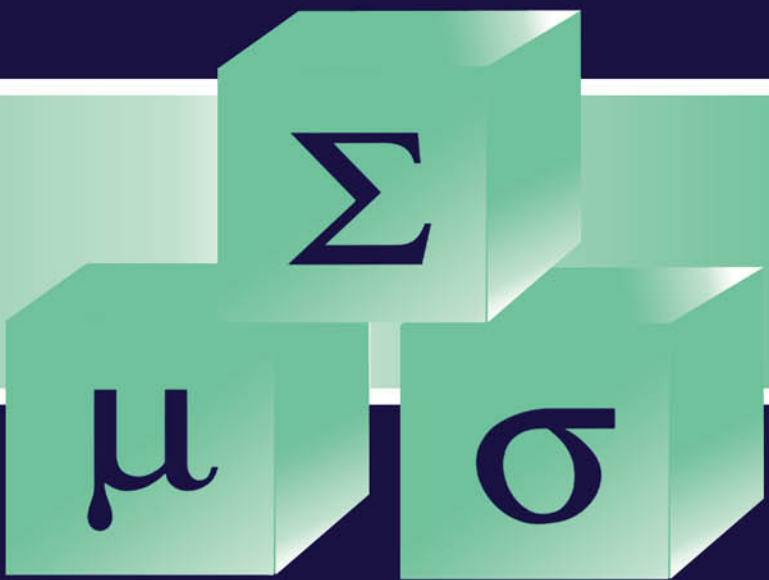
FREE CD ENCLOSED!
Book not returnable if software has been removed.

UNDERSTANDING STATISTICAL CONCEPTS USING S-PLUS

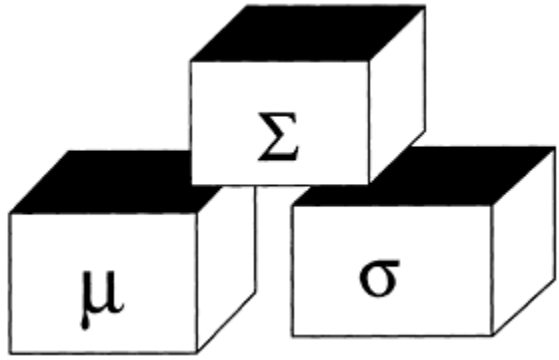


Randall E. Schumacker • Allen Akers

UNDERSTANDING STATISTICAL CONCEPTS USING S-PLUS

UNDERSTANDING STATISTICAL CONCEPTS USING S-PLUS

Randall E.Schumacker
Allen Akers
University of North Texas



LEA
2001

LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
Mahwah, New Jersey London

Lawrence Erlbaum Associates, Inc., Publishers
10 Industrial Avenue
Mahwah, NJ 07430

This edition published in the Taylor & Francis e-Library, 2009.

Cover design by Kathryn Houghtaling Lacey

Library of Congress Cataloging-in-Publication Data
Schumacker, Randall E.
Understanding statistical concepts using S-plus/Randall E.Schumacker and Allen Akers.
p. cm.
Includes bibliographical references and indexes.
ISBN 0-8058-3623-3 (pbk.: alk. paper)
1. Mathematical statistics—Data processing. 2. S-Plus. I. Akers, Allen. II. Title.
QA276.4.8366 2001
519.5′0285–dc21 00–065444
CIP

ISBN 1-4106-0087-4 Master e-book ISBN

ISBN 0-8058-3623-3 (Print Edition)

DEDICATED TO OUR CHILDREN

Rachel and Jamie

Joshua

TABLE OF CONTENTS

PREFACE x

ACKNOWLEDGMENTS xiv

PART I: INTRODUCTION AND STATISTICAL THEORY

1. Statistical Theory 3
2. Generating Random Numbers 9
3. Frequency Distributions 15
4. Stem and Leaf Plots 24
5. Population Distributions 32
6. Measures of Central Tendency 38
7. Measures of Dispersion 44
8. Sample Size Effects 49
9. Tchebysheff Inequality Theorem 55
10. Normal Bell-Shaped Curve 63

PART II: PROBABILITY AND PROBABILITY DISTRIBUTIONS

11. Probability 70
12. Joint Probability 79
13. Addition Law of Probability 84
14. Multiplication Law of Probability 89
15. Conditional Probability 93
16. Combinations and Permutations 99

PART III: MONTE CARLO AND STATISTICAL DISTRIBUTIONS

17.	Binomial Distribution	108
18.	Monte Carlo Simulation	116
19.	Normal Distribution	124
20.	t Distribution	130
21.	Chi-square distribution	139
22.	F Distribution	146

PART IV: SAMPLING AND INFERENCE

23.	Sampling Distributions	156
24.	Central Limit Theorem	162
25.	Confidence Intervals	173
26.	Hypothesis Testing	179
27.	Type I Error	190
28.	Type II Error	197

PART V: HYPOTHESIS TESTING IN RESEARCH

29.	z test statistic for proportions	205
30.	Chi-square test statistic	215
31.	t Test for Mean Differences	222
32.	Analysis of Variance	231
33.	Correlation	242
34.	Linear Regression	252

