
NATIONAL
LABORATORY**

MARTIN MARIETTA

**PLOTSAM: A Simulation of
Systematic and Random Sampling**

R. L. A. Franklin
R. H. Gardner
V. H. Dale
P. S. White

Environmental Sciences Division
Publication No. 3025

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULARS SECTION
OAK RIDGE, TENN.
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this
report, send its name with report card
the library will arrange a loan



OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ENVIRONMENTAL SCIENCES DIVISION

PLOTSAM: A SIMULATION OF SYSTEMATIC AND RANDOM SAMPLING

Renée Lee Ann Franklin¹
Robert H. Gardner
Virginia H. Dale
Peter S. White²

Environmental Sciences Division
Publication No. 3025

¹Present address: P.O. Box 2337, Nicholls State University,
Thibodaux, Louisiana 70310

²Department of Biology, University of North Carolina, Chapel Hill,
North Carolina 27514

Date Published -- November 1988

Prepared for the
Carbon Dioxide Research Division
Office of Energy Research
(Activity No. HA 02 05 00 0)

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400



3 4456 0283788 4

CONTENTS

	<u>Page</u>
LIST OF FIGURES	v
LIST OF TABLES	vii
ACKNOWLEDGMENTS	ix
ABSTRACT	xi
INTRODUCTION.	1
METHODS	3
DESCRIPTION OF PLOTSAM	3
SAMPLING DESIGN FOR A TEST CASE	14
RESULTS	17
DISCUSSION	27
REFERENCES	29
APPENDIX. LISTING OF THE PROGRAM PLOTSAM	31

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	Hierarchical structure of PLOTSAM	4
2	User-specified probability distributions of input parameters used in PLOTSAM with μ as the population mean and σ as the standard deviation	9
3	Diagram of systematic and random sampling methods	11
4	Diagram of the location of individuals within a plot and the corresponding variance-to-mean ratio	15
5	Sample mean and the ratio of the standard error to the mean based on sampling a normal distribution	18
6	Sample mean and the ratio of the standard error to the mean based on sampling a uniform distribution	19
7	Sample mean and the ratio of the standard error to the mean based on sampling a lognormal distribution	20
8	Sample mean and the ratio of the standard error to the mean based on sampling a triangular distribution	21

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Menu of choices for setting sampling conditions	6
2	Characteristics of the species to be sampled, as determined by the user	7
3	Summary tables printed by PLOTSAM with plot information and results from sampling and statistical tests	13
4	Results from sampling a normal distribution	22
5	Results from sampling a uniform distribution	23
6	Results from sampling a lognormal distribution	24
7	Results from sampling a triangular distribution	25

ACKNOWLEDGMENTS

This paper has been reviewed by John J. Beauchamp and Antoinette L. Brenkert. Discussions with W. M. Post and S. W. Christensen about the research have been helpful. Editing by Gay Marie Logsdon greatly improved the manuscript.

Renée Franklin was supported by a summer internship with Oak Ridge Associated Universities. The research was sponsored by the Carbon Dioxide Research Division, Office of Basic Research Sciences, and by the National Science Foundation's Ecosystem Studies Program under Interagency Agreement No. BSR-8315185 with the U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

ABSTRACT

FRANKLIN, R. L. E., R. H. GARDNER, V. H. DALE, and P. S. WHITE.
1988. PLOTSAM: A simulation of systematic and random
sampling. ORNL/TM-10670. Oak Ridge National Laboratory,
Oak Ridge, Tennessee. 68 pp.

PLOTSAM the computer program, that simulates sampling of organisms or environmental conditions, is described. The program has three major components: (1) a specification of the location and the statistical characteristics of the organisms or conditions, (2) a sampling, and (3) a computation of statistical tests. With PLOTSAM, the location of the organisms or site conditions can be specified as a systematic or bivariate normal distribution, and the attributes can have a normal, lognormal, uniform, or triangular distribution. Furthermore, sampling can be performed by placing quadrats randomly or systematically along one or more transects in a square plot. Descriptive statistics of the complete and sampled data sets can be examined. Then statistical tests can be performed to compare the complete and sampled data sets.

The advantage of using a simulation for sampling is that the sampled information can be compared to the complete population and that the accuracy and precision of the sampling methods thus can be tested. Knowledge of the adequacy of the sampling regime is otherwise difficult to obtain in many ecological studies because the populations are frequently so large that collecting all of the measurements is not feasible.

Using PLOTSAM, a test is performed to examine the effects on the estimates of the mean and the variance when the number of samples is increased and different distributions of the attribute are simulated. This test was parameterized with values obtained from a sampling of available soil nitrogen with ion exchange resin bags. The simulations suggest that the increase in accuracy associated with an increase in sample number is related to the skewness and kurtosis of the true population. The simulations show no differences between the random and

the systematic sampling designs. The ratio of the standard error to the mean can be used as a guide to indicate the adequacy of the sample.

This documentation of PLOTSAM allows other users to readily access and interpret the sampling program.

INTRODUCTION

Sampling is a basic part of environmental science because it is difficult to obtain complete information about a habitat, community, or ecosystem. The prime obstacles to collecting complete ecological information are the large amount of potential data (e.g., number of plants in an area) and the expense and difficulty of taking field measurements (e.g., estimates of soil nitrogen availability). After the data are obtained, it is often impossible to evaluate the degree to which the sample data are representative of the complete population.

This report describes PLOTSAM, a computer program that allows exploration of the adequacy of particular sampling schemes. First, PLOTSAM generates the spatial location of species within a plot and their attributes or environmental conditions. An attribute might represent the presence, age, size, or phenology of sessile organisms; the soil particle size or nutrient conditions; or another environmental condition. Second, PLOTSAM samples the attribute via quadrats placed systematically or randomly within the plot. Descriptive statistics and test statistics are computed in order to allow the user to compare the sampled data and the true population conditions.

The advantages of using a computer simulation are (1) the user can specify particular conditions of the system being sampled, (2) time and money required for collecting actual field samples are saved, (3) replication of the sampling scheme can be repeated with changes in only one condition at a time, and (4) the true distribution of the population is known. Therefore, statistical comparisons can be made between the sampled population and the known population in order to evaluate the adequacy of the sample. Computer simulations have proved useful in exploring the characteristics of particular sampling designs (e.g., Gauch 1973, Green 1979, Lim and Khoo 1985).

PLOTSAM can be used to examine the effects that of the sampling scheme and the distribution and location of the attributes have upon the sample statistics. The sampling design can be specified as quadrats either randomly or systematically placed, and various plot sizes and

numbers of samples can be simulated. The distribution of the species or site attribute can be normal, lognormal, triangular, or uniform. The program is written so that from one to five attributes can be sampled. The spatial location of the attributes can be specified as uniform or bivariate normal. Thus the location of individuals of a species might be established as random, clumped, or uniform. For a soil attribute, it is most reasonable to set up the program such that soil samples can be collected at any location in the plot.

PLOTSAM is used in this study to examine the effect on the adequacy of the sample when the number of quadrats is increased. Sample size is the focus of the exercise because it is frequently not given adequate attention in the design of experiments and field collections (e.g., see Croy and Dix 1984). This exercise was meant to simulate the sampling of available nitrogen in the soil through the use of ion exchange resin bags. The resin bag technique involves placing cation and anion exchange resins in nylon bags beneath the forest floor, removing the bags at specified intervals, chemically extracting the ions from the beads, and analyzing the extracts for nitrate and ammonium (Binkley and Matson 1983, Binkley et al. 1986, Hart and Binkley 1985). The resulting ion levels correlate with soil microtopography as well as plant distribution and abundance (Gibson 1987).

One would usually expect that increasing the sample size would improve the precision of the sample, but at some sample sizes the improvements in precision would be outweighed by the added expense of collecting more samples. For a normally distributed attribute, the decreases in precision that occur with increases in sample size are presented in power curves (e.g., see Dixon and Massey 1969). The simulations suggest that the increase in precision is related to the kurtosis and skewness of the distribution of the attribute.

METHODS

DESCRIPTION OF PLOTSAM

PLOTSAM, a TURBO PASCAL program (Borland International 1985), was created to simulate both systematic and random sampling for one or more plant species or soil characteristics. PLOTSAM runs on an IBM PC or compatible machine. The units of measurement are not defined in PLOTSAM; therefore, neither the type of observation nor the size of the study plot and sample quadrat are restricted.

PLOTSAM'S hierarchical structure is shown in Fig. 1. Constant, type, and variable declarations are made in the main portion of the program. The maximum number of species or soil characteristics allowed, the size of the plot, and the number of individuals in the plot are declared constants. The user can determine whether the sampling will be done systematically or randomly.

PLOTSAM uses a random number to assign the position and attribute of each species in the plot. RANDOM, the built-in TURBO PASCAL function, is used in the procedure URAND to generate the sequence of random numbers (called pseudo-random because they are chosen from a uniform distribution that may ultimately repeat) (Schildt 1986). The cycle length of the RANDOM numbers is very large and will not repeat within any PLOTSAM simulation. The same sequence will be generated for continuous runs of PLOTSAM unless the random number seed is changed.

The main program of PLOTSAM calls the following procedures to simulate a sampling: INITIALIZE, SETUP, GENERATEPLOTDATA, SUMMARIZE, ZEROARRAYS, SAMPLEPLOT, ACCUMULATE, DOCUSAM, PRINTSTAT, and TESTS. A source listing of PLOTSAM is presented in the Appendix.

INITIALIZE allows the user to set the random number seed; the number of species represented in the plot; and the name, mean, and standard deviation for the location of the individuals of each species. Furthermore, the user can indicate whether the results should be sent to a printer. INITIALIZE references a file that contains the default values for the length of the quadrat; the total number of quadrats; and the location, attributes, and frequency of each species.

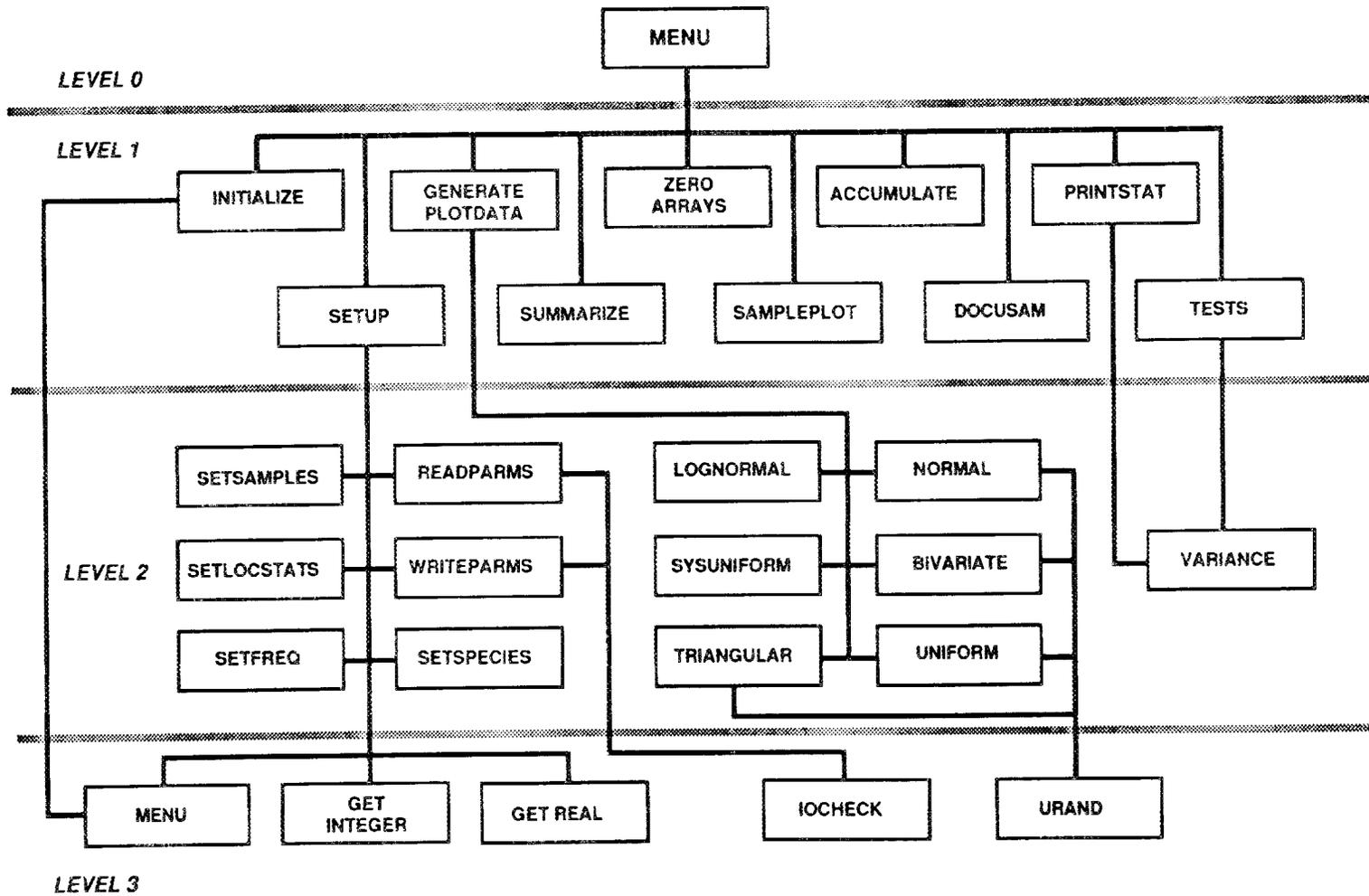


Fig. 1. Hierarchical structure of PLOTSAM.

SETUP creates a menu (see Table 1) to change the sampling conditions so that the user does not have to access the source code of PLOTSAM. The user can change the characteristics for each species as in (Table 2) and save the values in a file or reference a different file. In other words, the user needs to indicate the location of individuals of each species, as well as the mean, standard deviation, and range of the species characteristics.

