



University of Nebraska at Omaha  
DigitalCommons@UNO

Publications Archives, 1963-2000

Center for Public Affairs Research

7-1978

# A Computerized Demonstration of the Central Limit Theorem in Statistics

Paul S. T. Lee

*University of Nebraska at Omaha*

Follow this and additional works at: <https://digitalcommons.unomaha.edu/cparpubarchives>

 Part of the [Public Affairs Commons](#), and the [Statistical Theory Commons](#)

## Recommended Citation

Lee, Paul S. T., "A Computerized Demonstration of the Central Limit Theorem in Statistics" (1978). *Publications Archives, 1963-2000*.  
4.  
<https://digitalcommons.unomaha.edu/cparpubarchives/4>

This Article is brought to you for free and open access by the Center for Public Affairs Research at DigitalCommons@UNO. It has been accepted for inclusion in Publications Archives, 1963-2000 by an authorized administrator of DigitalCommons@UNO. For more information, please contact [unodigitalcommons@unomaha.edu](mailto:unodigitalcommons@unomaha.edu).



A COMPUTERIZED DEMONSTRATION OF THE  
CENTRAL LIMIT THEOREM IN STATISTICS

Paul S.T. Lee, Ph.D.

The publication of this manual was financed in part through a grant from the Improvement of Instruction Committee, Faculty Senate of the University of Nebraska at Omaha.

## PREFACE

This manual was developed from the author's experience in teaching statistical inference and research techniques. The purpose of this manual is to demonstrate the concept of the Central Limit Theorem in graphic and tabular forms through the use of pre-prepared computer programs. The demonstration consists of five computer programs written in Fortran G and designed to be run on the IBM 360/65 with OS (Operating System). The cost of running these programs is relatively minimal. It is hoped that students without prior knowledge in computer science would use the manual, with assistance from the instructor, to run programs and to verify the theorem as a class assignment. Those who have had some prior knowledge would use these programs in more creative ways in learning statistics.

The author acknowledges the Faculty Senate Committee for Improvement of Instruction, and the Dean's Office, College of Public Affairs and Community Service at the University of Nebraska at Omaha for its financial support for the publication of this manual. Deepest appreciation is due to Drs. Margaret Gessaman, David Hinton, Larry Stephens, Angela Edwards and Gaylon Oswalt for their valuable suggestions at the initial stage of preparing this manual. Special thanks are due to Chuck Sundermeier who prepared the major part of the computer programs, to Linda Ferring who edited the manuscript, Scott Samson who designed the cover page, Joyce Carson and Beverly Walker who typed and finalized the manual.

# TABLE OF CONTENTS

	Page
PREFACE . . . . .	i
LIST OF TABLES AND FIGURES . . . . .	iv
1. INTRODUCTION . . . . .	1
1.1 Purpose and Organization of this Manual . . . . .	1
1.2 The Central Limit Theorem in Statistics . . . . .	1
1.3 Problems in Conducting Sampling Experiments . . . . .	2
2. THE CONSTRUCTION OF THE POPULATION . . . . .	5
2.1 Sample vs. Population . . . . .	5
2.2 The Construction of Population . . . . .	5
2.3 The Sampling Scheme . . . . .	5
2.4 Notations . . . . .	8
3. A COMPUTERIZED SAMPLING EXPERIMENT . . . . .	9
3.1 The Preparation of a Computer Program . . . . .	9
3.2 A Tabular and Graphic Demonstration of the Central Limit Theorem . . . . .	10
3.3 The Verification of the Central Limit Theorem - A Computerized Demonstration . . . . .	15
4. CONCLUSION . . . . .	17
APPENDIXES . . . . .	19
A. PROGRAM FOR LEESAMPL . . . . .	21
B. PREPARATION OF INPUT DATA FOR LEESAMPL . . . . .	27
C. PROGRAMS FOR (1) LEEDATA, (2) LEEPLOT AND (3) LEEGRAF1 . . . . .	31
D. JOB CONTROL LANGUAGE FOR (1) LEESAMPL, (2) (2) LEESORT, (3) LEEDATA, (4) LEEPLOT AND (5) LEEGRAF1 . . . . .	35
E. ESTIMATE OF COST OF RUNNING SAMPLING PROGRAMS . . . . .	37

## LIST OF TABLES AND FIGURES

Tables	Page
2.2.1 AGES OF STUDENTS IN A CLASS: FREQUENCY AND CHARACTERISTICS OF POPULATION . . . . .	6
3.2.1 MEAN AGE OF STUDENTS IN A CLASS: DISTRIBUTION OF SAMPLE MEANS USING $n=2$ . . . . .	11
3.2.2 MEAN AGE OF STUDENTS IN A CLASS: DISTRIBUTION OF SAMPLE MEANS USING $n=5$ . . . . .	13
3.3.1 A COMPARISON OF STATISTICS AND PARAMETERS . . . . .	16
Figures	
2.2.1 AGES OF STUDENTS IN A CLASS: HISTOGRAM OF PARENT POPULATION . . . . .	7
3.2.1 MEAN AGE OF STUDENTS IN A CLASS: DISTRIBUTION OF SAMPLE MEANS FOR $n=2$ . . . . .	12
3.2.2 MEAN AGE OF STUDENTS IN A CLASS: DISTRIBUTION OF SAMPLE MEANS FOR $n=5$ . . . . .	14

## 1. INTRODUCTION

### 1.1 Purpose and Organization of this Manual

The Central Limit Theorem is one of the most important concepts in statistics. It provides a link between sample statistics and population parameters. It is a basic concept for understanding various hypothesis-testing techniques such as Student's t-distribution,  $\chi^2$ -distribution and F-distribution.

This theorem, however, is almost impossible to verify in the classroom, even with the use of a calculator. As a result, students often become confused and frustrated in their attempts to understand this fundamental part of the study of statistical inference.

For the past decade, the use of computers in education has stimulated a great deal of interest and computer facilities have been made easily accessible to both students and instructors. Computer softwares (programs) have been prepared for use by both math and non-math majors.

The computer has excellent potential for meeting the needs of students being introduced to the Central Limit Theory. The purpose of this manual is to develop a series of computer softwares capable of conducting sampling experiments for the verification of the Central Limit Theorem.

The remainder of this presentation includes a discussion of the problems experienced by students when first introduced to the Central Limit Theorem, a description of a computer program which may be used to verify this theorem, a demonstration of the use of the computer program for conducting a sampling experiment and a concluding discussion of the student's application of this method to assist in understanding one of the most important concepts in statistics.

### 1.2 The Central Limit Theorem in Statistics

The Central Limit Theorem is generally stated as follows:

Let a random variable  $y$  of  $N$  observations follow a distribution with population mean  $\mu$  and standard deviation  $\sigma$ . As the sample size,  $n$ , increases, the distribution of means,  $\bar{y}$ , of all possible samples of the same size approaches a normal distribution with mean equal to the population mean  $\mu_{\bar{y}} = \mu$ , and a standard deviation (or standard error) equal to  $\sigma_{\bar{y}} = \sigma/\sqrt{n}$ .<sup>1</sup>

Greek letters are used for the population parameters and English letters for sample statistics.

The theorem identifies a definite relationship between a population and its samples. More specifically, it contains the following three suppositions:

1. The distribution of the means of all possible samples approaches a normal distribution as the sample size increases, regardless of the distribution of the population itself.
2. The mean of the means of all possible samples of the same size drawn from any population is equal to the mean of that population,  $\mu_{\bar{y}} = \mu$ .
3. The variance among means of the samples is equal to the population variance divided by the size of the sample,  $\sigma_{\bar{y}}^2 = \sigma^2/n$ .

The third supposition needs to be modified by including finite population corrections (fpc) if the sampling fraction exceeds 5% of the population. In this case, the variance of the sample means is equal to the population variance divided by the size of the sample and multiplied by fpc,  $\frac{N-n}{N-1}$  or  $\sigma_{\bar{y}}^2 = \frac{\sigma^2}{n} \frac{N-n}{N-1}$ .

The implication of this theorem is that the probability or relative frequency of the sample means within any particular interval is already known before a single sample from a given population is drawn, provided that the mean and variance of that population are known. However, if the mean and variance of the population were already known, there might not be any statistical problems at all. The significance of the theorem is not the actual values of statistics and parameters<sup>3</sup> but the relationship between them. This relationship provides a theoretical basis for various statistical analysis techniques.

### 1.3 Problems in Conducting Sampling Experiments

The Central Limit Theorem can be verified through conducting a series of sampling experiments. However, students and instructors alike often find it an impossible task to verify the theorem because it generally requires a long process of drawing all possible samples and computing and compiling all the statistics from these samples. For example, for a hypothetical population of 30 observations ( $N=30$ ) with a sample size of 5 ( $n=5$ ), a total of 142,506 distinct samples have to be drawn and the same number of means and variances have to be obtained and compiled for the verification of the theorem.<sup>4</sup> The total number of all possible samples increases to 30,045,015 if the size of the sample increases to 10.