PART VI: REPLICABILITY OF FINDINGS

35.	Cross Validation	265
36.	Jackknife	272
37.	Bootstrap	279
38.	Meta-Analysis	285
39.	Significance Testing vs. Practical Importance	296
	GLOSSARY OF TERMS	302
	ABOUT THE AUTHORS	310
	APPENDIX	311
	ANSWERS TO CHAPTER EXERCISES	322
	AUTHOR INDEX	330
	SUBJECT INDEX	331

PREFACE

This book was written as a supplemental text for use with introductory or intermediate statistics books. The content of each chapter is appropriate for any undergraduate or graduate level statistics course. The chapters are ordered along the lines of many popular statistics books so it should be easy to supplement the chapter content and exercises with your statistics book and lecture materials. Each chapter lists a set of objectives and a summary of what the student should have learned from the chapter. The content of each chapter was written to enrich a students' understanding of statistical concepts as well as S-PLUS due to the inclusion of exercises that use S-PLUS script programs. The chapter exercises reinforce an understanding of the statistical concepts and the S-PLUS script programs in the chapters.

Computational skills are kept to a minimum in the book by including S-PLUS script programs that can be run for the exercises in the chapters. Students are not required to master the writing of S-PLUS script programs, but explanations of how the programs work and program output are included in each chapter. S-PLUS is an advanced statistical package with an extensive library of functions that offers flexibility in writing customized statistical routines. The S-PLUS script commands and functions provide the capability of programming object and dialog windows that are commonly used in Windows software applications. The S-PLUS software program also contains pull-down menus for the statistical analysis of data.

ORGANIZATION OF THE TEXT

The early chapters in the textbook offer a rich insight into how probability has shaped statistics in the behavioral sciences. In later chapters, the persons who created various statistics and hypothesis testing are highlighted. A final chapter brings together current thinking on significance testing versus practical importance of research findings.

The chapters are grouped into six parts. Part I includes chapters on statistical theory, random sampling, and basic descriptive statistics (frequency distributions, stem and leaf plots, central tendency, and dispersion). Basic concepts related to calculating and interpreting these measures are covered in the chapter exercises. Unique aspects of Part I are the generating of random numbers, presentation of population distributions, sample size effects, Tchebycheff Inequality Theorem, and determining the probability area under a population distribution. These chapters provide an understanding of how the theoretical normal distribution is developed and used in statistics.

Part II includes chapters that cover the basic ideas behind probability. The content covers probability based on independent outcomes, joint outcomes, combinations, and permutations. The importance of the addition and multiplication laws of probability in statistics is further explained. The importance of probability in understanding basic statistical concepts is further enhanced in the chapter exercises, especially the difference between exact probabilities and theoretical probabilities. The S-PLUS script programs are helpful in examining the various outcomes discussed in the chapters.

Part III includes chapters that extend an understanding of how probability and frequency distributions result in sampling distributions. The concept behind how sampling distributions are used in probability statistics is presented. The binomial distribution, normal distribution, t-distribution, chi-square distribution, and F-distribution are presented in the chapter exercises. Unique to this part is the discussion of how Monte Carlo methods can be used to create the sampling distribution of a statistic, e.g., z, t, chi-square, and F statistics.

Part IV includes the importance of understanding and interpreting results based on the sampling distribution of a statistic, which will be helpful in understanding chapter exercises in Part V. An important chapter on the Central Limit Theorem provides an understanding of how random samples from any population distribution will result in a normal sampling distribution of a statistic especially as sample size increases. This forms the basis for later discussion on confidence intervals, hypothesis testing, Type I error and Type II error. A key concept emphasized is the inference made from the sample statistic to the population parameter and the consequence of making that decision.

Part V contains the essential chapters on hypothesis testing in research. The basic statistics covered are the z-test, chi-square, t-test, analysis of variance, correlation, and regression. Each chapter begins with an elaboration of the origin of the statistical procedure and the person(s) who created them. Examples demonstrate the various types of research questions, how the statistic is used, and interpretation of results. The chapter content and exercises provide the necessary skills for students to better understand how these statistics are used to answer research questions. The logic and steps taken to conduct statistical hypothesis testing using the scientific method is emphasized.

Part VI introduces the importance of replicating research results. In the absence of being able to reproduce the findings in a research study (replicating the study with another random sample of data), cross-validation, jackknife, and bootstrap methods are used to estimate the replicability of findings. Cross-validation involves the random splitting of a sample, applying the statistical test on both sub-samples, and comparing results. Jackknife involves estimating the sample statistic several times based on dropping a different data value each time. The bootstrap procedure uses resampling (sampling with replacement) to estimate the amount of bias in a sample statistic as an estimate of a population parameter. A chapter also presents meta-analysis techniques, which quantitatively combine and synthesize the results of several related research studies. This provides an understanding of the role converting various statistics to a common scale for comparative purposes plays in meta-analysis. A final chapter compares statistical significance testing and the practical importance of research findings. Although hypothesis testing is stressed in Part V, the last chapter in Part VI provides an understanding that significance testing, i.e., $p<.05$, is not necessarily sufficient evidence of the practical importance of research results. It highlights the importance of reporting the sample statistic, significance level, confidence interval, and effect size. Reporting of these values extends the students' thinking beyond significance testing.