The procedures GETINTEGER and GETREAL check all values entered by the user to make sure they are of the proper type. MENU writes the necessary prompts on the screen. IOCHECK makes sure that the file being used in INITIALIZE and SETUP is accessible.

GENERATEPLOTDATA determines the mean attribute and the x-y coordinates for each individual of a species. For each axis, the attributes can have a normal, lognormal, uniform, or triangular distribution. The x-y coordinates are defined by either a sysuniform or bivariate distribution (discussed below). The parameter values MEAN, SD, MIN, and MAX are used in the following discussions to represent the average, standard deviation, minimum value, and maximum value of the species characteristics as specified by the user.

Through a procedure described by Rubinstein (1981, pp.89-90), the normal distribution is calculated from the sum of 12 random numbers (RAN) ranging from 0 to 1:

$$\left(\sum_{i=1}^{12} \text{RAN} - 6 \right) * \text{SD} + \text{MEAN} \quad . \quad (1)$$

Thus, the simulated attribute has a potential bound of six standard deviations on the error of estimation of the mean (Fig. 2A).

The lognormal distribution is computed by substituting the natural log of the MEAN and the SD into the exponential of the normal distribution (Eq. 1) (Fig. 2B).

Table 1. Menu of choices for setting sampling conditions

<C>	Set Species Attributes ^a
<X>	Set X-Y Location Statistics ^a
<F>	Set Species Frequencies ^a
<D>	Define Sample Methods
<I>	Input Parameters from Disk
<S>	Save Parameters to Disk
<R>	Run Simulation
<Q>	Quit Simulation

^aSee Table 2 for further information.

Table 2. Characteristics of the species to be sampled,
as determined by the user

Species attribute						
Name	Distribution ¹	Mean	SD	Min ¹	Max ¹	Frequency
One	normal	58.3	10	--	--	2500
Species location						
Location on Axis						
Name	Axis	Distribution ²	Mean	SD	Min	Max
One	x	sysuniform	25	--	0	50
	y	sysuniform	25	--	0	50

^aThe distribution of the species attribute could also be specified as lognormal, triangular, or uniform. The normal and lognormal use the MEAN and SD (not the MIN or MAX) to calculate the distribution. The triangular and uniform use the MEAN, MIN, and MAX (not the SD) to calculate the distribution.

^bThe distribution of the x-y location could also be specified by the user as bivariate. The location is calculated using the MEAN, MIN, and MAX for the sysuniform and using the MEAN and SD for the bivariate.

The uniform distribution (Fig. 2C) will generate values that are randomly placed between the MIN and MAX,

$$\text{RAN} * (\text{MAX} - \text{MIN}) + \text{MIN} . \quad (2)$$

following Rubinstein (1981, p. 40):

The triangular density function (Fig. 2D) increases linearly up to the mean and decreases linearly for values of the random variable that are greater than the mean. First a uniformly distributed random number (URAN) is calculated using Eq. (2). Then the attribute for a triangular distribution is computed with

$$\text{MAX} - \text{sqrt}[(\text{MAX} - \text{MIN}) * (\text{MAX} - \text{MEAN}) * (1 - \text{URAN})] , \quad (3)$$

if URAN >= MEAN or with

$$\text{MIN} + \text{sqrt}[(\text{MAX} - \text{MIN}) * (\text{MEAN} - \text{MIN}) * \text{URAN}] , \quad (4)$$

if URAN < MEAN.

The x-y location of the individuals within the square plot can be specified by either a systematic or a bivariate normal distribution. The function sysuniform places the number of individuals of each species (I) at regular intervals along transects in the plot. The values for the x locations are incremented from 1 to PLOTSIDE using the remainder (from integer division) of IN/PLOTSIDE (where IN = 1, ..., I and PLOTSIDE is the length of the plot). The values for the y location of the individuals are incremented by 1 when IN is one more than PLOTSIDE. The probability density function for sysuniform is the same as that for uniform (Fig. 2B), but the computations for sysuniform are different because it does not depend on RAN.

The bivariate distribution is computed using Box-Muller's transformation modified to generate two independent standard normal

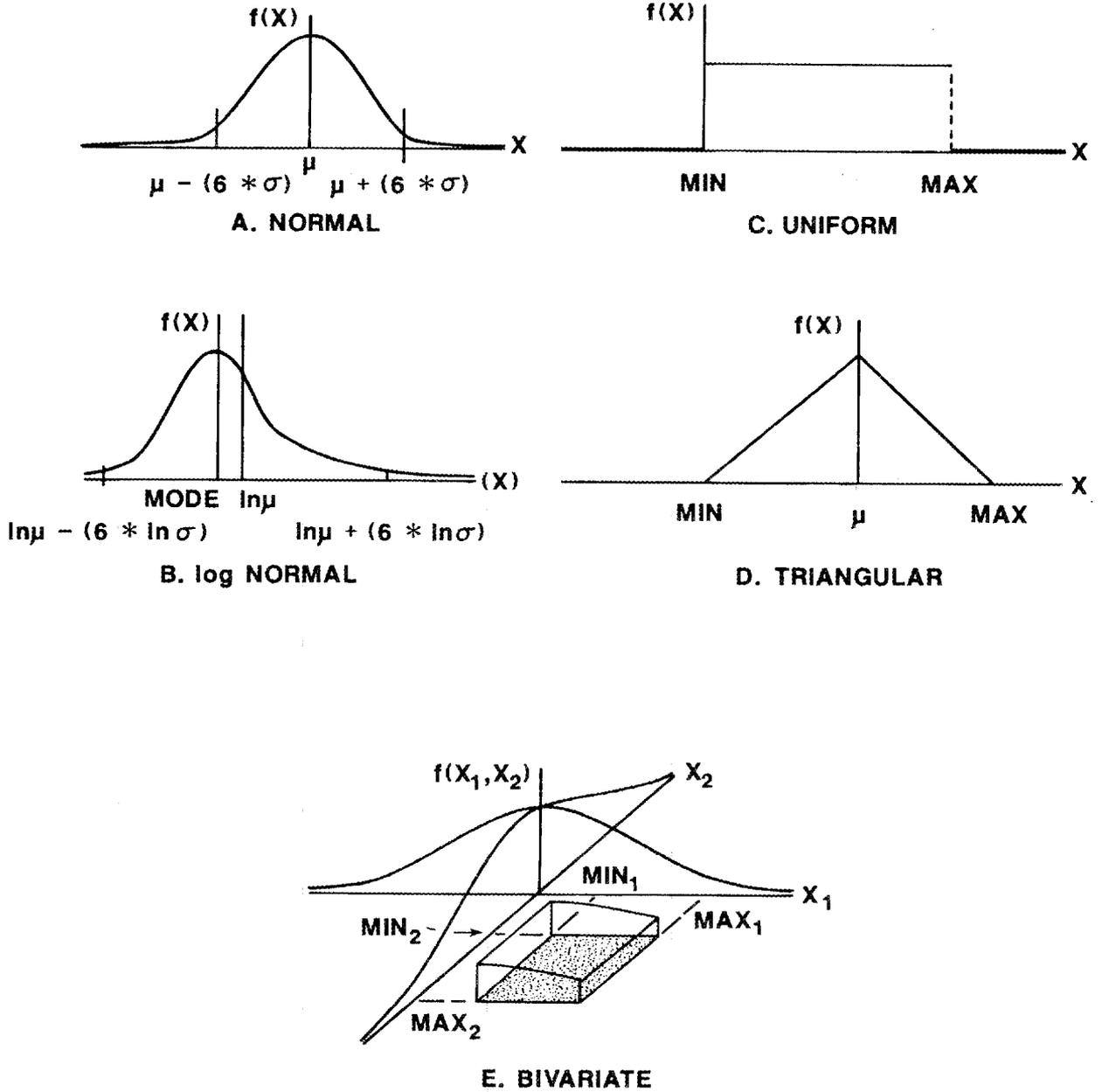


Fig. 2. User-specified probability distributions of input parameters used in PLOTSAM WITH μ as the population mean and σ as the standard deviation.

random variables (Fig. 2E) (see Rubinstein (1981), p. 86-87). The x and y locations are generated by the equation

$$X = \text{MEAN}_X + (Z1 * \text{SD}_X) , \quad (5)$$

$$Y = \text{MEAN}_Y + (Z2 * \text{SD}_Y) , \quad (6)$$

where MEAN_X , SD_X , MEAN_Y , and SD_Y are the mean and standard deviation indicated for x and y, respectively. The values for z1 and z2 are given by

$$Z1 = \text{sqrt}(-2 * \ln \text{RAN}_X) * \cos(2 \pi * \text{RAN}_Y) , \quad (7)$$

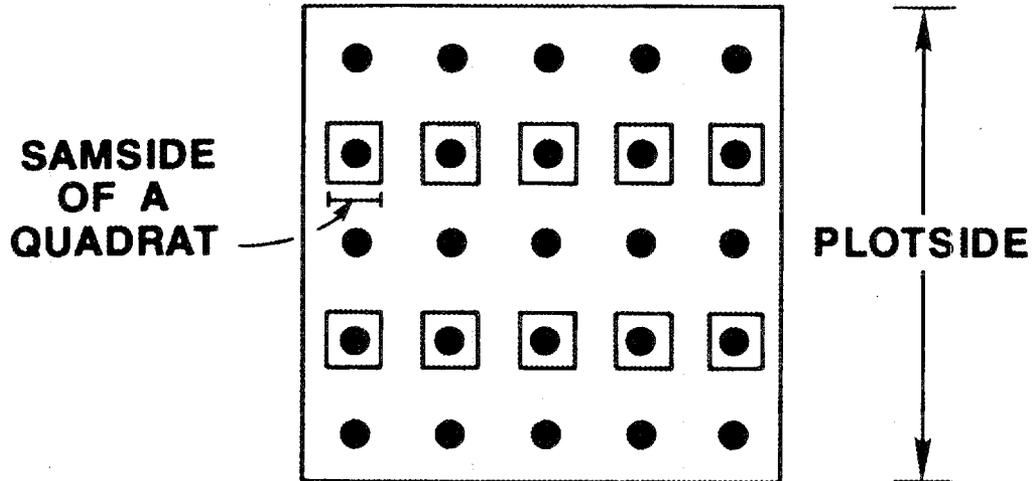
$$Z2 = \text{sqrt}(-2 * \ln \text{RAN}_X) * \sin(2 \pi * \text{RAN}_Y) , \quad (8)$$

where RAN_X and RAN_Y are the random numbers generate for x and y, respectively (Rubinstein, 1981, p. 86). When soil properties are the focus of the research, GENERATEPLOTDATA should be used to create complete coverage by specifying a sysuniform distribution with the mean and the range of the species locations equal to the mean and the range of the plot because soil characteristics can be measured at any point.

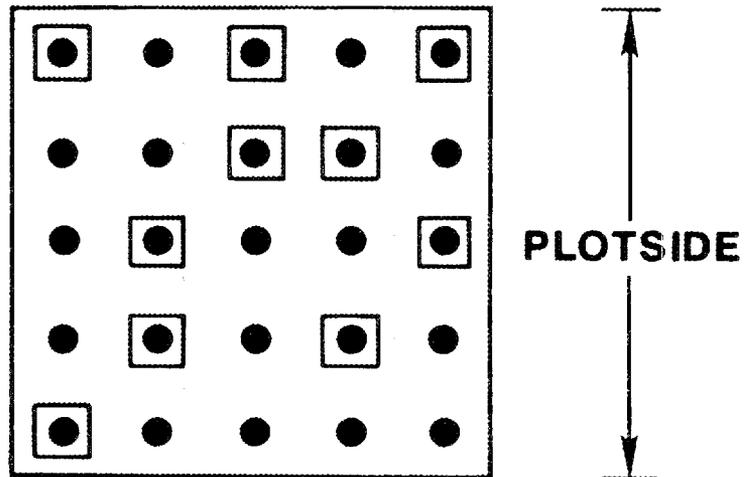
SUMMARIZE prints the total number of individuals, species, and samples; the size of the plot and quadrat; the plot ratio (plot area divided by the quadrat area); the sum of the mean attributes of the species; and the plot mean (the sum of the mean attribute divided by the total number of individuals in the plot).

ZEROARRAYS creates storage locations for the sampling's sum, as well as the minimum and maximum values and it initializes them to zero, 1000, and -1000, respectively.

SAMPLEPLOT generates sample points either systematically or randomly (Fig. 3). Samples are systematically taken at regular intervals along transects. The number of transects laid (TN) depends on the truncated value of the ratio of the number of samples (NSAMS) to



**A. SYSTEMATIC METHOD:
 QUADRATS ARE PLACED
 SYSTEMATICALLY ALONG
 2 TRANSECTS**



**B. RANDOM METHOD:
 QUADRATS ARE PLACED
 RANDOMLY USING RAN**

Fig. 3. Diagram of systematic and random sampling methods.

the length of the side of the plot (PLOTSIDE). If the NSAMS is a multiple of PLOTSIDE, then

$$TN = NSAMS/PLOTSIDE . \quad (9)$$

Otherwise,

$$TN = (NSAMS/PLOTSIDE) + 1 . \quad (10)$$

The location of each sample (SN) is given by the truncated values of the coordinates X(SN) and Y(SN) (where SN = 1, 2, ... ,N). The coordinates are generated in the following manner. For the systematic sample,

$$X(SN) = SN - [(J-1) * PLOTSIDE] , \quad (11)$$

$$Y(SN) = [J/(TN+1)] * PLOTSIDE , \quad (12)$$

where J is the transect being sampled. For the random sample,

$$X(SN) = RAN * (PLOTSIDE - SAMSIDE) , \quad (13)$$

$$Y(SN) = RAN * (PLOTSIDE - SAMSIDE) , \quad (14)$$

where SAMSIDE is the length of the quadrat.

DOCUSAM gives the user the option to print the sum of the mean attributes and the number of individuals in each quadrat.