With the assistance of a calculator, conducting such a sampling experiment would take several man-months work. Consequently, two types of compromise have been worked out: the mini-universe model and the fractional random sampling model. The former uses a mini-universe of 3 to 5 observations and conducts a complete sampling experiment using sampling with replacement techniques such as the one presented by Li.<sup>5</sup> The major shortcoming of the mini-universe model is that it is not realistic. In actual sampling situations, one seldom draws a sample that is larger in number of observations than the population from which the sample was taken. The fractional random sampling model generally uses a larger population

but conducts only a fractional random sampling such as the one used by Bhattacharyya.<sup>6</sup> This model is an approximation; it is an incomplete sampling experiment. Hence, it does not prove or disprove the Central Limit Theorem.

---

<sup>1</sup>See R. Lowell Wine, *Beginning Statistics* (Cambridge: Winthrop Publishers, 1976), pp. 184-186.

<sup>2</sup>William G. Cochran, *Sampling Techniques* (New York: John Wiley & Sons, Inc., Third Printing, 1960) pp.17-18.

<sup>3</sup>*Statistics* refers to the characteristics of the sample, which change from sample to sample; *parameters* refers to the characteristics of the population, which is a fixed variable.

<sup>4</sup>The formula used to derive these numbers is discussed in Section 2.3.

<sup>5</sup>Jerome C.R. Li, *Statistical Inference*, Volume I (Ann Arbor, Michigan: Edwards Brothers, Inc., 1968), pp.36-49.

<sup>6</sup>Gouri K. Bhattacharyya and Richard A. Johnson, *Statistical Concepts and Methods* (New York: John Wiley & Sons, 1977), pp. 210-220.





## 2. THE CONSTRUCTION OF THE POPULATION

### 2.1 Sample vs. Population

Much of the statistician's technical vocabulary is made up of common words given special meanings. For example, a "population" is a set of observations, or recorded information such as the heights of the students in a college. A "sample" is a collection of observations drawn from the population.

If the population is already known there may not be any need for a sample. If the population is not known, what is said to be true of the population is derived from a portion of it, or a sample.

In another example, if a manufacturer of light bulbs wished to know the average life span of his product, he could not burn out his entire stock for the sake of obtaining an answer. He must be content with testing a sample. The use of information obtained from a sample to draw conclusions about his entire production of light bulbs is called "statistical inference." The understanding of the relationship between a population and its samples is the first step in learning statistical inference.

### 2.2 The Construction of Population

For conducting a sampling experiment to demonstrate the relationship between a population and its samples as stated in the Central Limit Theorem, a population of 30 observations was constructed. These observations are recorded ages of students enrolled in a sophomore economics class at the University of Nebraska at Omaha, Spring semester, 1978. The frequency distribution and major characteristics of the population are shown in Table 2.2.1 and its histogram is shown in Figure 2.2.1.

It can be seen that the mean age of the population is 23 and the variance is 7.7. Its relatively flat histogram (Figure 2.2.1) indicates that the population does not follow a normal distribution.

### 2.3 The Sampling Scheme

The relation between a population and its samples refers to all the samples drawn from the population. There are many ways samples can be drawn. The sampling scheme used to draw all the samples of the same size from a given population is called sampling without replacement. Briefly, sampling without replacement is a method of selecting  $n$  number of observations such that when an observation has been drawn from the population, there is no chance the same observation could enter the same sample more than once. In contrast to sampling without replacement,

TABLE 2.2.1

AGES OF STUDENTS IN A CLASS:  
FREQUENCY AND CHARACTERISTICS OF POPULATION

Age ( $y$ )	Frequency ( $f$ )	$f_y$	$y - \mu$	$(y - \mu)^2$	$f(y - \mu)^2$
18	1	18	-5	25	25
19	2	38	-4	16	32
20	4	80	-3	9	36
21	3	63	-2	4	12
22	4	88	-1	1	4
23	4	92	0	0	0
24	2	48	+1	1	2
25	3	75	+2	4	12
26	3	78	+3	9	27
27	2	54	+4	16	32
28	2	56	+5	25	50
Total	30	$\Sigma f_y = 690$	0		$\Sigma f(y - \mu)^2 = 232$

$$\text{Mean } \mu = \frac{\Sigma f_y}{N} = \frac{690}{30} = 23$$

$$\text{Variance } \sigma^2 = \frac{\Sigma f(y - \mu)^2}{N} = \frac{232}{30} = 7.7$$

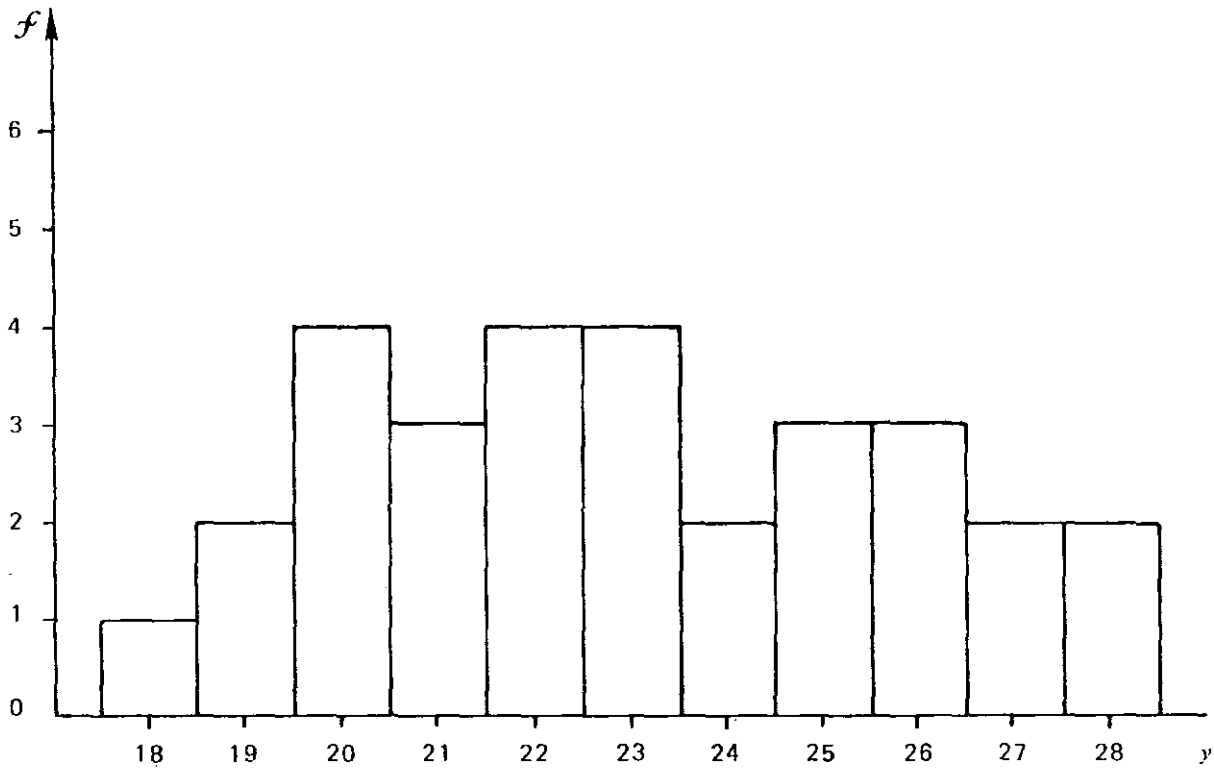


FIGURE 2.2.1: AGES OF STUDENTS IN A CLASS:  
HISTOGRAM OF PARENT POPULATION

the method of sampling with replacement is seldom used except in special circumstances, since there seems little point in having the same observation twice in the sample.

By using the scheme of sampling without replacement, the total number of distinct samples of size  $n$  that can be drawn from  $N$  observations in the population is given by the combinatorial formula:<sup>7</sup>

$$\binom{N}{n} = {}^N C_n = \frac{N!}{n! (N - n)!} \quad (2.3.1)$$

For example, if there are 30 observations ( $N=30$ ) in the population, and the sample size is 2 ( $n=2$ ), one will obtain a total of 435 distinct samples by applying these values to the above equation:

$$\begin{aligned} 30C_2 &= \frac{30 \times 29 \times \cancel{28} \times \cancel{27} \times \dots \times \cancel{2} \times \cancel{1}}{2 \times 1 \times \cancel{28} \times \cancel{27} \times \dots \times \cancel{2} \times \cancel{1}} \\ &= \frac{870}{2} \\ &= 435. \end{aligned}$$

If the sample size  $n$  increases to 5, the total number of all possible samples increases to 142,506.