S-PLUS PROGRAMS

Each chapter contains at least one S-PLUS program that produces computer output for the chapter exercises. A CD is available with the book and contains S-PLUS script programs that enhance the basic understanding and concepts in the chapters. The S-PLUS programs on the CD have an extension ".ssc", which refers to a script file in S-PLUS. A chapter number, e.g., chap01.ssc, precedes each script program. After mastering the concepts in the book, the S-PLUS software can be used for data analysis and graphics using pull-down menus. The statistical tests and methods for the replicability of findings in Parts V and VI are included in the S-PLUS statistical software.

When using the supplemental exercises for classroom instruction, S-PLUS will need to be installed on either a stand-alone IBM compatible personal computer (with a roll cart and a projection monitor) or on a local area network for use with a computer-equipped classroom. Information concerning the purchase and installation of S-PLUS on a local area network should be referred to your computing center personnel. Student versions of S-PLUS may also be available for purchase at university bookstores.

There are several Internet web sites that offer information, resources, and assistance with S-PLUS. These can be located by entering "S-PLUS" in the search engines accessible from any Internet browser software. The main Internet URL (Uniform Resource Locator) address for S-PLUS is: http://www.mathsoft.com/. The MathSoft Company markets the S-PLUS software. S-PLUS is a high level programming language for statistics and graphics that was developed at Bell Labs. A second URL is, http://lib.stat.cmu.edu/S/, which accesses Carnegie Mellon University's software library of functions and routines written by various authors. The S-PLUS programs were not written to run in R (an open-source version of S/S-PLUS), although R programs could be written to run most of the chapter exercises (with the exception of dialog boxes). More information about R programs and comparisons to S/S-PLUS is available at: http://lib.stat.cmu.edu/R/CRAN.

Randall E.Schumacher
Allen Akers

ACKNOWLEDGEMENTS

The preparation of this text was greatly facilitated by reviews from three anonymous reviewers, several graduate students at the University of North Texas who enrolled in a graduate level course on Statistical Theory and Simulation, and two colleagues in the field of statistics. They all provided much needed feedback on the importance and accuracy of the material in the chapters. The later chapters on the replication of research findings, especially the chapter on significance testing versus practical importance, were enriched from discussions with the students.

We would like to thank the editorial staff at Lawrence Erlbaum Associates, Inc. Publishers for their assistance in getting this book into print. We are especially grateful to Misty Berend for typing the statistical tables. Finally, we are grateful for permission to reproduce the selected statistical tables from Sir Ronald A.Fisher, F.R.S. and Frank Yates, *Statistical Tables for Biological, Agricultural, and Medical Research,* Oliver & Boyd, Ltd., Edinburgh, which were abridged in the John T.Roscoe, *Fundamental Research Statistics for the Behavioral Sciences* (2nd Ed.) book, 1975.

I

INTRODUCTION AND STATISTICAL THEORY

CHAPTER 1
STATISTICAL THEORY

CHAPTER OBJECTIVES

- To see how the computer can be used to generate random samples of data.
- To understand the difference between a population and a sample.
- To observe that different samples may lead to different sample estimates of population parameters.
- To see that most estimates from samples contain some error of estimation.
- To find a relationship between the size of a sample and the accuracy of the estimate.

The field of statistics uses numerical information obtained from samples to draw inferences about populations. A **population** is a well-defined set of individuals, events, or objects. A **sample** is a selection of individuals, events, or objects taken from a well-defined population. A sample is generally taken from a population with each individual, event, or object being independent and having an equally likely chance of selection. The sample average is an example of a random sample estimate of a population value, i.e., population mean. Population characteristics or **parameters** are inferred from sample estimates, which are called statistics. Examples of population parameters are population proportion, population mean, and population correlation. For example, a student wants to estimate the proportion of teachers in the state who are in favor of year-round school. The student might make the estimate on the basis of information received from a random sample of 500 teachers in the population comprised of all teachers in the state. In another example, a biologist wants to estimate the proportion of tree seeds that will germinate. The biologist plants 1,000 tree seeds and uses the germination rate to establish the rate for all seeds. In marketing research, the proportion of 1,000 randomly sampled consumers who buy one product rather than another helps advertising executives determine product appeal.

Because a sample is only a part of the population, how can the sample estimate accurately reflect the population characteristic? There is an expectation that the sample estimate will be close to the population value if the sample is representative of the population. The difference between the sample estimate and the population value is called **sample error**. In a random sample, all objects have an equal chance of being selected from the population. If the sample is reasonably large, this equally likely chance of any individual, event, or object being selected makes it likely that the random sample will represent the population well. Most statistics are based upon this concept of random sampling from a well-defined population. **Sampling error,** or the error in using a sample estimate as a population estimate does occur. In future chapters, you will learn that several random sample estimates can be averaged to better approximate a population value, although sampling error is still present.

In this chapter, you will use S-PLUS computer software and a STATISTICS program to simulate the sampling of data from a population. You are to determine what proportion of a certain large population of people favor stricter penalties. A random number generator will determine the responses of the people in the sample. A random number generator uses

an initial start number to begin data selection, and then uses the computer to generate other numbers at random. You will use the results of these simulated random samples to draw conclusions about the population.