ACCUMULATE computes the sum, minimum, and maximum of the sampling by using the storage locations created in ZEROARRAYS to keep a running total of the mean attributes and to check for the minimum and maximum value each time a sample point is generated.

After the N samples have been taken, PRINTSTAT is used to print the results (Table 3). From the results the procedure TESTS computes the Student's t-statistic and the variance-to-mean ratio and then compares them with the critical values entered by the user. The Student's t-distribution is used to test the hypothesis that the sample

Table 3. Summary tables printed by PLOTSAM with plot information and results from sampling and statistical tests

Information specified by user

```

-----
No individuals = 2500
No species    = 1
No samples    = 500
Plot size     = 50 X 50
Sample size   = 0.99 X 0.99
Plot ratio    = 2550.76
Plot sum      = 145772.70
Plot mean     = 58.3091
-----

```

Calculated attributes for species based on specification given above

```

-----
Mean      sd      Min      Max      CV
58.18    10.19   31.17    89.59    17.5%
-----

```

Student T-test

```

-----
No of samples taken = 500
Null hyp:           Mean = 58.3091
Altn hyp:           Mean <> 58.3091
Test stat:          T = -0.2802
                    with 499 degrees of freedom
Rej region:         ABS(T) > 1.965
-----

```

Description of distribution using variance/mean

```

-----
Sample variance     = 103.7382
Sample mean        = 58.18
Ratio of the sample variance
to the sample mean = 1.783
 $\chi^2_{499}$  = 1.11
The distribution is AGGREGATED.
-----

```

mean is equal to the true mean based on a two-sided test. The variance-to-mean ratio is used as a descriptive statistic of the pattern of availability of the species (Pielou 1969, p. 91). When compared to the χ^2 distribution with $N-1$ degrees of freedom, the variance-to-mean ratio describes the amount of randomness of the pattern. Rejection of the hypothesis of randomness indicates that the distribution of the species characteristics is either uniform or contagious (Fig. 4).

After the sampling and statistical tests results have been printed, the user can choose to sample the plot again or to stop the simulation.

SAMPLING DESIGN FOR A TEST CASE

A test case was set up to examine the effects on the precision of the estimate when the number of samples is increased. The test was meant to replicate sampling for available soil nitrogen with ion exchange resin bags. Therefore, the size of the quadrat was not allowed to vary. The test was set up to determine the number of samples that adequately characterizes specified patterns of available nitrogen.

Simulated sampling was conducted for a 50 x 50 plot gridded into 2500 sample points. Thus, the locations of the potential sites for soil collections were distributed (using the procedure SYSUNIFORM) such that each quadrat contained soil information. The length of the plot was set at 50 because the COMPAC Portable II computer used in this exercise indicated an overflow in memory for values greater than 56. The mean of the soil characteristic for each simulation was chosen from a population mean (μ) of 58.3 (based on Bloss 1987) and a standard deviation (σ) of 10 (based on evaluation of data from Bloss 1987 and numerous simulations with different values for the standard deviation). Since μ and σ are calculated for each simulation, there can be slight differences between simulations. In a series of simulations,

ORNL-DWG 87-18727

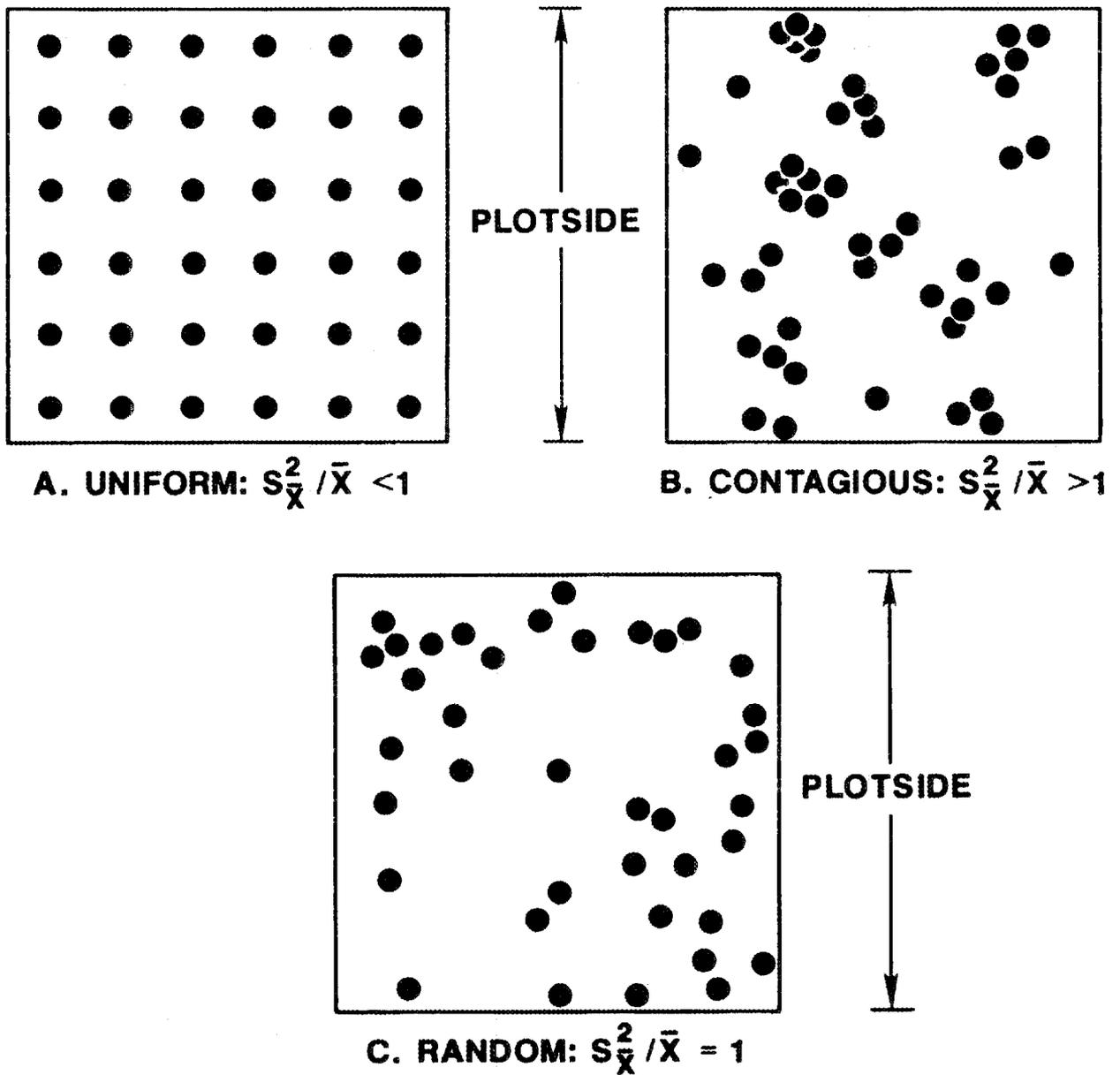


Fig. 4. Diagram of location of individuals within a plot and the corresponding variance-to-mean ratio.

the distribution of the soil characteristic was specified as normal, uniform, lognormal, and triangular. The maximum value of the soil nitrogen varied from 116 to 250, depending on the distribution used, while the minimum value remained zero. The number of samples varied from 2 to 1000 (where $N = 2, 5, 10, 25, 50, 100, 500, 1000$). The quadrat was of size 0.99×0.99 , which can be interpreted as an edge rule that excludes samples which fall on the margin of a 1×1 quadrat. Samples were taken using both systematic and random sampling methods. The Student's t -statistic and the variance-to-mean ratio were computed for each sampling method.

RESULTS

For all distributions, oscillations in the sample mean decrease, and the values become roughly equivalent to the specified mean as the sample number increases (see Fig. 5 to 8). The ratios of the standard error to the mean decrease when the sample number decreases, as would be expected. The value of the standard error-to-the mean ratio exceeds 10% for samples less than 500 for attributes in the lognormal distribution, less than 25 in the normal and triangular distribution, and less than 10 in the uniform distribution.

The samples from the lognormal distribution (Fig. 7) consistently under estimate the mean. This bias is expected from the lognormal distribution since the mode is less than the mean and since the distribution is positively skewed.

There is a general decrease in the bias of the simulated estimate of nitrogen availability as the sample size increases from 2 to 1000 for both systematic and random sampling methods (Tables 4-7). The decrease in bias is accompanied by an increase in the coefficient of variation (CV%) which is used to measure the relative variability of the sample. For some distributions, the variation increases as N increases from 2 to 5 and from 5 to 10, probably due to the imprecise estimate of the variance for small N.

The increase in the sample mean and the variance between two and five samples in the systematic sampling of an attribute with a lognormal distribution can be understood from the graph of the lognormal probability density function (Fig. 2C). A few samples would not be expected to represent the true variance around the mean because the lognormal distribution has a mode less than the mean. For the simulation in which two samples were taken, the values were chosen, by chance, from a small range (1.62 to 6.60) that produced a low mean and a low variance (Table 6). For the simulation in which five samples were taken, greater variability occurred because the points were chosen from a wider range (12.02 to 94.42) of the attribute (Table 6). Hence, acceptance with 95% confidence in the estimate of the mean availability

ORNL-DWG 87-18721

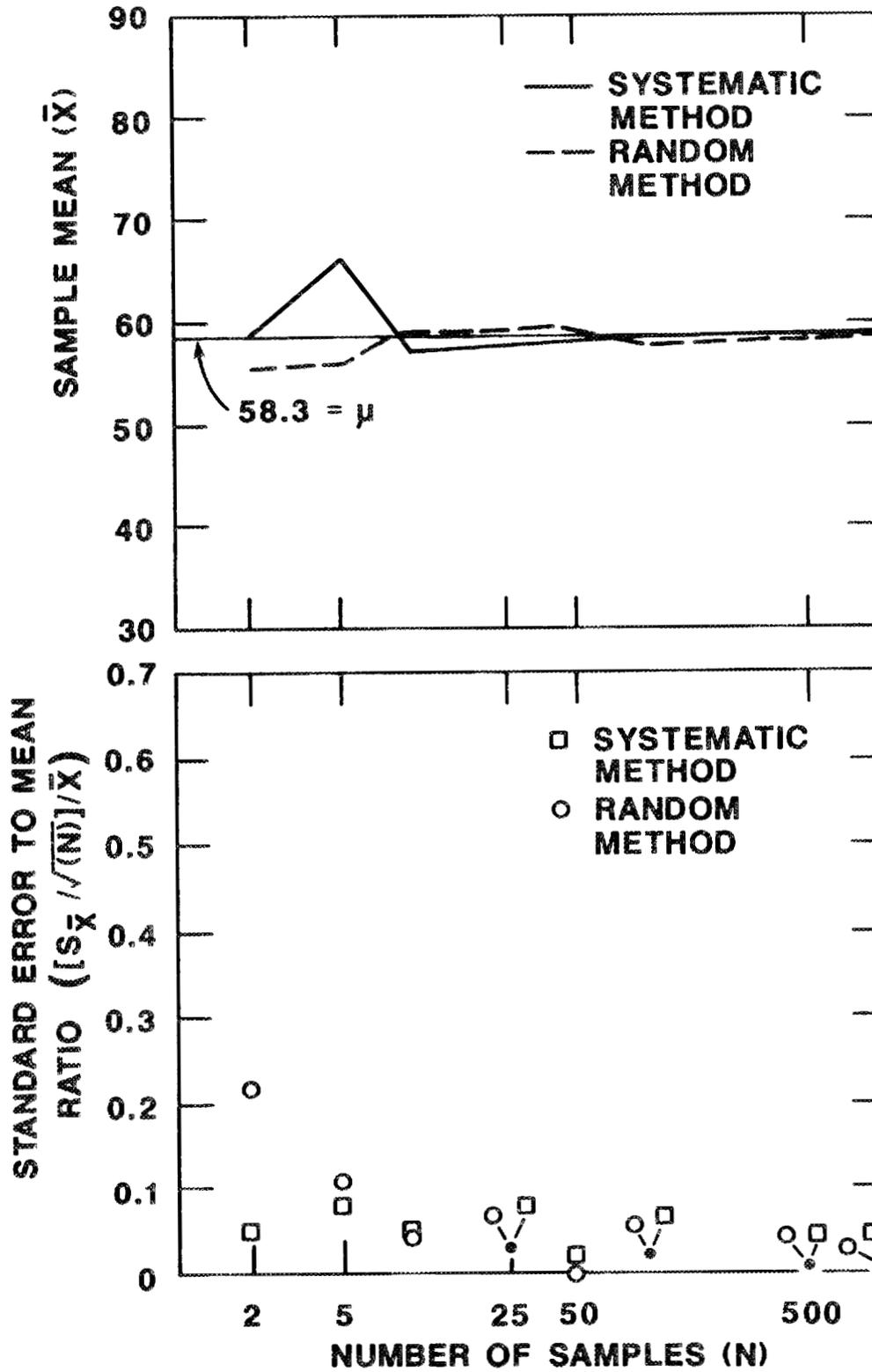


Fig. 5. Sample mean and the ratio of the standard error to the mean based on sampling a normal distribution.