## 2.4 Notations

In the foregoing sections, three types of means have been mentioned: the population mean, the sample mean and the mean of sample means. And two kinds of variances have been considered: population variance and the variance of sample means. To avoid possible confusion, the mathematical notations of the means and variances are listed in the following table:

<u>Distribution</u>	<u>Mean</u>	<u>Variance</u>	<u>Standard Deviation/Error</u>	<u>Number of Items</u>
Population	$\mu$	$\sigma^2$	$\sigma$	N (Size of population)
Sample	$\bar{y}$	$s^2$	$s$	n (Size of sample)
Sample means	$\mu_{\bar{y}}$	$\sigma_{\bar{y}}^2$	$\sigma_{\bar{y}}$ (Standard error)	${}^N C_n$ (No. of all possible samples)

<sup>7</sup>William G. Cochran, *Sampling Techniques* (New York: John Wiley & Sons, Inc., Third Printing, 1960), pp. 11-12.

### 3. A COMPUTERIZED SAMPLING EXPERIMENT

#### 3.1 The Preparation of a Computer Program

A computer program has been prepared for conducting the sampling experiment described in Section 2.3. This program contains 5 sub-programs, each designated by a particular name. All of them are written in Fortran G and are designed to be run on the IBM 360/65 with OS (Operating System). This section presents a detailed description of each of these sub-programs including input data requirements, output format and their major functions.

- a. LEESAMPL -- The program LEESAMPL is designed to draw all possible samples using a sampling without replacement scheme from a given population of up to 50 observations with incremental sample size of up to 10. Approximately 100,000 samples may be drawn in one run of LEESAMPL without requesting special scheduling of the program. If the total number of all possible samples exceeds the capacity of one run, the program stops and gives information needed to continue the sampling process for another run. In other words, if the total number of all possible samples is less than 100,000 only one run of LEESAMPL is required; if it is greater than 100,000, as many runs as needed to draw the samples are required by drawing 100,000 per run. The output from each run is stored in a separate file for use by the succeeding procedure, LEESORT. The output of LEESAMPL may be printed optionally. The detailed procedure for LEESAMPL is shown in Appendix A, the preparation of the input data and its card format are shown in Appendix B, the programs for the sub-programs in Appendix C and needed Job Control Language (JCL) for running are in Appendix D.
- b. LEESORT -- An IBM sort utility is used in LEESORT to sort the data produced by LEESAMPL so that the data can be summarized into a distribution table. One run of LEESORT is needed for each run of LEESAMPL. No input data are required except the sort utility control card which is shown along with LEESORT JCL in Appendix D.
- c. LEEDATA -- The program LEEDATA produces a summary from the sorted data output by LEESORT. Like LEESORT, one run of LEEDATA is required for each run of LEESAMPL. No input data are needed to run LEEDATA except

the LEEDATA JCL deck shown in Appendix D. The output of LEEDATA has two components: A listing of the frequency distribution table and a punched deck of cards containing the data shown in the frequency table. The program for LEEDATA is shown in Appendix C.

- d. LEEPLOT -- The major function of the program LEEPLOT is to prepare a file of plot data obtained from the combined output of LEEDATA runs. This file will be used as input data for the succeeding procedure, LEEGRAF1. LEEPLOT also uses the IBM sort utility. The required JCL for running this procedure is shown in Appendix D. The program for LEEPLOT is shown in Appendix C.
- e. LEEGRAF1 -- The program LEEGRAF1 produces graphs of the frequency distribution of all sample means, a list of the frequency distribution of the sample means and the mean of all possible sample means and the standard error of sample means. The JCL required for running this procedure is shown in Appendix D. As currently written, the plotter is a Complot Model PTC-4. With slight modification, the program could be easily changed to use a different plotter. The program for LEEGRAF1 is shown in Appendix C.

### 3.2 A Tabular and Graphic Demonstration of the Central Limit Theorem

After the programs and their respective JCL are prepared, the sampling experiment may be conducted through various runs of these programs. This section presents the results of two sampling experiments using the population given in Table 2.2.1. The first experiment uses a sample size of 2,  $n=2$ , and the second uses a sample of 5,  $n=5$ . The program LEESAMPL draws all possible samples, computes all sample means and stores them on a tape. The total number of all possible samples is 435 for  $n=2$  and 142,506 for  $n=5$  (Equation 2.3.1). The program LEESORT sorts the samples and the program LEEDATA summarizes these samples and their means in a frequency table. The program LEEPLOT prepares a data file for graphical presentation. Finally, the program LEEGRAF1 completes the experiment by giving a graph of all sample means.

Table 3.2.1 shows the frequency distribution of all sample means generated from the sampling experiment on  $n=2$ . It is a direct output of LEEDATA. The total number of samples is 435, and the standard deviation of sample means (standard error) is 1.9344. Table 3.2.2 shows the frequency distribution of sample means generated by LEEDATA on the sampling experiment using  $n=5$ . It can be seen at the bottom of the table that the total number of possible samples increases to 142,506 and the standard error reduces to 1.1547.

Figure 3.2.1 shows the graph of the sample means using  $n=2$ . This graph is a direct output of LEEGRAF1. One can verify it by drawing a histogram using the information given in Table 3.2.1. Similarly, Figure 3.2.2 is a direct output of LEEGRAF1 from a sampling experiment on  $n=5$ .

TABLE 3.2.1

MEAN AGE OF STUDENTS IN A CLASS:  
DISTRIBUTION OF SAMPLE MEANS USING  $n=2$

DISTRIBUTION OF SAMPLE MEANS

MEAN	FREQUENCY		
19.5	2.		
19.0	5.		
19.5	11.		
20.0	16.		
20.5	24.		
21.0	29.		
21.5	33.		
22.0	35.		
22.5	42.		
23.0	41.		
23.5	41.		
24.0	39.		
24.5	32.		
25.0	25.		
25.5	21.		
26.0	13.		
26.5	12.		
27.0	7.		
27.5	4.		
28.0	1.		
SUM OF DIFFERENCES SQUARED		.162400000E 04	04
TOTAL NUMBER OF SAMPLES		.435000000E 03	03
S SQUARED		.3741934776E 01	01
STD DEV		.1934408188E 01	01