You will be asked to run the S-PLUS STATISTICS program five times. The program can be run more times if you like. Each time you run the program, the true population proportion will be different. Consequently, each time you will be simulating a sample data set from a different population. The four different random samples will be chosen during each computer run. The random samples have various sample sizes: 5, 20, 135, and 1,280. Using these results, you will be able to observe the effect of sample size on the accuracy of estimation. The Gallop Poll, for example, uses a random sample of 1,500 people nationwide to estimate the presidential election outcome within +/− 2% error of estimation. This chapter will help you understand random sampling and how the sampling error of estimation is determined, i.e., difference between the sample statistic and the population parameter.

SUMMARY

In this chapter, you should learn that:

- A random sample is part of a population.
- Random samples are used to draw inferences about population parameters or characteristics.
- Different random samples lead to different sample estimates.
- The estimate from a sample is usually not equal to the population value.
- If a large sample is taken, the standard error is smaller.

How the program STATISTICS works

The STATISTICS program uses a pseudo-random number generator in S-PLUS to select a random number between 0 and 1 for the true proportion. Next, random samples of only 0 or 1 are drawn from the finite population (not values between 0 and 1). The probability of a 0 is (1− the population proportion) and the probability of a 1 is the same as the population proportion. Random samples of various sizes are taken and the sample proportion and estimation errors are calculated.

The size of the samples varies by the list of sample sizes in the variable *SampleSizes*. The "<−" operator is an assignment operator that places the vector of values (10,100,500,1000,1500) into the variable *SampleSizes*. The **c** before the parentheses means to concatenate these values into a single vector. *NumSamples* is assigned the **length** of the *SampleSizes* vector, which is equivalent to the number of individual sample sizes. *PopProp* is the true proportion in the population and is obtained by taking one random number from a uniform population that is between the values of 0 and 1. The **runif** command means to take a value from a **r**andom **unif**orm population and the number values (1,0,1) correspond to the number of values to be obtained (1), the bottom of the range of values (0), and the top of the range of values (1). There are several other commands within S-PLUS that allow for sampling from other distributions, such as normal **(rnorm),** binomial **(rbinom),** and exponential **(rexp)**.

The most complex parts of the program pertain to creating and using matrices, which will be covered in later chapters. The line which begins with *TrialData* <− **matrix** is setting the size of the matrix and associating labels with the values that will be written to it. The **for** statement begins the processing loop. The for command assigns to the variable **SampleSize** successive values listed in the **SampleSizes** vector. The parentheses are used to mark the beginning and end of the loop encapsulated by this **for** command. The first line within the processing loop creates a vector of values for the first sample and assigns it to the temporary variable *SampleData.* The **sample** command is a way to sample from a finite population, in this case either 0 or 1, but it can also be very useful for taking a subsample of larger samples. The 0:1 denotes the range of *integer* values between 0:1, which only includes 0 and 1, but the same notation could be used to create a vector of integer values from 1 to 10 (1:10). The **prob** keyword sets the probability of getting each value, with 0 having a probability of 1 minus the population proportion (1-*PopProp*) and 1 having a probability of the population proportion. The **size=***SampleSize* assures that this sample is the same size as the one corresponding to the loop iteration, and **replace=T** means to replace values that have been chosen from the population, so this is sampling WITH replacement. If taking a subsample of a larger population, you can request sampling WITHOUT replacement **(replace=F)**.

The next line sums all the zeros and ones from the sample to get the total number of people who were in favor, and then divides that value by the sample size to get the sample proportion. The next to the last line within the processing loop assigns values to one vector within the matrix built earlier in the program for outputting the data. The *i*<-*i*+1 line increments the counter used to keep track of the place within the matrix. The last line of the program produces a printout of the matrix. The values in the *SampleSizes* vector can be changed to simulate small or large sample sizes.

STATISTICS Program Output

	Size	No. in Favor	Sample Prop.	True Prop.	Est. Error
Sample 1	10	6	0600	0.677	0.077
Sample 2	100	59	0.590	0.677	0.087
Sample 3	500	331	0.662	0.677	0.015
Sample 4	1000	670	0.670	0.677	0.007
Sample 5	1500	1037	0.691	0.677	- 0.014

	Size	No. in Favor	Sample Prop.	True Prop.	Est. Error
Sample 1	20	1	0.050	0.098	0.048
Sample 2	200	18	0.090	0.098	0.008
Sample 3	1000	110	0.110	0.098	- 0.012
Sample 4	2000	194	0.097	0.098	0.001
Sample 5	3000	303	0.101	0.098	- 0.003

CHAPTER 1 EXERCISES

1. Run STATISTICS once to obtain the results of people who are in favor of stricter penalties for criminals using the four sample sizes below. Enter the results here.