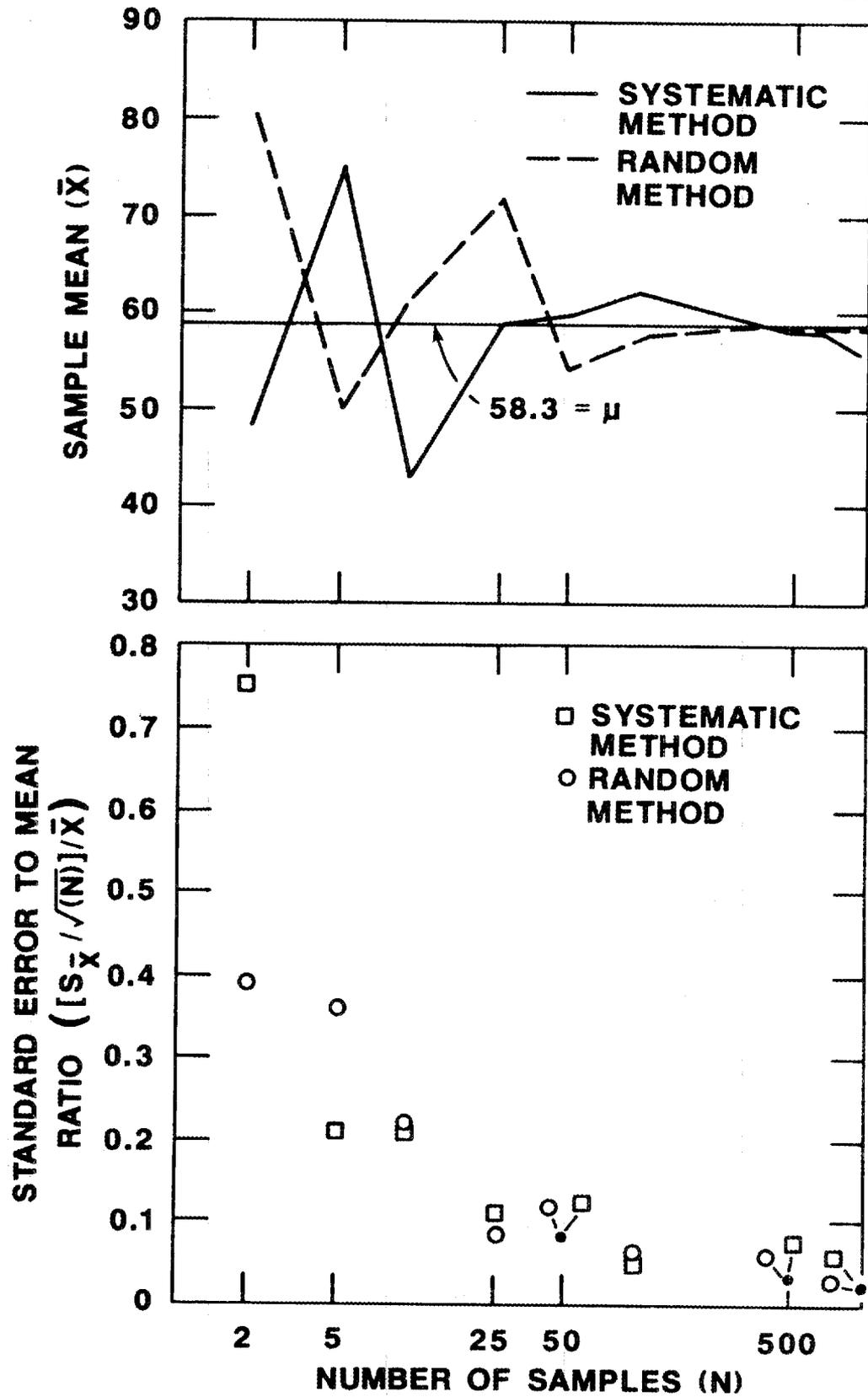


Fig. 6. Sample mean and the ratio of the standard error to the mean based on sampling a uniform distribution.

ORNL-DWG 87-18723

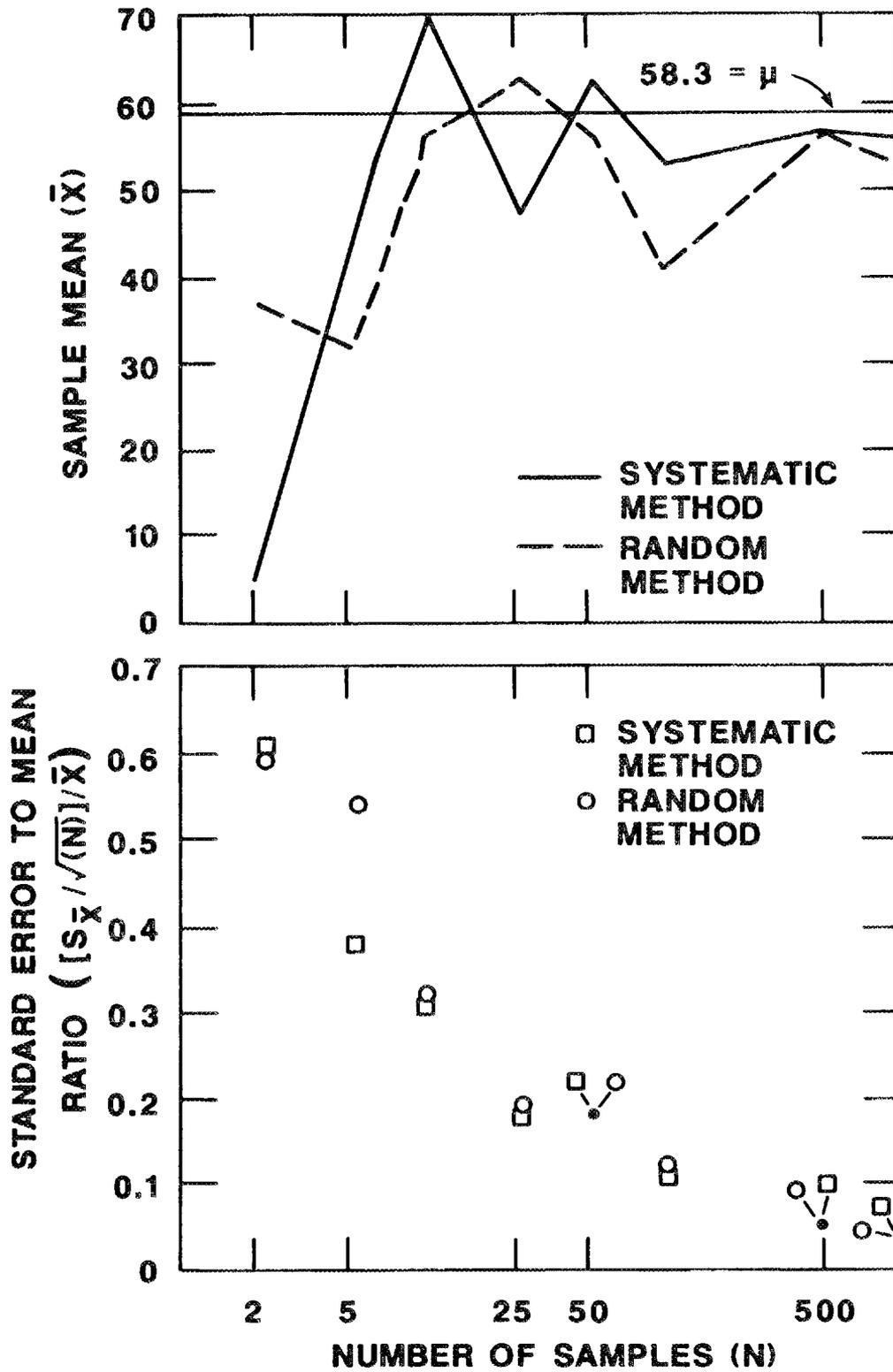


Fig. 7. Sample mean and the ratio of the standard error to the mean based on sampling a lognormal distribution.

ORNL-DWG 87-18724

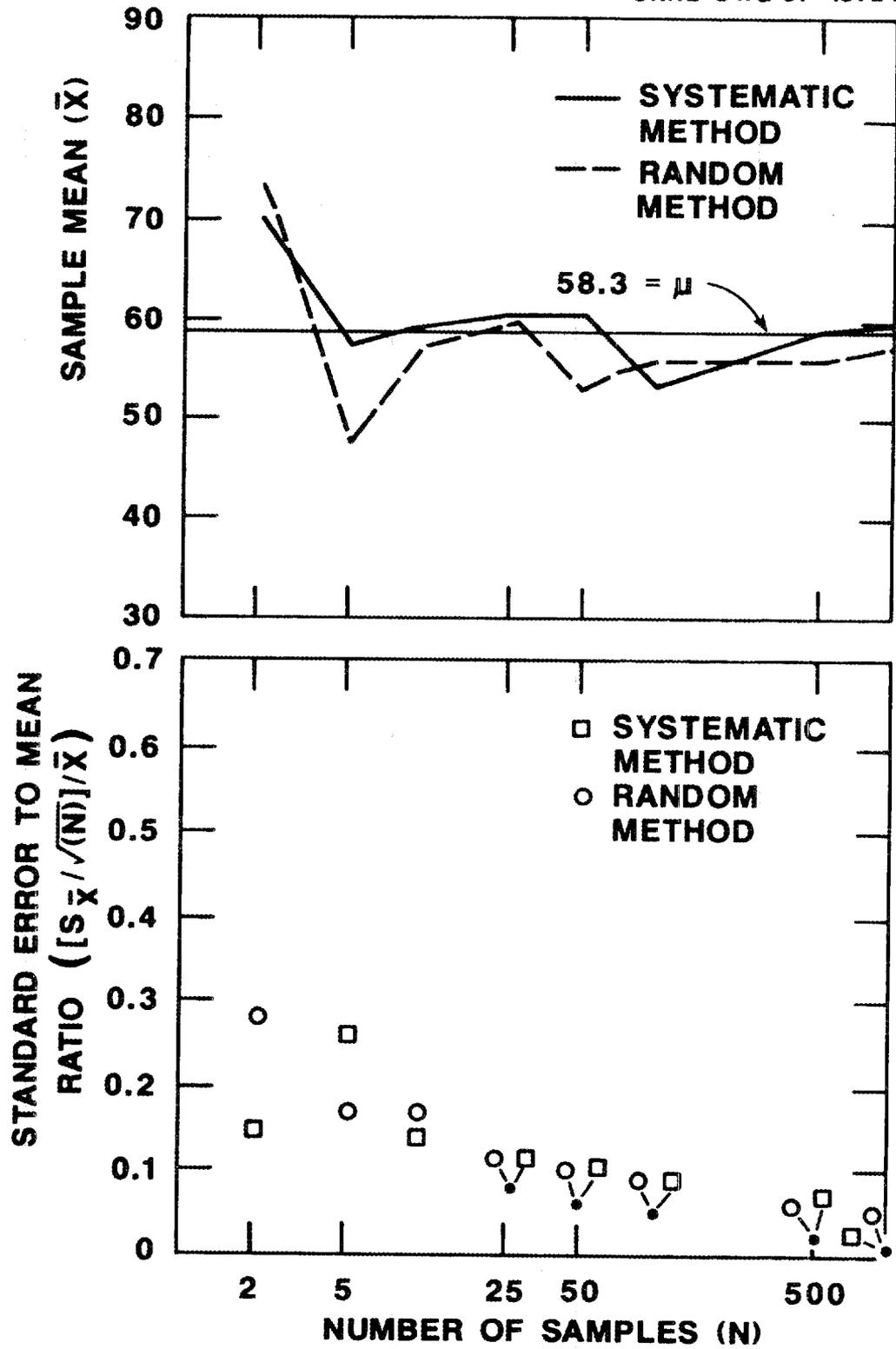


Fig. 8. Sample mean and the ratio of the standard error to the mean based on sampling a triangular distribution.

Table 4. Results from sampling a normal distribution (where N gives the number of samples and μ is the population mean as calculated from a mean of 58.31 specified by the user)

		Systematic Method							
N		2	5	10	25	50	100	500	1000
μ		58.57	58.08	58.31	57.99	58.36	58.36	58.30	58.57
\bar{X}		58.69	66.19	57.06	57.29	57.81	58.87	58.18	58.72
S_X^2		16.93	129.28	93.20	90.64	91.01	109.92	103.73	100.90
MIN		55.78	46.10	46.84	45.58	39.56	32.35	31.17	23.60
MAX		61.60	73.55	75.52	80.6	79.08	80.63	89.59	86.93
$S_{\bar{X}}^2$		0.288	1.953 ^a	1.633 ^a	1.582 ^b	1.574 ^c	1.867 ^d	1.783 ^e	1.718 ^d
CV%		7.0	17.2	16.9	16.6	16.5	17.8	17.5	17.1
T t-test		0.041	1.593	0.410	0.368	0.752	0.478	0.280	0.477
		Random Method							
N		2	5	10	25	50	100	500	1000
μ		58.57	58.08	58.31	57.99	58.36	58.36	58.30	58.57
\bar{X}		55.16	55.82	59.17	58.95	59.71	57.52	57.87	57.82
S_X^2		290.90	194.70	65.05	78.73	80.31	144.33	148.52	141.95
MIN		43.10	38.72	50.37	39.30	33.89	0.00	0.00	0.00
MAX		67.22	69.53	76.52	76.90	76.46	81.92	82.84	85.57
$S_{\bar{X}}^2$		5.274 ^e	3.488 ^c	1.099	1.335	1.345 ^a	2.509 ^a	2.566 ^d	2.454 ^d
CV%		30.9	25.0	13.6	15.1	15.0	20.9	21.1	20.6
T T-test		2.831	0.362	0.334	0.538	1.058	0.707	1.279	1.982 ^b

^aSignificance at $p = 0.10$.

^bSignificance at $p = 0.05$.

^cSignificance at $p = 0.01$.

^dSignificance at $p = 0.0005$.

^eSignificance at $p = 0.025$.