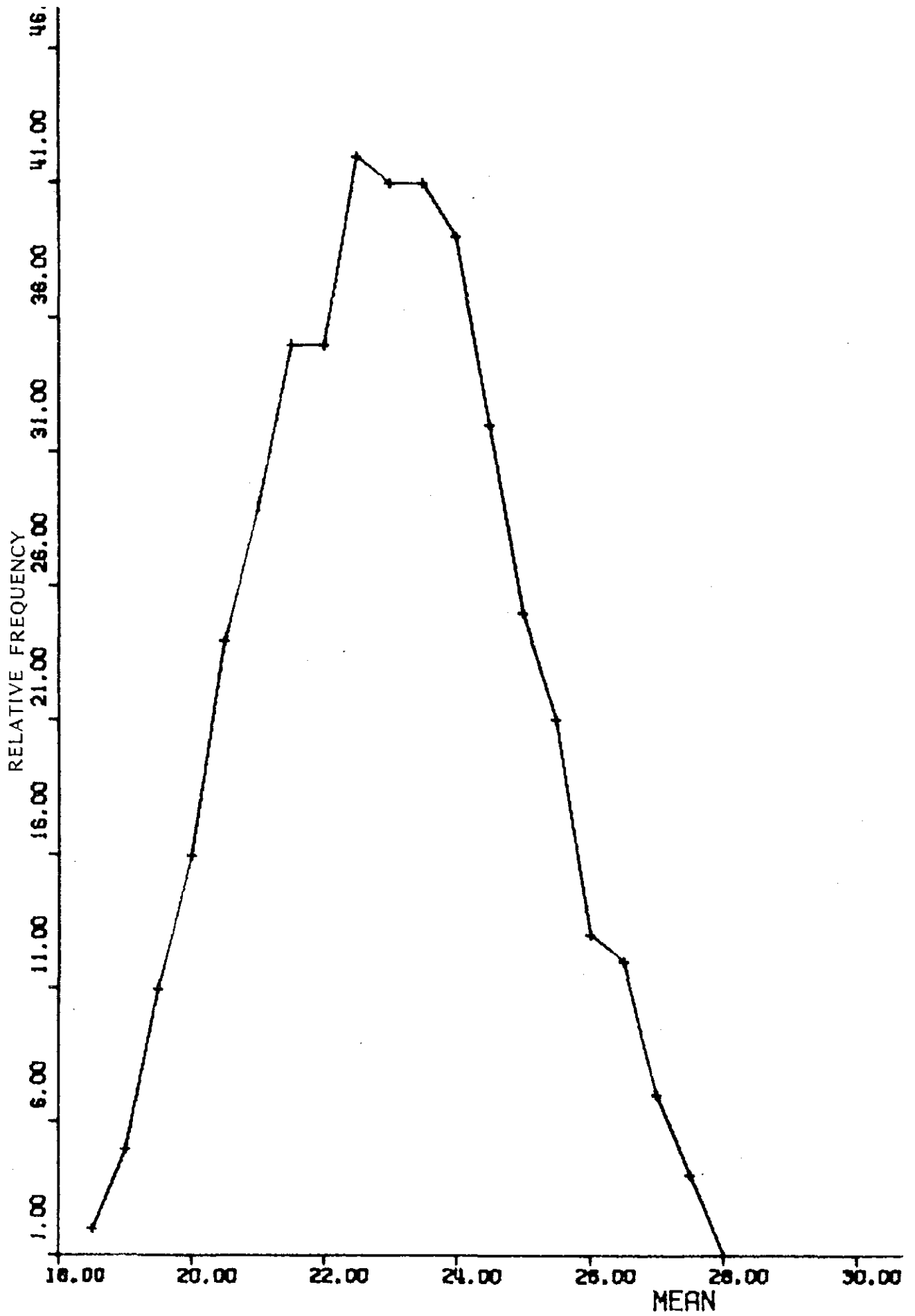


FIGURE 3.2.1: MEAN AGE OF STUDENTS IN A CLASS:  
DISTRIBUTION OF SAMPLE MEANS FOR  $n=2$

TABLE 3.2.2

MEAN AGE OF STUDENTS IN A CLASS:  
DISTRIBUTION OF SAMPLE MEANS USING n=5

DISTRIBUTION OF SAMPLE MEANS

MEAN	FREQUENCY
19.2	5.
19.4	20.
19.6	60.
19.8	132.
20.0	266.
20.2	462.
20.4	767.
20.6	1163.
20.8	1699.
21.0	2356.
21.2	3143.
21.4	4021.
21.6	5000.
21.8	5972.
22.0	6946.
22.2	7814.
22.4	8556.
22.6	9108.
22.8	9465.
23.0	9533.
23.2	9389.
23.4	8974.
23.6	8373.
23.8	7586.
24.0	6718.
24.2	5764.
24.4	4825.
24.6	3907.
24.8	3079.
25.0	2332.
25.2	1729.
25.4	1216.
25.6	836.
25.8	545.
26.0	336.
26.2	195.
26.4	110.
26.6	53.
26.8	27.
27.0	9.
27.2	3.
SUM OF DIFFERENCES SQUARED	.1700071875E 06
TOTAL NUMBER OF SAMPLES	.1425060000E 06
S SQUARED	.1333336830E 01
STD DEV	.1154702187E 01

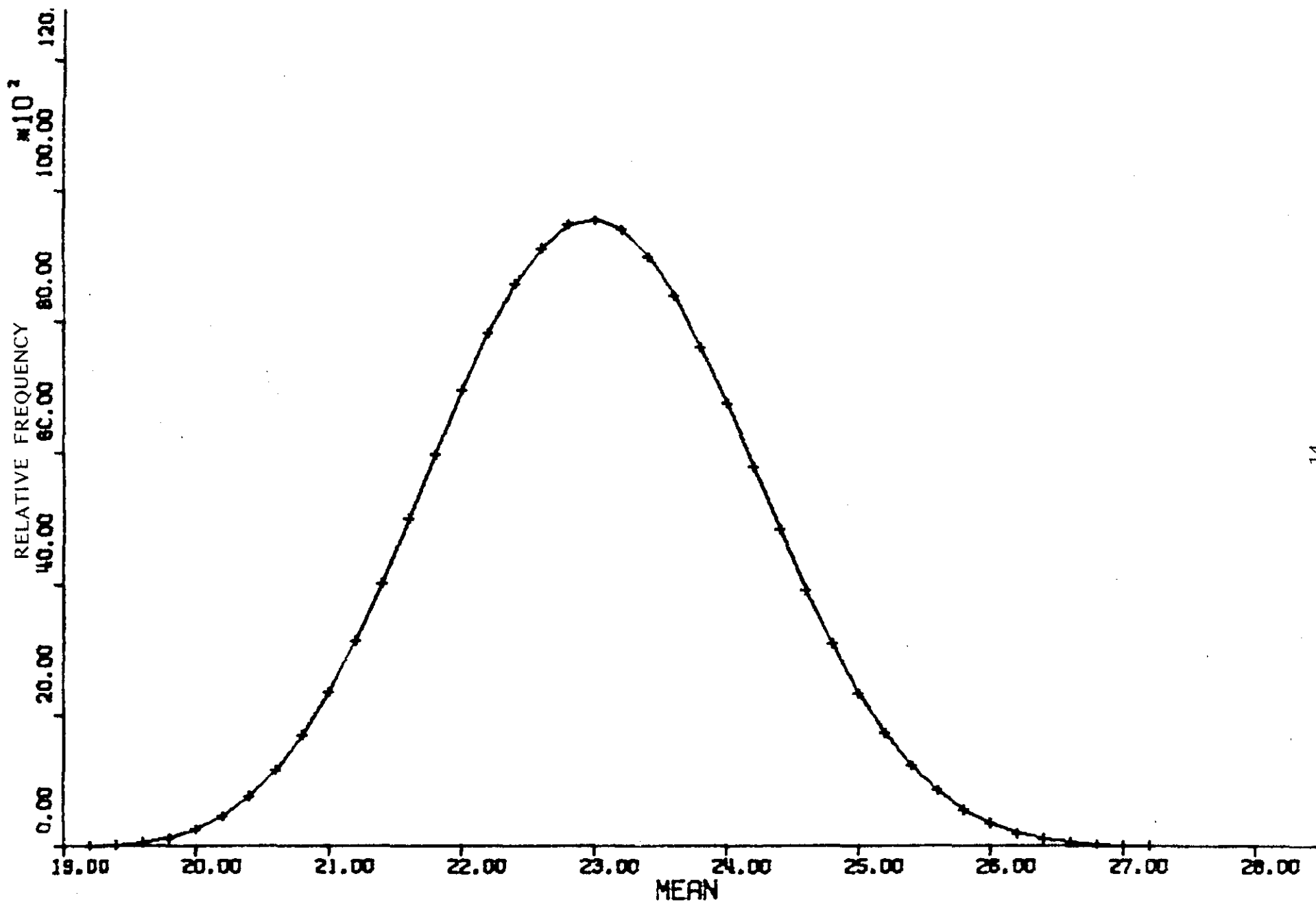


FIGURE 3.2.2: MEAN AGE OF STUDENTS IN A CLASS:  
DISTRIBUTION OF SAMPLE MEANS FOR n=5

### 3.3 The Verification of the Central Limit Theorem - A Computerized Demonstration

In review, the Central Limit Theorem encompasses three suppositions, as discussed in Section 1.2.

1. The distribution of sample means will approach a normal distribution as the sample size increases, regardless of the pattern of population distribution.
2. The mean of the sample means is equal to the population mean.
3. The variance of the sample means is equal to the population variance divided by the sample size  $n$ .

The information generated from the above sampling experiments may be used to verify these suppositions. Figures 3.2.1 and 3.2.2 show that the curve becomes smoother and more bell-shaped as the sample size increases. Even though the normalization can not be established without plotting the relative frequencies from Tables 3.2.1 and 3.2.2 on probability graph paper, the change from an irregular curve to a smooth and bell-shaped curve demonstrates the principle of supposition 1. The use of a sample size  $n=5$  produced a more normal curve than the use of  $n=2$ , and thus if the class used in the example were a large population and sampling were necessary to determine the mean age, Figures 3.2.1 and 3.2.2 show that increasing the sample size will increase the probability that the sample estimates will conform to the characteristics of the population.

Table 3.3.1 demonstrates the verification of suppositions 2 and 3 through a comparison of sample statistics and their counterparts of the population. Population mean (age 23) and variance (7.7) are obtained from Table 2.2.1, the sample means are obtained from the information given in Tables 3.2.1 and 3.2.2. Variances of samples are also obtained from the bottom section of these two tables.

Table 3.3.1 shows that the mean of the sample means obtained from the sampling experiment (23) is exactly equal to its population counterparts; and the variances of samples obtained from the computer output (3.7 and 1.3) are equal to those obtained directly through the statement in supposition 3.

The computer has thus allowed us to demonstrate that the mean and variance of all sample means could be established from the knowledge about the characteristics of the population from which samples are drawn. Through this demonstration, the Central Limit Theorem has been verified in graph and in tabular forms by using computer programs conducting sampling experiments.