SAMPLE	SAMPLE SIZE	NO. IN FAVOR	SAMPLE PROPORTION
A	5	_____	_____
B	20	_____	_____
C	135	_____	_____
D	1280	_____	_____

   a. Verify that the sample proportions are correct by using long division or a calculator. To find the sample proportion from the number in favor and the sample size, use the formula:

SAMPLE	SAMPLE PROPORTION=(NO. IN FAVOR)÷(SAMPLE SIZE) COMPUTATION
A	_____
B	_____
C	_____
D	_____

   b. Is the estimate of the population proportion the same for each of the samples? Yes__ No__

   c. Why do you think the sample proportions change? _____ _____

   d. What is the true population proportion?._____

2. The sample proportion (EST) is an estimate of the true population proportion (P). There are errors in the sample estimates.

   a. Calculate the error for each sample. Some of the errors may be positive or negative (Record the +/− sign with the error). Note: ERROR=EST−P.

			SAMPLE PROPORTION	
SAMPLE	SAMPLE SIZE	TRUE	SAMPLE	ERROR
A	5	_____	_____	_____
B	20	_____	_____	_____

C	135	_____	_____	_____
D	1,280	_____	_____	_____

b. Which of the four samples gave the best estimate? _____

3. Run the STATISTICS program 4 more times. Each time, compute the errors in the estimates (P will be different for each program run).

RUN1

SAMPLE	SIZE	TRUE	ERROR
A	5	_____	_____
B	20	_____	_____
C	135	_____	_____
D	1.280	_____	_____
	TRUE P	_____	

RUN 2

SAMPLE	SIZE	TRUE	ERROR
A	5	_____	_____
B	20	_____	_____
C	135	_____	_____
D	1,280	_____	_____
	TRUE P	_____	

RUN 3

SAMPLE	SIZE	TRUE	ERROR
A	5	_____	_____
B	20	_____	_____
C	135	_____	_____
D	1,280	_____	_____
	TRUE P	_____	

RUN 4

SAMPLE	SIZE	TRUE	ERROR
A	5	_____	_____
B	20	_____	_____
C	135	_____	_____
D	1,280	_____	_____
	TRUE P	_____	

a. For the four program runs, what was the largest and smallest amount of error for each sample size? (Disregard the plus or minus sign.)

	SAMPLE SIZE	LARGEST ERROR	SMALLEST ERROR
A	5	_____	_____
B	20	_____	_____
C	135	_____	_____
D	1,280	_____	_____

b. Was the sample proportion from a smaller sample ever a better estimate of the population proportion than the sample proportion from a larger sample? Yes_____ No_____.

c. If yes, for which samples (A,B,C, or D) were the errors smaller?

SMALLER SAMPLE _____	LEAST ERROR_____
LARGER SAMPLE_____	LEAST ERROR_____

d. Why is it possible for a smaller sample to occasionally give a better estimate than a larger sample?

4. Use the previous exercises to draw a conclusion about the effect of sample size on the estimate of the population proportion.

_____

_____

5. A newspaper survey indicates that 62% of the people in a certain state favor a bill to allow retail stores to be open on Sunday. Given the examples you just completed, what additional information would help you interpret this report?

_____

_____

TRUE OR FALSE QUESTIONS

T F a. A sample is part of a population.

T F b. The sample proportion always equals the population proportion.

T F c. The larger the sample size the more likely it is that a sample proportion will be close to the population proportion.

T F d. Each time a different random sample is taken from the same population the sample proportion could be different.

T F e. The sample proportion from a large sample is always a better estimate of the population proportion.

CHAPTER 2

GENERATING RANDOM NUMBERS

CHAPTER OBJECTIVES

- To understand how the computer generates random numbers.
- To understand how samples of random numbers are used in statistics.
- To investigate characteristics of a sequence of random numbers.
- To test the random number generator used by the computer.

Random numbers are used in statistics to investigate the characteristics of different population distributions. We will only be studying the characteristics of the normal distribution. This is because many of the variables that we measure are normally distributed. The statistics we use to test hypotheses about population characteristics based on random samples are created based on certain assumptions and characteristics of the normal distribution. Other population distributions exist (wiebull, hypergeometric, poisson, and elliptical), but we will not be studying their characteristics and associated statistics in the chapter exercises.

Early tables of random numbers helped gamblers to understand their odds of winning. In some instances, exact probabilities or odds of certain outcomes were generated from cards and dice. Today, high-speed computers using computer software with a numerical procedure (algorithm) can produce tables of random numbers. The first mainframe computer, a UNIVAC, produced a set of one million random numbers, which was published in a book by the Rand McNally Corporation. Personal desktop computers today run software that can generate random numbers.

Although many computers have software (mathematical algorithms) to generate random numbers, the software algorithms are not all the same and do not produce the same set of random numbers. Basically, a set of computer-generated numbers is not truly random, so they are called **"pseudo random numbers."** They are called "pseudo random numbers" because the numbers tend to repeat themselves after awhile (repeatedness), correlate with other numbers generated (correlatedness), and don't produce a normal distribution (normality). Consequently, when using a **random number generator,** it is important to report the type of computer used, type of random number generator software (algorithm), start value (start number), repeatedness (when numbers repeat themselves in the algorithm), correlatedness (when numbers begin to correlate in a sequence), and normality (whether or not a normal distribution was produced).