Table 5. Results from sampling a uniform distribution (where N gives the number of samples and μ is the population mean as calculated from a mean of 58.31 specified by the user)

		Systematic Method							
N		2	5	10	25	50	100	500	1000
μ		57.57	58.19	57.49	57.68	57.35	57.24	58.64	58.27
\bar{X}		48.28	75.01	42.69	58.52	59.73	61.95	58.04	58.38
S_x^2		2610.18	1195.21	762.98	977.97	1152.97	952.77	1092.86	1156.68
MIN		12.15	13.98	7.13	9.08	7.17	6.83	0.13	0.00
MAX		84.40	95.87	92.69	108.44	115.25	114.38	115.85	115.95
S_x^2/\bar{X}		54.06 ^a	15.93 ^a	17.87 ^a	16.71 ^a	19.30 ^a	15.05 ^a	18.82 ^a	19.81 ^a
CV%		105.8	46.1	64.7	53.4	56.8	49.3	57.0	58.3
T T-test		0.257	1.087	1.695	0.133	0.494	1.540	0.410	0.096
		Random Method							
N		2	5	10	25	50	100	500	1000
μ		57.57	58.19	57.49	57.68	57.35	57.24	58.64	58.27
\bar{X}		80.63	49.98	61.76	71.89	54.04	56.92	58.57	55.98
S_x^2		1986.82	1617.28	1876.48	820.07	917.05	1306.93	1097.99	1236.71
MIN		49.11	1.08	0.00	3.14	5.27	0.00	0.00	0.00
MAX		112.15	102.22	67.83	113.3	115.52	115.42	115.86	115.94
S_x^2/\bar{X}		24.64 ^a	32.35 ^a	30.38 ^a	11.40 ^a	16.97 ^a	22.96 ^a	18.74 ^a	22.09 ^a
CV%		55.3	80.5	70.1	39.8	56.0	63.5	56.6	62.8
T T-test		0.731	0.456	0.311	2.479 ^b	0.775	0.890	0.051	2.064 ^b

^aSignificance at $p = 0.0005$.

^bSignificance at $p = 0.05$.

Table 6. Results from sampling a lognormal distribution (where N gives the number of samples and μ is the population mean as calculated from a mean of 58.31 specified by the user)

		Systematic Method							
N		2	5	10	25	50	100	500	1000
μ		55.88	57.34	54.79	56.36	56.45	55.88	55.08	56.96
\bar{X}		4.11	44.31	69.87	46.5	62.07	52.36	56.44	55.63
S_X^2		12.39	1428.15	4807.03	1744.40	3689.23	3072.72	4005.59	3776.88
MIN		1.62	12.02	1.44	1.46	1.03	1.03	1.01	1.02
MAX		6.60	94.42	205.21	139.72	233.02	242.76	247.49	248.99
$S_{\bar{X}}^2$		3.017 ^a	32.22 ^b	88.79 ^b	37.51 ^b	59.43 ^b	58.68 ^b	70.97 ^b	67.89 ^b
CV%		85.7	85.3	99.2	89.8	97.9	105.9	112.1	110.5
T	T-test	20.79 ^c	0.770	0.687	1.180	0.653	0.6367	0.476	0.689
		Random Method							
N		2	5	10	25	50	100	500	1000
μ		55.88	57.34	54.79	56.36	56.45	55.88	55.08	56.96
\bar{X}		36.35	31.53	55.49	62.19	55.78	40.41	55.95	53.22
S_X^2		934.90	1424.47	3140.85	3418.70	4857.27	2260.35	3670.11	3449.43
MIN		14.73	0.00	6.41	2.45	1.01	0.00	0.00	0.00
MAX		57.97	96.15	162.85	215.97	241.06	248.13	247.66	248.99
$S_{\bar{X}}^2$		25.72 ^b	45.17 ^b	56.60 ^b	54.97 ^b	87.07 ^b	55.93 ^b	65.59 ^b	64.81 ^b
CV%		84.1	119.7	101.0	94.0	124.9	117.6	108.3	110.4
T	T-test	0.903	1.529	0.039	0.497	0.068	3.524 ^d	0.318	2.019 ^c

^aSignificance at $p = 0.10$.

^bSignificance at $p = 0.0005$.

^cSignificance at $p = 0.05$.

^dSignificance at $p = 0.001$.

Table 7. Results from sampling a triangular distribution (where N gives the number of samples and μ is the population mean as calculated from a mean of 58.31 specified by the user)

N	Systematic Method							
	2	5	10	25	50	100	500	1000
μ	58.37	57.82	58.26	57.55	57.97	58.00	58.39	58.76
\bar{X}	69.84	57.49	59.28	60.44	60.49	53.26	58.59	59.5
$\frac{2}{S_X^2}$	214.99	229.45	725.99	543.04	702.69	573.60	518.95	564.86
MIN	59.47	34.3	19.77	19.09	2.60	1.96	5.53	2.67
MAX	80.21	71.7	98.23	103.91	109.46	101.78	110.93	113.31
$\frac{2}{S_X^2/\bar{X}}$	3.078 ^a	3.991 ^b	12.24 ^c	8.985 ^c	11.61 ^c	10.76 ^c	8.85 ^c	9.49 ^c
CV%	21.0	26.4	45.4	38.6	43.8	45.0	38.9	39.9
T t-test	1.106	0.050	0.119	0.619	0.669	1.977 ^d	0.193	0.974
N	Random Method							
	2	5	10	25	50	100	500	1000
μ	58.37	57.82	58.26	57.55	57.97	58.00	58.39	58.76
\bar{X}	73.35	47.42	57.14	59.57	52.78	55.76	55.92	56.94
$\frac{2}{S_X^2}$	850.95	309.29	963.00	604.33	459.66	647.41	640.14	621.29
MIN	52.72	28.14	4.17	0.00	7.51	0.00	0.00	0.00
MAX	93.97	65.4	100.65	97.55	98.02	101.53	112.46	113.31
$\frac{2}{S_X^2/\bar{X}}$	11.60 ^e	6.522 ^f	16.85 ^c	10.14 ^c	8.70 ^c	11.61 ^c	11.44 ^c	10.91 ^c
CV%	39.8	37.1	54.3	41.3	40.6	45.6	45.2	43.8
T T-test	0.726	1.323	0.112	0.409	1.713 ^a	0.881	2.180 ^d	2.320 ^d

^aSignificance at p = 0.10.

^bSignificance at p = 0.01.

^cSignificance at p = 0.0005.

^dSignificance at p = 0.05.

^eSignificance at p = 0.001.

^fSignificance at p = 0.005.

of nitrogen as indicated by the Student's t-test would be premature if the density function for a lognormal distribution is not considered.

Rejection of the hypothesis that the value of the mean for N samples is identical to the plot mean first occurs in the random method with 25 samples for attributes in the uniform distribution (Table 5). The t-tests of subsequent random samples produce acceptance with 95% confidence (for samples equal to 1000 in the normal, uniform, and lognormal distributions and equal to 500 and 1000 in the triangular distribution) and 99.9% (for samples equal to 100 in the lognormal distribution). Rejection occurs only for the systematic method, with 90% confidence for 100 samples for attributes in the triangular distribution (Table 7).

DISCUSSION

The results of the simulation show no pattern of significance in the differences between the true mean and the sample mean. Even with some of the small samples, the Students' t-tests do not show significant differences in the mean values. These results may be a function of the random number generator. The results of the t-tests must be considered in view of the underlying distribution of the data since the test assesses differences in the mean values but not in the distributions.

For a given sample size, the probability of rejecting the null hypothesis (α) when the hypothesis is true is related to the probability of not rejecting the null hypothesis when, in fact, it is false (β). This fact is related to the two types of error that can occur in hypothesis testing. For a given sample size, α is inversely related to β , with the result that a lower probability of accepting a Type I error is associated with a higher probability of committing a Type II error. Both types of errors can be reduced by increasing the sample size. The effects of sample size on the types of errors have been analytically explored for the normal distribution. Power curves show how an increase in sample size improves the value of β for a given α (e.g., Dixon and Massey 1969).

The results of the simulations suggest that the sample results depend on the skewness and kurtosis of the underlying distribution of the data. For large sample sizes, the sample mean of the lognormal distribution is less than the population mean because of the skewness of that distribution. The kurtosis of the underlying distribution seems to affect the fluctuations in the sample mean as the number of samples is varied. At one extreme, the uniform distribution has estimates that differ by more than ten units from the mean until the sample size is 25 (Fig. 6). At the other extreme, the mean in the normal distribution is never more than ten units different from the population mean (Fig. 5).

The implications for sampling soil nitrogen availability are not straightforward. The decrease in the ratio of the standard error to the mean ratio (Fig. 5-8) shows that precision does increase with sample number (precision is a function of the square root of N). Although there are no criteria for determining a particular number of samples that adequately characterizes the true distribution, the change in the ratio of the standard error to the mean can be used as a guide. The change in this ratio depends on the underlying distribution of the soil characteristic. The greatest change occurs with the lognormal distribution, and the least, with the uniform distribution. Therefore, if the underlying distribution of the soil property is known, it can be used to determine the number of samples that are needed.

No difference occurs between the simulated random and systematic samplings. This lack of difference is expected because there are no spatial differences in the distributions across the plot. Comparisons between systematic and random samplings for various vegetation types show that systematic sampling is likely to be less accurate than random sampling if the attribute has a strongly marked spatial pattern (Greig-Smith 1983). For example, tilling of a field can result in a predictable pattern of species abundance or size that is associated with the ridges and furrows (Harper 1977). Random sampling provides data that meets the assumption of randomness required for most statistical tests. Systematic sampling is preferred in field studies because less time and money are needed to place and relocate the samples at a later date (Greig-Smith 1983, p. 21).

The sampling simulation can be useful in exploring the implications of sampling design on test statistics. Not only can species or soil characteristics be varied, but also locational distributions can change. PLOTSAM is an easily used code that runs on an IBM PC or compatible machine. This documentation of the code should make the program more readily accessible.

REFERENCES

- Binkley, D., and P. Matson. 1983. Ion exchange resin bag method for assessing forest soil nitrogen availability. *Soil Sci. Soc. Am.* 47:1050-1052.
- Binkley, D., J. Aber, J. Pastor, and K. Nadelfoffer. 1986. Nitrogen availability in some Wisconsin forests: comparison of resin bags and on-site incubations. *Biol. Fertil. Soils* 2:77-82.
- Bloss, S. L. 1987. The effect of rooting activity by European wild boars on soil nitrogen availability and leaching. M.S. project. Duke University, Durham, North Carolina.
- Borland International. 1985. Turbo Pascal, Version 3.0, Reference. Scotts Valley, California.
- Croy, C. D., and R. L. Dix. 1984. Notes on sample size requirements in morphological plant ecology. *Ecology* 65:662-666.
- Dixon, W. J. and F. J. Massey, Jr. 1969. *Introduction to Statistical Analysis*. McGraw-Hill, New York.
- Gauch, H. G. 1973. The relationship between sample similarity and ecological distance. *Ecology* 54:618-622.
- Gibson, D. J. 1987. Spatial and temporal heterogeneity in soil nutrient supply measured using in-situ ion exchange resin bags. *Plant Soil* 96:445-450.
- Green, R. H. 1979. *Sampling design and statistical Methods for Environmental Biologists*. John Wiley & Sons, New York.
- Greig-Smith, P. 1983. *Quantitative Plant Ecology* 3d ed. Plenum Press, New York.
- Harper, J. L. 1977. *Population Biology of Plants*. Academic Press, New York.
- Hart, S. C., and D. Binkley. 1985. Correlations among indices of forest soil nutrient availability in fertilized and unfertilized pine plantations. *Plant Soil* 85:11-21.
- Lim, T. M., and H. W. Khoo. 1985. Sampling properties of Gower's general coefficient of similarity. *Ecology* 66:1682-1685.
- Pielou, E. C. 1969. *An Introduction to Mathematical Ecology*. John Wiley & Sons, New York.

Rubinstein, R. Y. 1981. Simulation and Monte Carlo Method.
John Wiley & Sons, New York.

Schildt, H. 1986. Advanced Turbo Pascal. Osborne McGraw-Hill,
Berkeley.

APPENDIX. LISTING OF THE PROGRAM PLOTSAM


```
{+C+,U+}
Program PlotSam;
```

```
Const
```

```
PlotSide = 50.0;
Nind     = 2500;
Nuspe   = 5;
Nvar     = 2;
IOVal    : Integer = 0;
IOErr    : Boolean = False;
printit  : Boolean = False;
```

```
Type
```

```
menustr  = string[80];
namlist  = string[8];
```

```
Var
```

```
cx       : Array [1..Nind] of real;
cy       : Array [1..Nind] of real;
species  : Array [1..Nind] of integer;
dbh      : Array [1..Nind] of real;

name     : Array [1..Nuspe] of namlist;
no       : Array [1..Nuspe] of integer;
meandbh  : Array [1..Nuspe] of real;
sd       : Array [1..Nuspe] of real;
min      : Array [1..Nuspe] of real;
max      : Array [1..Nuspe] of real;
dist     : Array [1..Nuspe] of char;

mucx     : Array [1..Nuspe] of real;
sdcx     : Array [1..Nuspe] of real;
cxmin    : Array [1..Nuspe] of real;
cxmax    : Array [1..Nuspe] of real;
cxdist   : Array [1..Nuspe] of char;

rho      : Array [1..Nuspe] of real;

mucy     : Array [1..Nuspe] of real;
sdcy     : Array [1..Nuspe] of real;
cymin    : Array [1..Nuspe] of real;
cymax    : Array [1..Nuspe] of real;
cydist   : Array [1..Nuspe] of char;

sscp     : array[1..nvar,1..nvar] of real;
ssum, smin, smax : array[1..nvar] of real;

seed,    {random # seed}
Nsams,   {total # of samples to be taken}
i,       {global, loop variable }
j,       {global, for transect sampling on}
TN       : integer;{total # of transects to be laid}
plotrunc, {integer value for plotside}
Nsp,     {total # of species in plot}
ncnt,    {counter }
samnum   : integer;{counter for # of times a plot is sampled}
SamSide, {length of side of sample}
sum,     {accumulator}
xncnt,   {change ncnt to real}
plotmean, {mean of plot}
```

```

plotfactor      : real;      {plotside**2 / samside**2}
anst,
ans,
samethod,
doc             : char;      {Document sampling?}

infile,outfile,device      : text;
filename                 : string[14];

{$I Random.inc}
{$I Menu.inc}
{$I Inout.inc}
{$I Setup.inc}
{$I Stat.inc}
{$I Tests.inc}
Procedure GeneratePlotData;
var
  cnt,i,j,ycnt : integer;

begin
  cnt := 0;
  ycnt := 0;
  for i := 1 to Nsp do begin
    for j := 1 to No[i] do begin
      cnt := cnt + 1;
      species[cnt] := i;

      case cxdist[i] of      {pick random x coordinants}
        'B' : repeat
          bivariate(mucx[i],mucy[i],sdcx[i],sdcy[i],
                    rho[i],cx[cnt],cy[cnt]);
        until (cx[cnt]>cxmin[i]) and (cx[cnt]<cxmax[i])
              and (cy[cnt]>cymin[i]) and (cy[cnt]<cymax[i]);
        'E' : SysUniform(cnt,plotside,cx[cnt],cy[cnt],ycnt);
        'N' : cx[cnt] := Normal(mucx[i],sdcx[i],cxmin[i],cxmax[i]);
        'L' : cx[cnt] := logNormal(mucx[i],sdcx[i],cxmin[i],cxmax[i]);
        'T' : cx[cnt] := triangular(mucx[i],cxmin[i],cxmax[i]);
        'U' : cx[cnt] := uniform (cxmin[i],cxmax[i]);

      end {case};

      case cydist[i] of      {pick random y coordinants}
        'N' : cy[cnt] := Normal(mucy[i],sdcy[i],cymin[i],cymax[i]);
        'L' : cy[cnt] := logNormal(mucy[i],sdcy[i],cymin[i],cymax[i]);
        'T' : cy[cnt] := triangular(mucy[i],cymin[i],cymax[i]);
        'U' : cy[cnt] := uniform (cymin[i],cymax[i]);

      end {case};
    end
  end
end