TABLE 3.3.1

A COMPARISON OF STATISTICS AND PARAMETERS

Sample Size n	Mean		Variance of Sample Means	
	Sample	Population $\mu$	Obtained From Sampling Experiment	Obtained From Supposition 3 <sup>a/</sup>
2	$\mu_{\bar{y}} = \frac{\sum f \bar{y}}{C_n^N} = \frac{10,005}{435}$ <p style="text-align: center;">= 23</p>	$\mu = 23$	$\sigma_{\bar{y}}^2 = \frac{\sum f (\bar{y} - \mu)^2}{N C_n}$ $= \frac{1624}{435} = 3.7$	$\sigma_{\bar{y}}^2 = \frac{\sigma^2}{n} \left( \frac{N-n}{N-1} \right)$ $= \frac{7.7}{2} \times 0.9655 = 3.7$
5	$\mu_{\bar{y}} = \frac{\sum f \bar{y}}{C_n^N} = \frac{3,277,638}{142,506}$ <p style="text-align: center;">= 23</p>	$\mu = 23$	$\sigma_{\bar{y}}^2 = \frac{\sum f (\bar{y} - \mu)^2}{N C_n}$ $= \frac{190,007}{142,506} = 1.3$	$\sigma_{\bar{y}}^2 = \frac{\sigma^2}{n} \left( \frac{N-n}{N-1} \right)$ $= \frac{7.7}{5} \times 0.8621 = 1.3$

<sup>a/</sup>Since the sampling fraction,  $n/N$ , is larger than 5% of the total number of items in the population, the finite population correction is used in obtaining variances of sample means.

#### 4. CONCLUSION

The verification of the Central Limit Theorem, one of the most important concepts in statistics, has often been found tedious and time-consuming. These problems can be solved by the use of computer facilities in many campuses. The introduction of the use of computer in the classrooms as a teaching aid not only saves time and increases efficiency, it also stimulates interest in learning statistics.

This manual gives a computerized demonstration of the Central Limit Theorem upon a given population of recorded ages of students in an economics class at the University of Nebraska at Omaha. The demonstration consists of five computer programs written in Fortran G and designed to be run on the IBM 360/65 with OS. A detailed description of these procedures is given in Chapter 3. The cost of running these programs is relatively minimal (see Appendix E). It is hoped that students without prior knowledge in computer programming would follow the algorithms, with assistance from the instructor, to run programs and to verify the theorem as a class assignment. Those who have had some prior knowledge in computer programming would use these procedures in more creative ways such as running sampling experiments on a larger population and/or for various larger sample sizes.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

APPENDIXES



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

# APPENDIX A

## PROGRAM FOR LEESAMPL

PROGRAM NAME : LEESAMPL

THIS PROGRAM IS DESIGNED TO DRAW ALL POSSIBLE SAMPLES FROM A UNIVERSE WITHOUT REPLACEMENT. MAXIMUM UNIVERSE SIZE IS 50, MAXIMUM SAMPLE SIZE IS 10. THE PROGRAM ALLOWS FOR RESTARTING OF THE SAMPLING PROCESS. THIS WAS DONE SO THAT JOB MAY BE RUN WITHOUT MAKING SPECIAL ARRANGEMENT WITH THE COMPUTER CENTER AND TO AVOID RUNAWAY JOBS.

```
DIMENSION IUNV(50), ISAMP(10), JRST(10)
DATA JRST/10*0/
DATA I1,I2,I3,I4,I5,I6,I7,I8,I9,I10/10*1/
SUM=0.0
```

READ PARAMETER CARD OF UNIVERSE SIZE AND SAMPLE SIZE, NUMBER OF DECIMALS TO ROUND RESULTS TO, AND A RESTART CODE

```
READ (5,1) IUSIZE, ISSIZ, NDEC, IRST
1 FORMAT (3I2, I1)
WRITE (6,2) IUSIZE, ISSIZ, NDEC
2 FORMAT (1H1, 'SAMPLING PROBLEM', /, 10X, 'UNIVERSE OF ', I2, ' ITEMS', /,
1 10X, 'SAMPLE SIZE ', I2, ' WITHOUT REPLACEMENT', /, 10X,
1 'ROUNDING MEAN VALUES TO ', I2, ' DECIMAL PLACES', //)
WRITE (6,5555) IRST
5555 FORMAT (1X, 'RESTART ', I5)
```

READ TWO CARDS CONTAINING ELEMENTS OF THE UNIVERSE

```
NSAMP=0
JSAMP=0
READ (5,3) (IUNV(I), I=1, 40)
READ (5,3) (IUNV(I), I=41, 50)
3 FORMAT (40I2)
WRITE (6,4)
4 FORMAT (10X, 'ELEMENTS OF UNIVERSE', /)
WRITE (6,5) (IUNV(I), I=1, IUSIZE)
5 FORMAT (10X, 25I3, /, 10X, 25I3)
WRITE (6,6)
6 FORMAT (1H1, 'SAMPLE NUMBER', 9X, 'SAMPLE UNITS', 16X, 'MEAN', /)
```

READ SUBSCRIPTS OF LAST SAMPLE IF A RESTART AND FIX SUBSCRIPTS TO EQUAL THE NEXT SAMPLE

```
IF (IRST .EQ. 0) GO TO 2000
READ (5,4001) JRST, NSAMP, SUM
4001 FORMAT (10I2, I10, E15.4)
4003 CONTINUE
```

BEGIN TO SELECT A SAMPLE

```
2000 WRITE (6,7000) JRST, NSAMP, SUM
7000 FORMAT (1X, 10I3, I10, E15.4)
NSTPI=IUSIZE-(ISSIZ-1)
I11=1
```

THE LOOP BEGINNING WITH DO 801 CONTROLS THE SELECTION OF THE FIRST ELEMENT OF A SAMPLE

```
IF (IRST .NE. 0) I11=JRST(1)
DO 801 I1=I11, NSTPI
ISAMP(1)=IUNV(I1)
```

```

C      IF (ISSIZ .EQ. 1) GO TO 8001
C
C      IF SIZE WAS ONE STOP SELECTING ELEMENTS
C
      II2=II+1
      NSTP2=NSTP1+1
      IF (IRST .NE. 0) II2=JRST(2)
      DO 802 I2=II2,NSTP2
      ISAMP(2)=IUNV(I2)
      IF (ISSIZ .EQ. 2) GO TO 8002
C
C      IF SAMPLE SIZE WAS 2 -- STOP SELECTING ELEMENTS
C
      II3=I2+1
      NSTP3=NSTP2+1
      IF (IRST .NE. 0) II3=JRST(3)
      DO 803 I3=II3,NSTP3
      ISAMP(3)=IUNV(I3)
      IF (ISSIZ .EQ. 3) GO TO 8003
C
C      IF SAMPLE SIZE WAS 3 -- STOP SELECTING ELEMENTS
C
      II4=I3+1
      NSTP4=NSTP3+1
      IF (IRST .NE. 0) II4=JRST(4)
      DO 804 I4=II4,NSTP4
      ISAMP(4)=IUNV(I4)
      IF (ISSIZ .EQ. 4) GO TO 8004
C
C      IF SAMPLE SIZE WAS 4 -- STOP SELECTING ELEMENTS
C
      II5=I4+1
      NSTP5=NSTP4+1
      IF (IRST .NE. 0) II5=JRST(5)
      DO 805 I5=II5,NSTP5
      ISAMP(5)=IUNV(I5)
      IF (ISSIZ .EQ. 5) GO TO 8005
C
C      IF SAMPLE SIZE WAS 5 -- STOP SELECTING ELEMENTS
C
      II6=I5+1
      NSTP6=NSTP5+1
      IF (IRST .NE. 0) II6=JRST(6)
      DO 806 I6=II6,NSTP6
      ISAMP(6)=IUNV(I6)
      IF (ISSIZ .EQ. 6) GO TO 8006
C
C      IF SAMPLE SIZE WAS 6 -- STOP SELECTING ELEMENTS
C
      NSTP7=NSTP6+1
      II7=I6+1
      IF (IRST .NE. 0) II7=JRST(7)
      DO 807 I7=II7,NSTP7
      ISAMP(7)=IUNV(I7)
      IF (ISSIZ .EQ. 7) GO TO 8007
C
C      IF SAMPLE SIZE WAS 7 -- STOP SELECTING ELEMENTS
C
      II8=I7+1
      NSTP8=NSTP7+1
      IF (IRST .NE. 0) II8=JRST(8)
      DO 808 I8=II8,NSTP8