A true random set of numbers has no pattern, and if graphed, would appear as scattered data points across the graph. Because true random numbers have no pattern, the next number generated would not be predicted and would appear with approximately the same frequency as any other number. The concept is simple, but often requires visual confirmation (graph) or other statistical test of randomness and/or normality. Software programs often

include statistical tests for testing the randomness and normality of computer-generated random sample data.

In this chapter you will execute the random number generator in the S-PLUS program. Because all random number generators are not the same, acceptable properties for these random numbers should include:

1. Approximate, equal proportions of odd and even numbers should occur.
2. Each number between 0 and 9 is generated approximately one-tenth of the time.
3. For the five consecutive sets of generated number combinations, the percentages should be approximately equal to the theoretical probabilities.

SUMMARY

In this chapter, you should learn that:

- A sequence of random numbers is not truly random (unique).
- A sequence of random numbers is typically unpredictable, but a long sequence of random numbers will tend to repeat, correlate, and not appear normal.
- Our expectation is that about half of the generated numbers are odd and half are even.
- The frequency of occurrence for any random integer between 0 and 9 is approximately one-tenth of the time.
- A set of randomly generated numbers can be tested for randomness and normality.

How the RANDOM program works

The RANDOM program tests the randomness of the pseudo-random number generator used in S-PLUS. The majority of the program code classifies combinations of numbers and formats the output. The creation of sample data and the calculation of the relative frequencies of odd and even digits, and each individual digit, are all contained within the first few lines of the program. Random numbers are sampled from the integer values 0 through 9; the relative frequency of the odd numbers is determined using the modulus (or remainder function) in combination with the **sum** function and dividing by the sample size. The relative frequency of the even numbers is determined in the same manner, only using all values that were not determined to be odd *(SampleSize*−**sum***(SampleData*%%2*))*. The relative frequency of each digit is determined by the familiar **factor** and **table** combination, and then all raw data are put into groups of five numbers.

The main processing loop of the program is used primarily to classify the groups of numbers based on various combinations of repeat values. It iterates from 1 to the number of rows in the *GroupedData* matrix, which is the first dimension **(dim)** of that matrix. The first line within the matrix concatenates the values within the current row with no space separation between them (**sep**=“”). Next, a double use of the **table** and **factor** function combination yields the various amounts of repeat values within the sample group. The loop begins from the inside and works out. First, the number of times that each number (0 to 9) comes up within the group is tallied, then the outer pair of **factor** and **table** functions count

how many of each number of repeats (three of a kind, four of a kind, etc.) are in the group. The next few lines of code use the information contained within the vector just created to classify the different combinations of repeats into unique event values. The event values are fairly arbitrary in this program, unlike earlier programs that used the binary coding scheme, and could really be anything as long as they were matched up with the appropriate labels when they were output. Finally, the last line within the processing loop adds the raw group of numbers to an output vector.

The vectors are factored and tabled to determine how many of each unique event occurred within the sample. The next line builds an output matrix from the relative frequencies that were determined at the beginning of the program, along with their theoretical probabilities, which have been typed directly into the program instead of being calculated. After this, dimension names are assigned for a matrix, a matrix of the event relative frequencies is built, and dimension names are subsequently assigned to that matrix. Finally, the output begins with the groups of numbers from the sample being printed with the cat function using the keyword **fill** to assure that lines greater than 80 characters will be wrapped to the next line. Then the two output matrices are printed using the **print.char.matrix** function with a fixed number of decimal places. The **scientific** keyword was used in the second case because there was a problem with some values being represented in scientific notation due to the fact that the default is to print anything with its lead digit more than four places from the decimal in scientific notation. This change increased output to six places.

The RANDOM program allows you to adjust the sample size until you find one that closely approximates the theoretical probabilities. An experienced S-PLUS programmer could easily create an outer loop that would permit the use of automatic iterations for increasingly larger sample sizes. One could also determine the sum of squared errors within each area for a test of randomness (relative frequency of even/odd, relative frequency of each digit, and relative frequency of combinations). These results could then be output in a table with the corresponding sample size to determine which sample size yields a reasonably close estimate to the theoretical probabilities. Additionally, a line graph could be created to view the relationship between the sum of squared error and different sample sizes. The S-PLUS programming required to accomplish these tasks, however, is beyond the scope and activity for the examples in this chapter.

RANDOM Program Output

```
Number groups:
86526 76376 99385 00396 54480 28574 05502 20873 10308 69425 69363 05530
                    |Relative Frequency  |Probability       |
--------------------+--------------------+------------------+
Odd                 |0.45                |0.50              |
--------------------+--------------------+------------------+
Even                |0.55                |0.50              |
--------------------+--------------------+------------------+
0                   |0.17                |0.10              |
--------------------+--------------------+------------------+
1                   |0.02                |0.10              |
--------------------+--------------------+------------------+
```

2	0.08	0.10
3	0.13	1.10
4	0.07	0.10
5	0.15	0.310
6	0.13	0.10
7	0.07	0.10
8	0.10	0.10
9	0.08	0.10

	Relative Frequency	Probability
No duplicates	0.2500	0.3024
One pair	0.4167	0.5040
One triple	0.0000	0.0720
Two pairs	0.3333	0.1080
Pair & triple	0.0000	0.0090
Four alike	0.0000	0.0045
All alike	0.0000	0.0001

CHAPTER 2 EXERCISES

1. Run RANDOM for N=60. Record the twelve groups of five numbers (5*12=60 numbers) in the blanks below.

Numbers: _____ _____ _____ _____

_____ _____ _____ _____

_____ _____ _____ _____

Write the results below.