```

```

      Case dist[i] of          (pick random attributes)

        'N' : dbh[cnt] := Normal(meandbh[i],sd[i],min[i],max[i]);
        'L' : dbh[cnt] := logNormal(meandbh[i],sd[i],min[i],max[i]);
        'T' : dbh[cnt] := triangular (meandbh[i],min[i],max[i]);
        'U' : dbh[cnt] := uniform (min[i],max[i]);

      end {case};
    { if brackets are removed from Write statement below, the location
    { and value of each individual will be printed on screen          }
    { Writeln(cnt:3,' ',cx[cnt]:6:1,' ',cy[cnt]:6:1,' ',           }
    { dbh[cnt]:10:4); }
  end {j};
end {i};

end {GeneratePlotData};

Procedure Summarize;

var
  i : integer;
  sum : real;

begin
  plotfactor := (Plotside*Plotside) / (SamSide*SamSide);
  sum := 0.0;
  for i := 1 to Nind do
    sum := sum + dbh[i];
  plotmean := sum/Nind;
  GotoXY(1,2); ClrEol;
  Write('Simulation Summary: ');
  NormVideo;
  Writeln('No individ = ',Nind:5);
  Writeln('          No species = ',Nsp:5);
  Writeln('          No samples = ',Nsams:5);
  Writeln('          Plot size = ',PlotSide:8:2,' X ',PlotSide:8:
  Writeln('          Sample size = ',SamSide:8:2,' X ',SamSide:8:2);
  Writeln('          Plot ratio = ',Plotfactor:8:2);
  Writeln('          Plot sum = ',sum:8:2);
  Writeln('          Plot mean = ',plotmean:10:4);

  if printit then begin
    Writeln(1st);
    Write(1st,'Simulation Summary: ');
    Writeln(1st,'No individ = ',Nind:5);
    Writeln(1st,'          No species = ',Nsp:5);
    Writeln(1st,'          No samples = ',Nsams:5);
    Writeln(1st,'          Plot size = ',PlotSide:8:2,' X ',Plot
    Writeln(1st,'          Sample size = ',SamSide:8:2,' X ',SamSi
    Writeln(1st,'          Plot ratio = ',Plotfactor:8:2);
    Writeln(1st,'          Plot sum = ',sum:8:2);
    Writeln(1st,'          Plot mean = ',plotmean:10:4);
    Writeln(1st);
  end {if printit};
end {Summarize};

Procedure SamplePlot (p,j,TN:integer;samethod:char);

```

```

var
  sx, sy : real;
  i : integer;

begin
  CASE samethod of
    'S' : BEGIN

        sx := TRUNC(p - ((j - 1)*plotside));
        sy := TRUNC((j/(TN+1))*plotside);
        { if bracket are removed from Writeln below, }
        { the coordinates of the sample will be printed }
        { Writeln(1st,'sx = ',sx:6:2,' sy = ',sy:6:2) }
      END;
    'R' : BEGIN
        sx := urand(seed) * (PlotSide-SamSide);
        sy := urand(seed) * (PlotSide-SamSide);
      END
  END;

  for i := 1 to Nind do begin
    if (cx[i]<=(sx+SamSide)) and (cx[i]>=sx) and
       (cy[i]<=(sy+SamSide)) and (cy[i]>=sy) then begin

        sum := dbh[i] + sum;
        ncnt := ncnt + 1;

      end {if};
    end {i};
  end {SamplePlot};

  Procedure DocuSam(i:integer;xcnt,sum:real);

  Begin
    Write(1st,'S.N. = ',i:3);
    Writeln(1st,'      sum after SAMPLEPLOT = ',sum*xcnt:8:2);
    Writeln(1st,'      ncnt = xcnt after SAMPLEPLOT = ',xcnt:6:2);
    Writeln(1st,'      sum after sum/xcnt = ', sum:8:2)
  End; {DocuSam}

  begin
    (main routine)

    Initialize;
    SetUp;
    GeneratePlotData;

    ClrScr;
    Summarize;
    samnum := 0;
    Repeat
      ZeroArrays;

      Write('Sampling Method : <S>ystematic or <R>andom ?');
      Readln(samethod);
      samethod := Ucase(samethod);

      Menu(21,'Do you want sum of SAMPLEPLOT printed? <Y,N> ');
    
```

```

Read(kbd,doc);
doc := Upcase(doc);

Menu(21,'Conduct T and Chi-square Tests ? <Y,N>');
Read(kbd,anst); anst := Upcase(anst);

GotoXY(1,10);
NormVideo;
Write('Stats: ---Estimated Totals for Plot-----');
Writeln( '      ---Estimated Mean DBH per indiv ---');
Write('S.N.   Mean      sd      Min      Max      CV');
Writeln( '      Mean      sd      Min      Max      CV');

If printit or (doc = 'Y') then
begin
  Writeln(1st,'Stats: ---Estimated Mean per indiv--- ');
  Writeln(1st,'      Mean      sd      Min      Max      CV');
end {if printit};

Case samethod of
  'S' : Begin
    Writeln(1st,'SYSTEMATIC SAMPLING':25);
    plotrunc:= TRUNC(plotside);
    If (Nsams MOD plotrunc) = 0
      then TN := Nsams DIV plotrunc
      else TN := (Nsams DIV plotrunc) + 1
    End;
  'R' : Begin
    Writeln(1st,'RANDOM SAMPLING');
    TN := 1
    End;
End; {Case}

NormVideo;

j := 1;

For i := 1 to Nsams do
  Begin
    If i = (j * plotrunc) + 1
      then j := j + 1;
    ncnt := 0; xncnt := 0.0; sum := 0.0;
    SamplePlot(i,j,TN,samethod);
    xncnt := ncnt;
    if ncnt>0 then sum := sum/xncnt;
    Accumulate(xncnt*plotfactor,sum);
    If doc = 'Y'
      then DocuSam(i,xncnt,sum);
    GotoXY(1,12+samnum); ClrEol;
    Write(i:3);
  End; {for i}

GotoXY(5,12+samnum);
PrintStat (Nsams);
samnum := samnum+1; if (samnum>9) then samnum := 0;

If anst = 'Y'

```

```
    then Tests(Nsams,dist[1]);  
    Menu(21,'Sample plot again ? <Y,N>');  
    Read(kbd,ans); ans := Uppcase(ans);  
    Menu(21,'');  
    Menu(22,'');  
    Menu(23,'');  
    Menu(24,'');  
  
    Until ans='N';  
end.
```

```

Procedure Tests(N:integer;Dis:char);
VAR
  Smean,SV,tALPHA,T,VARMEAN,CHIVAL:REAL;
BEGIN
  Smean := ssum[2]/n;
  SV := Variance(2,2,N);
  T := ((Smean - plotmean) / sqrt(SV)) * sqrt(n);
  VARMEAN:= SV/Smean;
  Menu(22,'T test : ENTER CRITICAL VALUE FOR ALPHA=.05');
  Menu(23,' AND (N-1) DOF FOR TWO-SIDED HYPOTHESIS: ');
  READLN(tALPHA);
  Menu(24,'Chi-square : ENTER THE VALUE OF THE CHI-SQUARE/(N-1) DOF : ');
  READLN(CHIVAL);

  Writeln(LST,' ');
  WRITELN(LST,'          STUDENT T TEST FOR A          ');

  CASE Dis OF
    'U': WRITELN(LST,'          UNIFORM DISTRIBUTION of ATTRIBUTE');
    'N': WRITELN(LST,'          NORMAL DISTRIBUTION of ATTRIBUTE');
    'L': WRITELN(LST,'          LOGNORMAL DISTRIBUTION of ATTRIBUTE');
    'T': WRITELN(LST,'          TRIANGULAR DISTRIBUTION of ATTRIBUTE')
  END;
  Writeln(LST,'NO OF SAMPLES TAKEN = ',N:4);
  Writeln(LST,'NULL HYP: MEAN = ',plotmean:8:4);
  Writeln(LST,'ALTN HYP: MEAN<> ',plotmean:8:4);
  Writeln(LST,'TEST STAT: T = ',T:8:4);
  Writeln(LST,' WITH ',(N-1):3,' DEGREES OF FREEDOM');
  Writeln(LST,'REJ REGION: ABS(T) > ',tALPHA:8:4);
  Writeln(LST);
  Writeln(LST);
  IF ABS(T) > tALPHA
  THEN
    BEGIN {THEN}
      Writeln(LST,'T DOES FALL IN THE REJECTION REGION. ');
      WRITE(LST,'WE ARE 95% CONFIDENT THAT THE SAMPLE MEAN WILL NOT '
        Writeln(LST,'EQUAL THE TRUE (PLOT) MEAN. ')
    END {THEN}
  ELSE
    BEGIN {ELSE}
      Writeln(LST,'T DOES NOT FALL IN THE REJECTION REGION. ');
      WRITE(LST,'WE ARE 95% CONFIDENT THAT THE SAMPLE MEAN WILL EQUAL '
        Writeln(LST,'THE TRUE (PLOT) MEAN. ');
    END {ELSE}
  END;
  Writeln(LST);
  Writeln(LST,'          CHI-SQUARE TEST ');
  Writeln(LST,'SAMPLE VARIANCE = ',SV:10:4);
  Writeln(LST,'SAMPLE MEAN = ',Smean:8:2);

  Writeln(LST,'RATIO OF THE SAMPLE VARIANCE TO THE SMEAN = ',VARMEAN:8:4);
  Writeln(LST,'VALUE OF THE CHI-SQUARE/',(N-1):4,' DOF DISTRIBUTION = ',CH
  IF VARMEAN > CHIVAL
  THEN
    Writeln(LST,'THE DIST FOR THIS SAMPLE ISN" T RANDOM. ');
  IF VARMEAN =1.0
  THEN Writeln(LST,'THE DIST IS RANDOM. ')
  ELSE
    IF VARMEAN > 1.0
    THEN Writeln(LST,'THE DIST IS AGGREGATED. ')
    ELSE Writeln(LST,'THE DIST IS UNIFORM. ');

```

```
WRITELN(LST, '*****  
WRITELN(LST, '*****  
WRITELN(LST);  
DELAY(10000);  
END; {Tests}
```

```

Function Variance (i,j,n :integer) : real;

begin
  Variance := (sscp[i,j] - ssum[i]*ssum[j]/n)/(n-1);
end {function Variance};
  {equation produces a unbiased estimate for Variance}

Procedure ZeroArrays;

var
  i,j : integer;

begin
  ncnt := 0;
  for i := 1 to nvar do begin
    ssum[i] := 0.0;
    smin[i] := 1.E30;
    smax[i] := -1.E30;

    for j := 1 to nvar do
      sscp[i,j] := 0.0;

    end {for};
  end {ZeroArrays};

Procedure Accumulate (x1, x2 : real);

var
  i,j : integer;
  x : array [1..Nvar] of real;

begin
  x[1] := x1;      {xncnt * plotfactor}
  x[2] := x2;      {sum generated in SAMPLEPLOT}
  for i := 1 to Nvar do begin

    ssum[i] := ssum[i] + x[i];
      {ssum[1] = sum of # of hits in quadrat laid * plotfactor}
      {ssum[2] = sum of all samples}

    {If i = 2
     then
      Writeln(1st,' ssum in ACCUMULATE = ',ssum[2]:6:2);}
    if smin[i] > x[i] then smin[i] := x[i];
    if smax[i] < x[i] then smax[i] := x[i];

    for j := 1 to nvar do
      sscp[i,j] := sscp[i,j] + x[i]*x[j];
      {sscp[1,1] = sum of squares of x1}
      {sscp[1,2] = not used          }
      {sscp[2,1] = not used          }
      {sscp[2,2] = sum of the squares of x2}

    end {for};
  end {Accumulate};

Procedure Printstat (n : integer);

begin
  {Write out basic statistics}

```

```

for i := 1 to nvar do begin
  Write (ssum[i]/n:6:2,' ');
  if variance(i,i,n) > 0.0 then Write (sqrt(variance(i,i,n)):6:2,' ')
    else Write (' ');
  Write (smin[i]:6:2,' ',smax[i]:6:2,' ');
  if (ssum[i] <> 0.0) and (variance(i,i,n) > 0.0) then
    Write (sqrt(variance(i,i,n))/(ssum[i]/n)*100.0:6:1,'% ');
end {for};

if printit then begin
  Write(lst,' ');
  for i := 2 to nvar do begin
    Write (lst,ssum[i]/n:6:2,' ');
    if variance(i,i,n) > 0.0 then Write (lst,sqrt(variance(i,i,n)):6:2,' ')
      else Write (lst,' ');
    Write (lst,smin[i]:6:2,' ',smax[i]:6:2,' ');
    if (ssum[i] <> 0.0) and (variance(i,i,n) > 0.0) then
      Write (lst,sqrt(variance(i,i,n))/(ssum[i]/n)*100.0:6:1,'% ');
    end {for};
    writeln(lst);
  end {if};
end {Printstat};