```

```
ISAMP(8)=IUNV(I8)
IF (ISSIZ .EQ. 8) GO TO 8009
```

IF SAMPLE SIZE WAS 8 -- STOP SELECTING ELEMENTS

```
I19=I8+1
NSTP9=NSTP8+1
IF (IPST .NE. 0) I19=JRST(9)
DO 809 I9=I19,NSTP9
ISAMP(9)=IUNV(I9)
IF (ISSIZ .EQ. 9) GO TO 8009
```

IF SAMPLE SIZE WAS 9 -- STOP SELECTING ELEMENTS

```
I110=I9+1
NSTP10=NSTP9+1
IF (IPST .NE. 0) I110=JRST(10)
DO 810 I10=I110,NSTP10
ISAMP(10)=IUNV(I10)
```

THE SAMPLE HAS NOW BEEN DRAWN  
WRITE THE SAMPLE ELEMENTS AND COMPUTE THE SAMPLE MEAN

THE FORMAT SELECTED DEPENDS UPON THE SAMPLE SIZE

THE FORMAT OF THE OUTPUT FILES IS AS FOLLOWS

PRINTED FILE

PRINT ONLY THE NUMBER OF ELEMENTS OF THE SAMPLE SIZE  
AND THE MEAN ON UNIT NUMBER 08.  
THE USE OF THIS STATEMENT: //FT08F001 DD DUMMY  
SUPPRESSES OUTPUT OF THIS FILE

PERMANENT FILE

THE FILE IS OF FIXED RECORD LENGTH (24 BYTES EBCDIC).  
COL 1 THROUGH 4 SAMPLE MEAN ROUNDED TO NDEC DECIMAL PLACES  
THE NEXT 20 BYTES CONTAIN THE SAMPLE ELEMENTS, 2 BYTES PER  
ELEMENT WITH UNUSED SPACE FILLED WITH BLANKS SO THAT  
THE RECORD LENGTH IS THE SAME REGARDLESS OF SAMPLE SIZE.

```
CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
```

```
SUM=SUM+SMEAN
```

```
WRITE (12,3010) IJ,(ISAMP(I),I=1,10)
```

```
3010 FORMAT (14,10I2)
```

```
2010 WRITE (8,2111) NSAMP,(ISAMP(I),I=1,10),SMEAN
```

```
2111 FORMAT (3X,110,6X,10I3,3X,F6.2)
```

```
IRST=0
```

```
JSAMP=JSAMP+1
```

```
IF (JSAMP .GT. 100000) GO TO 100
```

```
810 CONTINUE
```

```
8009 CONTINUE
```

```
IF (ISSIZ .NE. 9) GO TO 809
```

```
CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
```

```
SUM=SUM+SMEAN
```

```
WRITE (12,3009) IJ,(ISAMP(I),I=1,9)
```

```
3009 FORMAT (14,9I2,2X)
```

```
2009 WRITE (8,2109) NSAMP,(ISAMP(I),I=1,9),SMEAN
```

```
2109 FORMAT (3X,110,6X,9I3,6X,F6.2)
```

```
IRST=0
```

```
JSAMP=JSAMP+1
```

```

      IF (JSAMP .GT. 100000) GO TO 100
809  CONTINUE
8003 CONTINUE
      IF (ISSIZ .NE. 8) GO TO 308
      CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
      SUM=SUM+SMEAN
      WRITE (12,3008) IJ,(ISAMP(I),I=1,8)
3008 FORMAT (14,8I2,4X)
2008 WRITE (8,2108) NSAMP,(ISAMP(I),I=1,8),SMEAN
2108 FORMAT (3X,I10,6X,8I3,9X,F6.2)
      IRST=0
      JSAMP=JSAMP+1
      IF (JSAMP .GT. 100000) GO TO 100
809  CONTINUE
8007 CONTINUE
      IF (ISSIZ .NE. 7) GO TO 307
      CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
      SUM=SUM+SMEAN
      WRITE (12,3007) IJ,(ISAMP(I),I=1,7)
3007 FORMAT (14,7I2,6X)
2007 WRITE (8,2107) NSAMP,(ISAMP(I),I=1,7),SMEAN
2107 FORMAT (3X,I10,6X,7I3,12X,F6.2)
      IRST=0
      JSAMP=JSAMP+1
      IF (JSAMP .GT. 100000) GO TO 100
307  CONTINUE
8006 CONTINUE
      IF (ISSIZ .NE. 6) GO TO 306
      CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
      SUM=SUM+SMEAN
      WRITE (12,3006) IJ,(ISAMP(I),I=1,6)
3006 FORMAT (14,6I2,8X)
2006 WRITE (8,2106) NSAMP,(ISAMP(I),I=1,6),SMEAN
2106 FORMAT (3X,I10,6X,6I3,15X,F6.2)
      IRST=0
      JSAMP=JSAMP+1
      IF (JSAMP .GT. 100000) GO TO 100
806  CONTINUE
8005 CONTINUE
      IF (ISSIZ .NE. 5) GO TO 305
      CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
      SUM=SUM+SMEAN
      WRITE (12,3005) IJ,(ISAMP(I),I=1,5)
3005 FORMAT (14,5I2,10X)
2005 WRITE (8,2105) NSAMP,(ISAMP(I),I=1,5),SMEAN
2105 FORMAT (3X,I10,6X,5I3,18X,F6.2)
      IRST=0
      JSAMP=JSAMP+1
      IF (JSAMP .GT. 100000) GO TO 100
905  CONTINUE
8004 CONTINUE
      IF (ISSIZ .NE. 4) GO TO 304
      CALL MEAN (ISSIZ,ISAMP,NDEC,NSAMP,IJ,SMEAN)
      SUM=SUM+SMEAN
      WRITE (12,3004) IJ,(ISAMP(I),I=1,4)
3004 FORMAT (14,4I2,12X)
2004 WRITE (8,2104) NSAMP,(ISAMP(I),I=1,4),SMEAN
2104 FORMAT (3X,I10,6X,4I3,21X,F6.2)
      IRST=0
      JSAMP=JSAMP+1
      IF (JSAMP .GT. 100000) GO TO 100
804  CONTINUE

```

```

8003 CONTINUE
IF (ISSIZ .NE. 3) GO TO 803
CALL MEAN (ISSIZ, ISAMP, NDEC, NSAMP, IJ, SMEAN)
SUM=SUM+SMEAN
WRITE (12, 3003) IJ, (ISAMP(I), I=1, 3)
3003 FORMAT (14, 3I2, 14X)
2003 WRITE (8, 2103) NSAMP, (ISAMP(I), I=1, 3), SMEAN
2103 FORMAT (3X, I10, 6X, 3I3, 24X, F6.2)
IRST=0
JSAMP=JSAMP+1
IF (JSAMP .GT. 100000) GO TO 100
803 CONTINUE
8002 CONTINUE
IF (ISSIZ .NE. 2) GO TO 302
CALL MEAN (ISSIZ, ISAMP, NDEC, NSAMP, IJ, SMEAN)
SUM=SUM+SMEAN
WRITE (12, 3002) IJ, (ISAMP(I), I=1, 2)
3002 FORMAT (14, 2I2, 16X)
2002 WRITE (8, 2202) NSAMP, (ISAMP(I), I=1, 2), SMEAN
2202 FORMAT (3X, I10, 6X, 2I3, 27X, F6.2)
IRST=0
JSAMP=JSAMP+1
IF (JSAMP .GT. 100000) GO TO 100
302 CONTINUE
8001 CONTINUE
IF (ISSIZ .NE. 1) GO TO 301
CALL MEAN (ISSIZ, ISAMP, NDEC, NSAMP, IJ, SMEAN)
SUM=SUM+SMEAN
WRITE (12, 3001) IJ, ISAMP(1)
3001 FORMAT (14, I2, 14X)
2001 WRITE (8, 2201) NSAMP, ISAMP(1), SMEAN
2201 FORMAT (3X, I10, 3X, I3, 30X, F6.2)
IRST=0
JSAMP=JSAMP+1
IF (JSAMP .GT. 100000) GO TO 100
801 CONTINUE
WRITE (6, 8888) NSAMP, SUM
8888 FORMAT (1X, I10, F15.4)
WRITE (6, 101)
101 FORMAT (1X, 'END OF JOB')
GO TO 104
100 CONTINUE
WRITE (7, 102) I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, NSAMP, SUM
102 FORMAT (10I2, I10, F15.4)
WRITE (6, 103) I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, NSAMP
103 FORMAT (1X, 'JOB STOPPED -- SUBSCRIPTS OF LAST SAMPLE ', 10I3, ' NUMB
IER OF SAMPLES ', I10)
104 STOP
END
SUBROUTINE MEAN (ISSIZ, ISAMP, NDEC, NSAMP, IJ, SMEAN)
DIMENSION ISAMP(10)
C
C SAMPLE OF SIZE ISSIZ IS NOW SELECTED -- COMPUTE MEAN AND ROUND TO NDEC
C DECIMAL PLACES
SMEAN=0.0
DO 1001 I=1, ISSIZ
1001 SMEAN=SMEAN+ISAMP(I)
SMEAN=SMEAN/ISSIZ
IF (NDEC) 1004, 1004, 1003
1003 SMEAN=SMEAN*(10.0**NDEC)
1004 SMEAN=SMEAN+.5
IJ=SMEAN
SMEAN=IJ
IF (NDEC) 1002, 1002, 1003
1003 SMEAN=SMEAN/(10.0**NDEC)
1002 NSAMP=NSAMP+1
RETURN
END

```



## APPENDIX B

### PREPARATION OF INPUT DATA FOR LEESAMPL

(Unless otherwise specified all data are to be punched as integer numbers and right adjusted in their designated fields.)