	RELATIVE FREQUENCY	PROBABILITY
ODD		
EVEN		
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

a. Check that the relative frequencies are correct for the 60 numbers.
b. What is the largest absolute difference between the relative frequencies and the probabilities?_____
c. How is the relative frequency for ODD related to the relative frequencies for the ten digits (0–9)?_____

2. Complete the following table from the run of RANDOM for N=60.

	RELATIVE FREQUENCY	PROBABILITY
NONE		
ONE PAIR		
ONE TRIPLE		
TWO PAIRS		
PAIR & TRIPLE		
FOUR ALIKE		
ALL ALIKE		

a. Look at the twelve groups of five numbers recorded in Exercise 1.
Have the duplicates been counted correctly and the relative frequencies computed correctly?
RELATIVE FREQUENCY=FREQUENCY/(NUMBER OF GROUPS OF 5)

b. How is the probability of ALL ALIKE calculated?

_____

_____

c. Find the sum of the relative frequencies._____
Why does the sum have this value? _____

_____ _____

3. Run RANDOM for N=200.

a. What is the maximum absolute value of the differences between the relative frequencies and their respective probabilities? _____

b. What is the maximum absolute difference between the relative frequencies of the duplicates and their respective probabilities? _____

4. Run RANDOM for N=500.

a. What is the maximum absolute value of the differences between the relative frequencies and their respective probabilities? _____

b. What is the maximum absolute difference between the relative frequencies of the duplicates and their respective probabilities?_____

5. On the basis of the runs for N=200 and N=500, are you satisfied with the performance of the random number generator? _____. Why, or why not?

_____

_____

_____

TRUE OR FALSE QUESTIONS

T	F	a.	It is easy to recognize a set of random numbers.
T	F	b.	In any set of truly random numbers, exactly half are even.
T	F	c.	If five of a kind appears consecutively once in a sequence of 10,000 numbers, this is evidence that the numbers may not be random.
T	F	d.	In a group of five random numbers, it is more probable that a pair will be found, than finding all of the numbers to be different.
T	F	e.	About seven times out of one hundred in a group of five random digits, a triple will appear.

CHAPTER 3

FREQUENCY DISTRIBUTIONS

CHAPTER OBJECTIVES

- To observe the shapes of histograms and corresponding ogives
- To develop an understanding of histograms and ogives
- To demonstrate the relationship between the shape of a histogram and its corresponding ogive.

A **histogram** is a graph of a frequency distribution of numerical data for different categories of events, individuals, or objects. A **frequency distribution** indicates the individual number of events, individuals, or objects in the separate categories. Most people easily understand histograms because they resemble bar graphs often seen in newspapers and magazines. An ogive is a graph of a cumulative frequency distribution of numerical data from the histogram. A **cumulative frequency distribution** indicates the successive addition of the number of events, individuals, or objects in the different categories of the histogram, which always sums to 100. An ogive graph displays numerical data in an S-shaped curve with increasing numbers or percentages that eventually reach 100%. Because cumulative frequency distributions are rarely used in newspapers and magazines, most people never see them. Frequency data from a histogram, however, can easily be displayed in a cumulative frequency ogive.

This chapter will provide you with an understanding of the histogram and its corresponding ogive. You will gain this experience quickly without the work involved in data entry and hand computation. You will be able to view the histogram and cumulative frequency distributions for different sample data sets. The S-PLUS program can be used to display the histogram frequency distributions and ogive cumulative frequency distributions.

To simplify the graphical display and provide similar comparisons between the types of histograms, all histograms in the S-PLUS program will have ten categories. The data for each category are not listed; rather the categories are numbered 1 to 10. You will be asked to enter the frequency for each of the ten categories and the frequencies must be integers greater than zero. The S-PLUS program will print a table listing the frequencies you specified, the relative frequencies, and the less-than-or-equal cumulative relative frequencies. The S-PLUS program prints a histogram and a corresponding ogive, which is output in a separate window (GSD2).

SUMMARY

In this chapter, you should learn that:

- Histograms and ogives have different shapes and vary depending on frequency.
- An ogive always increases from 0% to 100% for cumulative frequencies.

- The shape of a histogram determines the shape of its related ogive.
- A uniform histogram is flat; its ogive is a straight line sloping upward.
- An increasing histogram has higher frequencies for successive categories; its ogive is concave and looks like part of a parabola.
- A decreasing histogram has lower frequencies for successive categories; its ogive is convex and looks like part of a parabola.
- A uni-modal histogram contains a single mound; its ogive is S-shaped.
- A bi-modal histogram contains two mounds; its ogive can be either reverse S-shaped or double S-shaped depending upon the data distribution.
- A right-skewed histogram has a mound on the left and a long tail on the right; its ogive is S-shaped with a large concave portion.
- A left-skewed histogram has a mound on the right and a long tail on the left; its ogive is S-shaped with a large convex portion.