```

```
Procedure Initialize;
```

```
var
```

```
  i : integer;
  ans : char;
```

```
begin
```

```
  seed      := 1231;
  ClrScr;
  GotoXY(1,8);
  Writeln (' Random number seed = ',seed,' ');
  Menu(9,'Change the value of seed <Y,N> ');
  Read(kbd,ans);
  If UpCase(ans)='Y' then
    GetInteger (seed,'Type in the new value of seed ',1,15555);
  Menu(10,'Printed output <Y,N> ');
  Read(kbd,ans);
  if UpCase(ans)='Y' then Printit := true;

  Nsams     := 50;
  SamSide   := 0.9;
  filename  := 'testt583.dat';

  Writeln(' ');
  Write('Enter # of species to sample for : ');
  Readln(Nsp);
```

```
  for i := 1 to Nsp do begin
```

```
    Write('Enter',i:4,' species" name : ');
    Readln(name[i]);
    Write('Enter',i:4,' species" mean : ');
    Readln(meandbh[i]);
    Write('Enter',i:4,' species" sd   : ');
    Readln(sd[i]);
    min[i]      := 0.0;
    max[i]      := 50.0;
    dist[i]     := 'N';
    No[i]       := Nind div Nsp;
    cxdist[i]   := 'E';
    cydist[i]   := 'E';
    cxmin[i]    := 0.0;
    cymin[i]    := 0.0;
    cxmax[i]    := PlotSide;
    cymax[i]    := PlotSide;
    mucx[i]     := Plotside/2.0;
    mucy[i]     := Plotside/2.0;
    sdcx[i]     := mucx[i];
    sdcy[i]     := mucy[i];
    rho[i]      := 0.0;
```

```
  end {for i};
```

```
end {Initialize};
```

```
Procedure SetSamples;
```

```
var
```

```
  ans : char;
```

```
begin
```

```

Repeat
  GotoXY(1,20); ClrEol;
  Writeln ('Number of Samples = ',Nsams:4,' Size of Plot = ',
           SamSide:8:4,' X ',Samside:8:4);
  Menu (22,'Change <N>umber of samples <S>ize of Plot or <Q>uit ');
  Read(kbd,ans); ans := UpCase(ans);

  if ans='S' then
    GetReal(SamSide,' New plot size = ',0.0,PlotSide/2.0);

  if ans='N' then
    GetInteger(Nsams,' Number of Samples = ',1,10000);

Until ans='Q';

end {SetSamples};

Procedure SetSpecies;

var
  i,j : integer;
  ans : char;

begin
  Repeat
    ClrScr; NormVideo;
    Writeln('Species Characteristics: ');
    Writeln;
    Writeln('          Name      Dist      MeanDbh      Sd      Min      Max')
    for i := 1 to Nsp do
      Writeln(i:4,' ',Name[i]:8,' ',dist[i],' ',Meandbh[i]:10:4,
              ' ',Sd[i]:10:4,' ',min[i]:10:4,' ',max[i]:10:4);
    GetInteger(j,'Type number to change -- zero to return ',0,Nsp);

    if (j in [1..Nsp]) then begin
      Repeat
        Menu(22,
             '<N>ame <D>ist <M>ean <S>t.dev. m<I>n ma<X> or <R>eturn ');
        Read(kbd,ans); ans := UpCase(ans);
        Menu(22,' ');

        Case ans of
          'N' : begin
            menu (23,'Type in new name ');
            Readln(name[j]);
            menu (23,' ');
            end;

          'D' : begin
            Repeat
              menu (23,'Select new distribution <N> <L> <U> <T> ');
              Readln(dist[j]);
            Until (dist[j] in ['N','L','U','T']);
            menu (23,' ');
            end;

          'M' : GetReal(meandbh[j],'new mean value = ',0.0,1000.0);
          'S' : GetReal(sd[j],'new standard dev = ',0.001,Meandbh[j]);
        end;
      end;
    end;
  end;
end;

```

```

        'I' : GetReal(min[j], 'new minimum = ', 0.0, Meandbh[j]);
        'X' : GetReal(max[j], 'new maximum = ', Meandbh[j], 1.E6);

        end {case};

        Until ans= 'R';
    end {if};
Until j=0;

end {SetSpecies};

Procedure SetLocStats;

var
    i,j : integer;
    choice,ans,card : char;
    working : string[80];
    piece : string[3];

begin
    Repeat
        ClrScr; NormVideo;
        Writeln('Species Location Characteristics: ');
        Writeln;
        Writeln('          Name  Loc Dist      Mean          Sd          Min          Max
for i := 1 to Nsp do begin
    Writeln(i:4, ' ', Name[i]:8, '      x ', cxdist[i], ' ', mucx[i]:10:4,
            ' ', Sdcx[i]:10:4, ' ', cxmin[i]:10:4, ' ', cxmax[i]:10:4);
    Write('          y ', cydist[i], ' ', mucy[i]:10:4,
            ' ', Sdcy[i]:10:4, ' ', cymin[i]:10:4, ' ', cymax[i]:10:4);
    if cydist[i]='B' then Writeln(' r = ', rho[i]:6:4) else Writeln;
end {for i};
GetInteger(j, 'Type number to change -- zero to return ', 0, Nsp);

    if (j in [1..Nsp]) then begin
        Repeat
            Menu(23, 'change <X>, <Y> ');
            Read(kbd, choice); choice := UpCase(choice);
            Until choice in ['X', 'Y'];
            Menu(23, ' ');
            str(j:3, piece);
            working := 'Now changing variable ' +
                piece + ', coordinant ' + choice;

            NormVideo;
            GotoXY(1, 24); Write(working);

            Repeat
                Menu(22,
                    '<D>ist <M>ean <S>t.dev. m<I>n ma<X> <C>orrelation or <R>eturn
                Read(kbd, ans); ans := UpCase(ans);
                Menu(22, ' ');

            Case ans of

                'C' : begin
                    menu (23, ' new value of rho = ');
                    Readln(rho[i]);
                    end;

```

```

'D' : begin
  menu (23,
    '<B>ivar <N>or <L>ognor <U>niform <T>riang <E>SysU
  Read(kbd,card); Card := UpCase(card);
  if choice='X' then cxdist[j] := card
    else cydist[j] := card;
  if card='B' then begin
    cxdist[j] := 'B';
    cydist[j] := 'B';
  end {if};
  if cxdist[j]<>cydist[j] then begin
    NormVideo; GotoXy(38,24);
    Write(^G,
      'Warning: X-Y distributions are dissimilar');
    Delay(3000);
  end {if};
  menu (23, ' ');
end;

'M' : begin
  if choice='X' then
    GetReal(mucx[j], 'new mean x value = ',
      0.0, Plotside)
  else
    GetReal(mucy[j], 'new mean y value = ',
      0.0, PlotSide);
  end;

'S' : begin
  if choice='X' then
    GetReal(sdcx[j], 'new sd x value = ',
      0.0, Plotside)
  else
    GetReal(sdcy[j], 'new sd y value = ',
      0.0, PlotSide);
  end;

'I' : begin
  if choice='X' then
    GetReal(cxmin[j], 'new min x value = ',
      0.0, cxmax[j])
  else
    GetReal(cymin[j], 'new min y value = ',
      0.0, cymax[j]);
  end;

'X' : begin
  if choice='X' then
    GetReal(cxmax[j], 'new mzx x value = ',
      cxmin[j], Plotside)
  else
    GetReal(cymax[j], 'new max y value = ',
      cymin[j], PlotSide);
  end;

end {case};

  Until ans= 'R';
end {if};
Until j=0;

```

```

end {SetLocStats};

Procedure SetFreq;

var
  i,j,sum : integer;
  f       : real;
  ans     : char;
  check   : boolean;

begin
  Repeat
    ClrScr;
    NormVideo;
    Writeln; Writeln;
    Writeln('    Species Frequency Distribution');
    Writeln;
    Writeln('No    Name    Number    Frequency');
    Writeln;
    sum := 0;
    for i := 1 to Nsp do begin
      f := no[i] / Nind;
      Writeln(i:2,'    ',name[i]:5,'    ',no[i]:5,'    ',f:8:2);
      sum := sum + no[i];
    end {for i};
    GotoXY(40,7); Write(' total amount = ',sum:8);
    GotoXY(40,8); Write('  no of indiv = ',Nind:8);
    GotoXY(40,9); Write('  difference = ',sum-Nind:8);

    GetInteger(j,'Type number to change -- zero to return ',0,Nsp);

    if (j in [1..Nsp]) then begin
      Repeat
        Menu (22,'Change Species <A>mount or <F>requency ');
        Read(kbd,ans); ans := UpCase(ans);
        Menu (22,' ');

        if ans='A' then
          GetInteger(No[j],' New amount = ',0,Nind);

        if ans='F' then begin
          GetReal(f,' New percent = ',0.0,100.0);
          No[j] := Round(Nind * f);
        end;
      Until ans in ['A','F'];
    end {if};

    sum := 0;
    check := true;
    for i := 1 to nsp do
      sum := sum + no[i];
    if (abs(sum-Nind)>1) then check := false;
    if not check then Write('^G');

  Until check and (j=0);

end {SetFreq};

```

```

Procedure SetUp;
  var
    ans : char;
  begin
    Repeat
      ClrScr;
      Menu( 4,'          <C>  Set Species Attributes (dbh stats)');
      Menu( 6,'          <X>  Set X-Y location statistics');
      Menu( 8,'          <F>  Set Species Frequencies ');
      Menu(10,'         <D>  Define Sample Methods');
      Menu(12,'         <I>  Input Parameters from Disk');
      Menu(14,'         <S>  Save Parameters to Disk');
      Menu(16,'         <R>  Run Simulation');
      Menu(20,'        <Q>  Quit Simulation');
      Writeln;
      Read (kbd,ans); ans := UpCase(ans);

      Case ans of

        'D'  : SetSamples;
        'X'  : SetLocStats;
        'I'  : ReadParms;
        'S'  : WriteParms;
        'F'  : SetFreq;
        'C'  : SetSpecies;
        'Q'  : Halt;

      end {case};
    Until ans='R';
  end {SetUp};

```

```
Procedure Menu(yloc : integer; choices : menustr);
```

```
var
```

```
  len,i   : integer;
```

```
begin
```

```
  GotoXY(1,yloc); ClrEol; LowVideo;
```

```
  len := length(choices);
```

```
  for i := 1 to len do begin
```

```
    if choices[i]='>' then LowVideo;
```

```
    Write(choices[i]);
```

```
    if choices[i]='<' then NormVideo;
```

```
  end {for};
```

```
end {Menu};
```

```
Procedure GetInteger(var Val : Integer; Msg : Menustr; Low,High : Integer);
```

```
begin
```

```
  {$I-,R-}           {turn off I/O, range checking}
```

```
  Repeat
```

```
    Menu(23,Msg);
```

```
    Read(Val);
```

```
  Until (IOresult=0) and (Low<=Val) and (Val<=High);
```

```
  {$I+,R+}
```

```
  Menu(23,' ');
```

```
end {GetInteger};
```

```
Procedure GetReal(var XVal : Real; Msg : Menustr; XLow,XHigh : Real);
```

```
begin
```

```
  {$I-,R-}           {turn off I/O, range checking}
```

```
  Repeat
```

```
    Menu(23,Msg);
```

```
    Read(XVal);
```

```
  Until (IOresult=0) and (XLow<=XVal) and (XVal<=XHigh);
```

```
  {$I+,R+}
```

```
  Menu(23,' ');
```

```
end {GetReal};
```

```

Function Urand(var IY : integer): real;
var
  i : integer;
  x : real;

  const
    m2 : integer = 0;
    itwo : integer = 2;

begin
  if (m2 = 0) then begin
    Randomize;
    for i := 1 to IY do
      x := random;
    m2 := 1;
  end {if};

  Urand := random;
end;                                     {end Urand}

Function Triangular (mean, minimum, maximum : real) : real;
var
  rv : real;

begin
  rv := urand(seed);
  if rv > (mean-minimum)/(maximum-minimum) then
    Triangular := maximum - sqrt((maximum-minimum)*(maximum-mean) *
      (1.0 - rv))
  else
    Triangular := minimum + sqrt((maximum-minimum) * (mean-minimum) * rv);
end {Triangular};

Function LogTriangular(mean, minimum, maximum : real) : real;

begin
  mean      := ln(mean);
  minimum   := ln(minimum);
  maximum   := ln(maximum);
  LogTriangular := Exp(Triangular (mean, minimum, maximum));
end {LogTriangular};

Function Uniform ( minimum, maximum : real) : real;

begin
  Uniform := Urand(seed)*(maximum - minimum) + minimum;
end {Uniform};

Function LogUniform (minimum,maximum : real) : real;

begin
  minimum   := ln(minimum);
  maximum   := ln(maximum);
  LogUniform := Exp(uniform(minimum,maximum));
end {LogUniform};

Function Normal (mean, stdev, minimum, maximum : real) : real;
var