1. Parameter Card: This card specifies the parameters to be used in the process as described below:

<u>Card Columns</u>	<u>Parameter Specification</u>
1 - 2	Size of the universe, N.
2 - 4	Size of the sample, n.
6	Number of decimals in mean
7	Is this a restart? Answer "0" if it is "No," and Answer "1" if it is "Yes."

2. Population Cards: This card specifies the observations of the population. Punch the observations of the population on two cards. Beginning from column 1 card 1, each element occupies two columns. Hence, card 1 contains 40 elements and the first 20 columns of card 2 contains the remaining 10 elements. A total of two data cards is required even though the total number of elements in the universe may be smaller than 50.
3. Restart Data: This card is to be in input deck only if column 7 of the parameter card above has been answered yes(1). It contains the data necessary for a successful restart of the sampling process. It is prepared from a card punched by the computer at the end of the immediately preceding sampling run. It contains the following data:
  - a. subscripts of the elements of the universe for the next sample to draw upon restart
  - b. total number of samples drawn in all prior sampling runs
  - c. sum of the means of all samples drawn in prior sampling runs (punched in E format)

Items b and c above are automatically determined by the prior run; however, item a must be determined in the following manner: the immediately preceding run will give the investigator the subscripts for the last sample drawn,



the total number of samples drawn to date and the sum of the means of prior samples drawn in punched card and printed form. The format of the punched card matches the input format required for this data. Use the following procedure to determine subscripts for next sample.

4. Determination of Sample Subscripts: The formula for the maximum possible subscript for the first element of a sample size  $n$  taken for a universe of size  $N$  is:  $s_1=N-(n-1)$ . The formula for the maximum possible subscript of the second element is  $s_2=s_1+1$ . The general formula for the  $n$ th maximum possible subscript is:  $s_n=s_{n-1}+1$ .

The subscripts to be entered as the subscripts of the next sample are calculated from the subscripts of the last sample such that none of the subscripts entered exceed their maximum allowable values. They are calculated from right to left, computing the subscript for the  $n$ th element first. This number is normally 1 greater than the subscripts of the last sample. However, several things should be considered:

- If the value thus calculated exceeds the maximum allowable value, the maximum allowable is substituted and 1 is added to the  $(n-1)$ th subscript.
- If this value does not exceed the maximum allowable value, the subscripts for the next sample are determined as shown: subscripts 1 through  $n-2$  equal the values given them for the last sample drawn, subscript  $n-1$  equals the value just computed, and subscript  $n$  equals the maximum allowable value.
- If the value exceeds the maximum allowable value, the maximum allowable value is substituted for subscript  $N-1$  and the 1 is added to subscript  $n-2$  and the process described in b and c is repeated for subscript  $n-2$ . This process is repeated until it yields a valid set of sample subscripts.

An example of the above described process follows:

Assume:  $n=5$ ,  $N=30$ . When the sampling stopped, the subscript values for the last sample were: 6, 26, 28, 29, 30.

The maximum allowable values for the subscripts are: 26, 27, 28, 29, 30. Using the process described above, the set of subscripts to enter for restart are: 6, 27, 28, 29, 30. The input is to be punched as follows:

<u>Item</u>	<u>Card Columns</u>
Restart subscript 1	1-2
" " 2	3-4
" " 3	5-6
" " 4	7-8
" " 5	9-10
" " 6	11-12
" " 7	13-14
" " 8	15-16
" " 9	17-18
" " 10	19-20
Total number of samples drawn in prior run	21-30
Sum of samples drawn in prior run	31-45

Output is as follows:

1. A permanent file containing the actual samples drawn during the run as well as the sample means.
  2. A printed copy of the samples (optional).
  3. Information necessary for the restart of the sampling process or a message indicating that the process has been completed.
5. Running Scheme: The following running scheme was used for an actual run using  $N=30$ ,  $n=5$ . This required two runs of LEESAMPL, with the output stored as file 1 and file 2, respectively, on a standard labeled tape. LEESORT was run twice, and two punched card decks were produced. LEEDATA was run once, using the two decks produced above as input. LEEGRAF1 was run to produce graph
6. Job Control Language (JCL) required to run each program: The required JCL to run each of the programs is shown in Appendix D. These changes may need to be made in the JCL shown in Appendix D:
- a. The proper account numbers for the job card must be provided.
  - b. Appropriate vol ser numbers of tapes (tape numbers), must be used for each run.
  - c. Appropriate label numbers (file numbers) and PSN's (file names) must be used.
  - d. The non-standard control cards (cards with asterisk (\*) instead of slash (/) in the first two columns) are designed to use at the University of Nebraska Computer Network only. Students at other institutions are advised to consult their local computer facilities for any modification that is locally needed for executing the programs.

THE UNIVERSITY OF CHICAGO PRESS

APPENDIX C

PROGRAMS FOR (1) LEEDATA, (2) LEEPLLOT AND (3) LEEGRAF1

(1) LEEDATA

PROGRAM NAME: LEEDATA

THIS PROGRAM PRODUCES A SUMMARY OF THE OUTPUT FILE PRODUCED BY LEESURT

THE FORMAT OF THE OUTPUT FILE IS AS FOLLOWS:

COL 1 THROUGH 8 VALUE

COL 9 THROUGH 16 FREQUENCY COUNT OF THAT VALUE

INPUT IS READ ON UNIT NUMBER 10

OUTPUT IS PRODUCED ON UNIT NUMBER 11

```
NPTS=0
READ (10,1,END=100) IVAL
1  FORMAT (I4)
   JVAL=IVAL
   KVAL=1
31  READ (10,1,END=100) IVAL
   IF (IVAL-JVAL) 20,10,20
20  WRITE (11,2) JVAL,KVAL
   WRITE (6,6666) JVAL,KVAL
   ?  FORMAT (2I8)
   JVAL=IVAL
   NPTS=NPTS+1
   KVAL=1
   GO TO 30
10  KVAL=KVAL+1
30  GO TO 31
100 WRITE (11,2) JVAL,KVAL
   WRITE (6,6666) JVAL,KVAL
6666 FORMAT (1X,2I10)
   NPTS=NPTS+1
   WRITE (6,101) NPTS
101  FORMAT (1X,I8,' POINTS FOR SUMMARY')
   STOP
   END
```

(2) LEEPLOT

```
C
C
C
C
C
C
C
PROGRAM NAME: LEEPLOT
THIS PROGRAM PREPARES A FILE OF PLOT DATA FROM THE COMBINED
OUTPUT OF ALL LEE DATA RUNS. INPUT DATA FROM LEE DATA IS READ ON
UNIT NUMBER 10 AND WRITTEN ON UNIT NUMBER 11
C
C
NPTS=0
READ (10,1) IVALA,IVALB
1 FORMAT (2I8)
JVALA=IVALA
JVALB=IVALB
31 READ (10,1,END=100) IVALA,IVALB
IF (IVALA-JVALA) 20,10,20
20 WRITE (11,1) JVALA,JVALB
WRITE (6,6666) JVALA,JVALB
6666 FORMAT (1X,2I8)
JVALA=IVALA
JVALB=IVALB
NPTS=NPTS+1
GO TO 31
10 JVALB=JVALB+IVALB
GO TO 31
100 WRITE (11,1) JVALA,JVALB
WRITE (6,6666) JVALA,JVALB
NPTS=NPTS+1
WRITE (6,101) NPTS
101 FORMAT (1X,I8,' POINTS TO PLCT')
STOP
END
```

### (3) LEEGRAF1

PROGRAM NAME: LEEGRAF1

THIS PROGRAM CREATES A PLOT FILE OF THE DATA PRODUCED BY LEEPLOT  
THE PLOT FILE IS BUILT ON UNIT NUMBER 04. THIS FILE IS ACCESSED AT  
JND FOR PLOTTING BY THE COMPLIT PTC-4 PLOTTER.  
THE PROGRAM MAY EASILY BE CHANGED FOR USE WITH OTHER PLOTTERS.  
INPUT DATA IS READ ON UNIT NUMBER 10.

```
CALL A3YLN2('PAUL S. T. LEE', '506606299', 0, 14)
DIMENSION XVAL(435), YVAL(435), XLABEL(1), YLABEL(3)
DATA XLABEL/'MEAN'/
DATA YLABEL/'FR', 'EQJE', 'NCY' /
I=1
XVAL(1)=0.0
YVAL(1)=0.0
```

```
DATA TO BE PLOTTED IS READ INTO ARRAY POSITIONS 2 THRU N+1
IF THERE ARE N PCINTS TO PLOT
THIS IS DONE SO THAT THE SUBROUTINE SCALE WILL SCALE DATA
FOR A GOOD SHAPED CURVE AND INCLUDE THE ORIGIN IN THE GRAPH.
```

```
I=2
SUM=0.0
TSTD=0.0
YNUM=0.0
2 READ (10,1,END=100) XVAL(I),YVAL(I)
1 FORMAT (F8.1,F8.0)
I=I+1
GO TO 2
100 I=I-1
```

PRINT DISTRIBUTION

```
WRITE (6,200)
200 FORMAT (1H1, 'DISTRIBUTION OF SAMPLE MEANS', /)
WRITE (6,201)
201 FORMAT (1X, 'MEAN', 5X, 'FREQUENCY')
DO 202 J=2, I
SUM=SUM+XVAL(J)*YVAL(J)
YNUM=YNUM+YVAL(J)
202 WRITE (6,203) XVAL(J), YVAL(J)
203 FORMAT (1X, F5.1, 5X, F8.0)
SUM=SUM/YNUM
DO 204 J=2, I
STD=((SUM-XVAL(J))*(SUM-XVAL(J)))*YVAL(J)
204 TSTD=TSTD+STD
WRITE (6,666) TSTD
WRITE (6,667) YNUM
666 FORMAT (1X, 'SUM OF DIFFERENCES SQUARED ', E15.10)
667 FORMAT (1X, 'TOTAL NUMBER OF SAMPLES ', E15.10)
TSTD=TSTD/(YNUM-1.)
WRITE (6,668) TSTD
668 FORMAT (1X, 'S SQUARED ', E15.10)
TSTD=SQRT(TSTD)
WRITE (6,669) TSTD
669 FORMAT (1X, 'STD DEV ', E15.10)
```

NPTS=NUMBER OF POINTS TO PLOT

```
NPTS=I
CALL PLOT (1.5, 0.5, -3)
CALL SCALE (YVAL, 9.50, NPTS, 1, 10.0)
```

(3) LEEGRAFI - Continued

```
CALL AXIS(0.0,0.0,YLABEL,12,9.50,90.0,YVAL(NPTS+1),YVAL(NPTS+2),
1 10.0)
CALL SCALE (XVAL,0.5,NPTS,1,10)
CALL AXIS (0.0,0.0,XLABEL,-4,9.50,0.0,XVAL(NPTS+1),XVAL(NPTS+2),
1 10.0)
X=0.
CALL PLOT (X,0.0,-3)
LAST=NPTS+2
DO 700 I=2, LAST
XVAL(I-1)=XVAL(I)
700 YVAL(I-1)=YVAL(I)
NPTS=NPTS-1
CALL LINE (XVAL,YVAL,NPTS,1,1,3)
CALL PLOT (0.0,0.0,999)
STOP
END
```

## APPENDIX D

### JOB CONTROL LANGUAGE FOR (1) LEESAMPL, (2) LEESORT, (3) LEEDATA, (4) LEEPLT AND (5) LEEGRAF1

#### (1) JCL FOR LEESAMPL

```
//LEESAMPL JOB ZYSXDA.....00,'DRAW SAMPLES',CLASS=F,PRTY=0
/*ROUTE EXEC UNL
/*JOBPARM B=N041,TAPES=1,TIME=10
/*JOBPARM HOLD=PURGE
//MOUNT EXEC CONSOL
//READF DD *
        MOUNT TAPE T01103 RING IN
//RUN EXEC PGM=LEESAMPL,TIME=10
//STEPLIB DD DSN=A.ZYSX.DA160800.PAUL.LEE,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT07F001 DD SYSOUT=B
//FT08F001 DD DUMMY
//FT12F001 DD UNIT=TAPE,VOL=SER=T01103,LABEL=(5,SL),
// DCB=(RECFM=FB,LRECL=24,BLKSIZE=4800),
// DSN=U30S05B
//FT05F001 DD *      INSERT DATA CARDS FOLLOWING THIS CARD
```

#### (2) JCL FOR LEESORT

```
//LEESORT JOB ZYSXDA.....00,'SORT DATA',CLASS=C,PRTY=6
/*ROUTE EXEC UNL
/*JOBPARM BIN=N041,TAPES=2
/*JOBPARM HOLD=PURGE
//MOUNT EXEC CONSOL
//READF DD *
        MOUNT T01103 NO RING
        LT0160 RING IN
//RUN EXEC SORT,TIME=2
//SORTIN DD UNIT=TAPE,VOL=SER=T01103,LABEL=(5,SL),DISP=OLD,
// DSN=U30S05B
//SORTOUT DD UNIT=TAPE,VOL=SER=LT0160,LABEL=(1,SL),DISP=(NEW,KEEP),
// DCB=(RECFM=FB,LRECL=24,BLKSIZE=4800),DSN=SORTED
//SYSIN DD *
        SORT FIELDS=(1,4,CH,A)
```



### (3) JCL FOR LEEDATA

```
//LEEDATA JOB ZYSXDA.....00,'SUMMARIZE SAMPLE',CLASS=C,PRTY=6
/*ROUTE EXEC UNL
/*JOBPARM B=N041,TAPES=1
/*JOBPARM HOLD=PURGE
//MOUNT EXEC CONSOL
//READF DD *
      MOUNT LT0160 NO RING
//RUN EXEC PGM=LEEDATA
//STEPLIB DD DSN=A.ZYSX.DA160800.PAUL.LEE,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT11F001 DD SYSOUT=B
//FT10F001 DD UNIT=TAPE,VOL=SER=LT0160,DISP=OLD,LABEL=(1,SL,,IN)
// DSN=SORTED
```

### (4) JCL FOR LEEPLOT

```
//LEEPLUT JOB ZYSXDA.....00,'PLOT DATA',CLASS=C,PRTY=6
/*ROUTE EXEC UNL
/*JOBPARM TAPES=0,BIN=N041
/*JOBPARM HOLD=PURGE
//SORT EXEC SORT
//SORTIN DD DDNAME=INPUT
//SORTOUT DD UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
// SPACE=(8000,(5,2)),DSN=%%DATA,DISP=(NEW,PASS)
//SYSIN DD *
      SORT FIELDS=(1,3,CH,A)
//INPUT DD *      INPUT DATA FOLLOWS THIS CARD
//RUN EXEC PGM=LEEPLUT      INPUT DATA PRECEDES THIS CARD
//STEPLIB DD DSN=A.ZYSX.DA160800.PAUL.LEE,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT11F001 DD SYSOUT=B
//FT10F001 DD DSN=%%DATA,DISP=(OLD,PASS)
```

### (5) JCL FOR LEEGRAF1

```
//LEEGRAF1 JOB ZYSXDA.....00,'PLOT',CLASS=A
/*ROUTE EXEC UNL
/*JOBPARM B=N041
//RUN EXEC PGM=LFEGRAF1
//STEPLIB DD DSN=A.ZYSX.DA160800.PAUL.LEE,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD UNIT=SYSDA,VOL=SER=TEMP,DISP=(NEW,KEEP),
// DCB=(RECFM=FB,LRECL=144,BLKSIZE=1440),
// SPACE=(TRK,(2,2),RLSE),
// DSN=PLOT1
//FT04F001 DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=%%PLT_LOG,
// SPACE=(TRK,(1,1)),DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//FT05F001 DD *
```

## APPENDIX E

### ESTIMATE OF COST OF RUNNING SAMPLING PROGRAMS

The following cost estimates are based upon using priority 0 and rates in effect June, 1978.

#### LEESAMPL

Approximate cost: \$8.00 for 100,000 samples.

Cost is proportional to number of samples, i.e., 150,000 samples = \$12.00

50,000 samples = \$ 4.00

#### LEESORT

Approximate cost: \$2.00 for 100,000 samples.

Cost is also proportional to number of samples.

#### LEEDATA

Approximate cost: \$1.50 for 100,000 samples.

Cost is proportional to number of samples.

#### LEEPLLOT

Approximate cost: \$1.00 per sampling scheme.

#### LEEGRAF1

Approximate cost: \$1.00 per plat produced.