```

```

    i      : integer;
    norsum : real;

begin
    norsum := 0.0;
    for i := 1 to 12 do
        norsum := norsum + urand(seed);
                                {norsum := norsum + Random;}
        norsum := (norsum - 6.0) * stdev + mean;
        if (norsum < minimum) or (norsum > maximum) then
            norsum := normal(mean, stdev, minimum, maximum);
        normal := norsum;
    end {normal};

Function LogNormal (mean,stdev,minimum,maximum : real) : real;

begin
    mean      := ln(mean);
    stdev     := ln(stdev);
    if (minimum<>0) then minimum := ln(minimum);
    maximum   := ln(maximum);
    LogNormal := Exp(normal(mean, stdev, minimum, maximum));
end {LogNormal};

Procedure BiVariate (xmu, ymu, sdx, sdy, rho : real; var x, y : real);

{ Box-Muller's transformation modified by Jansson in
  Jansson, Birger 1966. Random Number Generators, Victor
  Petersons Bokindustri Artiebolaag, Stockholm. 205 p. }

var
    e1, e2, v : real;

begin
    e1 := Urand(seed);
    e2 := Urand(seed);

    v := sqrt(-2.0 * ln(e1));
    x := xmu + sdx * v * (sqrt(1.0 - rho * rho) *
        cos(2.0 * pi * e2) + rho * sin(2.0 * pi * e2));
    y := ymu + sdy * v * sin(2.0 * pi * e2);
end;

Procedure SysUniform(ct:integer;side:real;var x,y:real;var ycnt:integer);

var    d:integer;

Begin
    d := TRUNC(side);

    If (ct MOD d) > 0
        then    x := ct MOD d
        else    x := side;

    If (ct MOD d) = 1
        then
            Begin    {increment y by 1}
                ycnt := ycnt + 1;
                y := ycnt
            End      { then increment y}

```

```
End;   else   y := ycnt;
```

```
{SI-,R-}      (InOut.inc)
```

```
procedure IOCheck;
```

```
    (This routine sets IOErr equal to IOresult, then sets
     IOFlag accordingly. It also prints out a message on
     the 24th line of the screen, then waits for the user
     to hit any character before proceeding.      )
```

```
var
```

```
    Ch                : Char;
```

```
begin
```

```
    IOVal := IOresult;
```

```
    IOErr := (IOVal <> 0);
```

```
    GotoXY(1,24); ClrEol;      { Clear error line in any case }
```

```
    if IOErr then begin
```

```
        Write(Chr(7));
```

```
        case IOVal of
```

```
            $01 : Write('File does not exist');
```

```
            $02 : Write('File not open for input');
```

```
            $03 : Write('File not open for output');
```

```
            $04 : Write('File not open');
```

```
            $05 : Write('Can't read from this file');
```

```
            $06 : Write('Can't write to this file');
```

```
            $10 : Write('Error in numeric format');
```

```
            $20 : Write('Operation not allowed on a logical device');
```

```
            $21 : Write('Not allowed in direct mode');
```

```
            $22 : Write('Assign to standard files not allowed');
```

```
            $90 : Write('Record length mismatch');
```

```
            $91 : Write('Seek beyond end of file');
```

```
            $99 : Write('Unexpected end of file');
```

```
            $F0 : Write('Disk write error');
```

```
            $F1 : Write('Directory is full');
```

```
            $F2 : Write('File size overflow');
```

```
            $FF : Write('File disappeared');
```

```
        else Write('Unknown I/O error: ',IOVal:3)
```

```
        end;
```

```
        Read(Kbd,Ch)
```

```
    end
```

```
end; { of proc IOCheck }
```

```
Procedure ReadParms;
```

```
var
```

```
    NoPl,APdim,FPdim,i,j : integer;
```

```
    ans: char;
```

```
begin
```

```
    Repeat
```

```
        Clrscr;
```

```
        Menu(5,'Default file for model parameters is '); Writeln(filename);
```

```
        Menu(6,'Change this name <Y,N> ');
```

```
        Read(kbd,ans); ans := UpCase(ans);
```

```
        If ans='Y' then begin
```

```
            Menu(8,'Input new file name ');
```

```
            Read(filename);
```

```
        end {if};
```

```
    Until ans in ['Y','N'];
```

```
    Assign(infile,filename);
```

```

IOcheck;
If IOval=0 then begin

    reset(infile);
    IOcheck;
end {if};

                                                                    {input data stream}

If IOval=0 then begin
    Readln(infile,nsams);
    Readln(infile,samSide);

    for i := 1 to Nsp do begin
        Readln(infile,name[i]);
        Readln(infile,sd[i]);
        Readln(infile,min[i]);
        Readln(infile,max[i]);
        Readln(infile,dist[i]);
        Readln(infile,No[i]);
        Readln(infile,cxdist[i]);
        Readln(infile,cydist[i]);
        Readln(infile,cxmin[i]);
        Readln(infile,cymin[i]);
        Readln(infile,cxmax[i]);
        Readln(infile,cymax[i]);
        Readln(infile,mucx[i]);
        Readln(infile,mucy[i]);
        Readln(infile,sdcx[i]);
        Readln(infile,sdcy[i]);
        Readln(infile,rho[i]);
    end;
end {if};
Close(infile);
IOcheck;

end {ReadParms};

Procedure WriteParms;

var
    i,j : integer;

begin
    ClrScr;
    Menu(5,'Default name for output file is '); Write(filename);
    Menu(6,'Change this name <Y,N> ');
    Read(kbd,ans); ans:=UpCase(ans);
    If ans='Y' then begin
        Menu(8,'New file name ? ');
        Read(filename);
    end {if};

    Assign(outfile,filename); IOcheck;
    Rewrite(outfile); IOcheck;

    If IOval=0 then begin
        Writeln(outfile,nsams);
        Writeln(outfile,samSide);

        for i := 1 to Nsp do begin
            Writeln(outfile,name[i]);

```

```
    Writeln(outfile,sd[i]);
    Writeln(outfile,min[i]);
    Writeln(outfile,max[i]);
    Writeln(outfile,dist[i]);
    Writeln(outfile,No[i]);
    Writeln(outfile,cxdist[i]);
    Writeln(outfile,cydist[i]);
    Writeln(outfile,cxmin[i]);
    Writeln(outfile,cymin[i]);
    Writeln(outfile,cxmax[i]);
    Writeln(outfile,cymax[i]);
    Writeln(outfile,mucx[i]);
    Writeln(outfile,mucy[i]);
    Writeln(outfile,sdcx[i]);
    Writeln(outfile,sdcy[i]);
    Writeln(outfile,rho[i]);
end;

end {if};
Close(outfile); IOcheck;

end {WriteParms};

{$I+,R+}
```


INTERNAL DISTRIBUTION

- | | | | |
|-------|------------------|--------|-------------------------------|
| 1. | S. M. Bartell | 20. | M. L. Poutsma |
| 2. | R. B. Clapp | 21. | D. E. Reichle |
| 3-7. | V. H. Dale | 22. | M. G. Turner |
| 8-13. | R. H. Gardner | 23. | W. Van Winkle |
| 14. | R. L. Graham | 24. | Central Research Library |
| 15. | S. G. Hildebrand | 25-39. | ESD Library |
| 16. | A. W. King | 40-41. | Laboratory Records Department |
| 17. | R. J. Luxmoore | 42. | Laboratory Records, RC |
| 18. | P. D. Parr | 43. | ORNL Patent Office |
| 19. | W. M. Post | 44. | ORNL Y-12 Technical Library |

EXTERNAL DISTRIBUTION

45. Thomas G. Bahr, Director, Office of Water Policy, U.S. Department of Interior, C. Street between 18th and 19th Street, NW, Washington, DC 20240
46. Ann M. Bartuska, Program Coordinator, U.S. Environmental Protection Agency/NCSU Acid Precipitation Program, North Carolina State University, Box 5186, Raleigh, NC 27650
47. David A. Bennett, Acid Deposition Assessment Staff, Office of Research and Development, RD-675, U.S. Environmental Protection Agency, West Tower, Room 743, 401 M Street, SW, Washington, DC 20460
48. William Brandes, Director, Center for Water Resources Research, The University of Tennessee, Knoxville, TN 37916
49. Robert W. Brocksen, Living Lakes Inc., 1090 Vermont Street, NW, Suite 510, Washington, DC 20005
50. Ralph Brooks, Tennessee Valley Authority, Knoxville, TN 37902
51. R. T. Brown, Water Systems Development Branch, Division of Water Resources, Tennessee Valley Authority, Norris, TN 37828
52. J. D. Buffington, Office of Biological Services, U.S. Fish and Wildlife Service, 1730 K Street, NW, Washington, DC 20240
53. Dewey L. Bunting, The University of Tennessee, Knoxville, TN 37916-1610
54. Thomas M. Burton, Department of Zoology, Michigan State University, East Lansing, MI 48824
55. J. Thomas Callahan, Associate Director, Ecosystem Studies Program, Room 336, 1800 G Street, NW, National Science Foundation, Washington, DC 20550
56. R. R. Colwell, Director, Maryland Biotechnology Institute, Microbiology Building, University of Maryland, College Park, MD 20742
57. W. E. Cooper, Department of Zoology, College of Natural Sciences, Michigan State University, East Lansing, MI 48824

58. Kenneth W. Cummins, Department of Fisheries & Wildlife, Oregon State University, Corvallis, OR 97331
59. Colbert E. Cushing, Ecosystems Department, Battelle-Pacific Northwest Laboratories, Richland, WA 99352
60. Ecological Analysis, Inc., Library, RD#2, Goshen Turnpike, Middletown, NY 10940
61. Stuart G. Fisher, Department of Zoology, Arizona State University, Tempe, AZ 85281
62. G. J. Foley, Director, Environmental Monitoring Systems Laboratory, MD-75, Research Triangle Park, NC 27711
- 63-67. R. L. A. Franklin, P.O. Box 2337-NSU, Thibodaux, LA 70310
68. Clayton S. Gist, Oak Ridge Associated Universities, Oak Ridge, TN 37830
69. Robert Goldstein, Environmental Assessment Department, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
70. C. P. Goodyear, National Fisheries Center-Leetown, U.S. Fish and Wildlife Service, Route 3, P.O. Box 41, Kearneysville, WV 25430
71. D. Heyward Hamilton, Office of Health and Environmental Research, U.S. Department of Energy, Washington, DC 20545
72. W. Frank Harris, Executive Officer to the Assistant Director for Biological, Behavioral, and Social Sciences, National Science Foundation, Washington, DC 20550
73. Ray Herrmann, Chief, Water Resources, Field Support Laboratory, National Park Service, 107 C Natural Resources, Colorado State University, Fort Collins, CO 80523
74. J. W. Huckabee, Manager, Ecological Studies Program, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
75. H.B.N. Hynes, Department of Biology, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
76. George Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
77. R. H. Kennedy, Environmental Laboratory, P.O. Box 631, Waterways Experiment Station, Vicksburg, MS 39180
78. Robert P. Lackey, U.S. Environmental Protection Agency, Corvallis, OR 97330
79. George H. Lauff, W. K. Kellogg Biological Station, Michigan State University, Hickory Corners, MI 49060
80. Harry Leland, U.S. Geological Survey, Water Resources Division, 345 Middlefield Road, Menlo Park, CA 94025
81. W. M. Lewis, Jr., Department of E.P.O. Biology, University of Colorado, Boulder, CO 80302
82. G. E. Likens, Director, The New York Botanical Garden, Institute of Ecosystem Studies, The Mary Flagler Cary Arboretum, Box AB, Millbrook, NY 12545
83. C. J. Mankin, Director, Oklahoma Geological Survey, The University of Oklahoma, 830 Van Vleet Oval, Room 163, Norman, OK 73019

84. Bernard Manowitz, Department of Energy and Environment, Brookhaven National Laboratory, Upton, Long Island, NY 11973
85. G. Richard Marzolf, Division of Biology, Kansas State University, Manhattan, KS 66506
86. Jack S. Mattice, Electric Power Research Institute, Palo Alto, CA 94303
87. Helen McCammon, Director, Ecological Research Division, Office of Health and Environmental Research, Office of Energy Research, ER-75, U.S. Department of Energy, Washington, DC 20545
88. J. Frank McCormick, The University of Tennessee, Knoxville, TN 37916-1610
89. Judith L. Meyer, Department of Zoology, University of Georgia, Athens, GA 30602
90. Joe Nix, Department of Chemistry, Ouachita Baptist University, Arkadelphia, AR 71923
91. L. J. Paulson, Lake Mead Limnological Research Center, University of Nevada, Las Vegas, NV 89154
92. John Peine, Uplands Field Research Laboratory, Great Smoky Mountains National Park, Twin Creeks Area, Gatlinburg, TN 37738
93. Ralph Perhac, Environmental Assessment Department, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
94. Paul G. Risser, Vice President for Research, The University of New Mexico, Scholes Hall 108, Albuquerque, NM 87131
95. Richard J. Ruane, Tennessee Valley Authority, Chattanooga, TN 37401
96. E. J. Rykiel, Office of Health and Environmental Research, U.S. Department of Energy, Washington, DC 20545
97. Carl Schofield, Department of Natural Resources, Cornell University, Ithaca, NY 14850
98. J. A. Stanford, University of Montana Biological Station, East Shore, Bigfork, MT 59911
99. Wayne Swank, Coweeta Watershed Experimental Station, U.S. Forest Service, Route 1, Box 216, Otto, NC 28736
100. G. Swartzman, Center for Quantitative Science, University of Washington, Seattle, WA 98195
101. K. W. Thornton, FTN & Associates, P.O. Box 927, Vicksburg, MS 39180
102. Robin Vannote, Stroud Water Research Center, R.D. #1, Box 512, Avondale, PA 19311
103. E. B. Welch, 306 Moore Hall, University of Washington, Seattle, WA 98105
104. R. G. Wetzel, W. K. Kellogg Biological Station, Michigan State University, Hickory Corners, MI 49060
- 105-109. P. S. White, Department of Biology, University of North Carolina, Chapel Hill, NC 27514

110. Frank J. Wobber, Ecological Research Division, Office of Health and Environmental Research, Office of Energy Research, ER-75, U.S. Department of Energy, Washington, DC 20545
111. Office of Assistant Manager for Energy Research and Development, Oak Ridge Operations, P.O. Box 2001, U.S. Department of Energy, Oak Ridge, TN 37831
- 112-121. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831