How the program FREQUENCY works

The part of the program that can be changed is a list of values relating to a score distribution observed in a given classroom. The length of this list does not matter; it is never specifically referenced in the program. The *Class* object is given a value for a vector of numbers using the c function that was introduced in Chapter 1. Each number within the vector is divided by the sum of all values within the vector. In most programming languages this would involve creating a processing loop similar to the one in the RANDOM program that would add up all the values and then go back and divide each value by the total. In S-PLUS, vectors can be easily created and used. For the FREQUENCY program, the first processing loop is replaced by the simple **sum***(Class),* which gets a total for all of the values, and this result is then divided into each of the values within the vector by simply typing *Class/***sum***(Class)*. No additional step is necessary. This feature in S-PLUS makes the language relatively short and easy to follow.

The next program line follows the same logic, only the cumulative sum **(cumsum)** of the vector is determined at each point and these values are divided by the overall sum of the values to give a vector of values labeled *CumRelFreq*. Scaling of the histogram height is performed next so that the histogram bars are not too small compared to the vertical scaling of the graph. The "if then else" clause is used to provide vertical scaling that will be either one tenth greater than the highest relative frequency, or 1 if the value is .95 or above. The **round** function is implemented to insure that the maximum value is set to an even tenth **(digits=1)**. The **barplot** function is used to draw the histogram of the relative frequencies with the **RelFreq** vector as the specified target. The **plot** function is used to draw the ogive of the cumulative relative frequencies with **CumRelFreq** as the target.

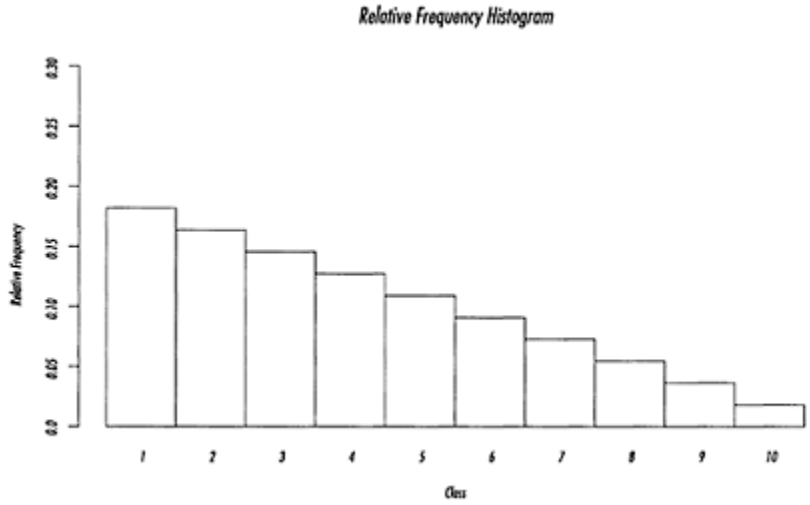
The last part of the FREQUENCY program builds a matrix of the class score distribution along with the associated relative frequencies and cumulative relative frequencies. The line beginning *TableData* **<- matrix** prepares the matrix and initializes all values within it to 0 and makes the dimensions of the matrix to be **length***(Class)* rows and 3 columns. The **dimnames** keyword sets the labels for the dimensions and will be used in later chapters with other vectors. The **for** loop iterates from 1 to the number of values within *Class* and assigns each row within the *TableData* matrix to the respective *Class* vector value, relative

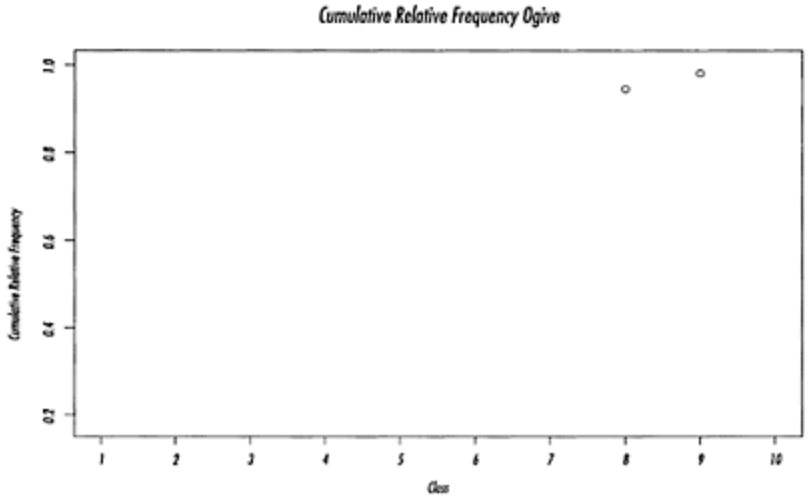
frequency, and cumulative relative frequency, rounding each frequency to three decimal places. You will see some error in the cumulative numbers due to the rounding of the cumulative values. The final line of the program simply prints out the *TableData* matrix.

You can change the values within the *Class* vector to obtain different shaped histograms and corresponding ogives. The original vector of 10 values breaks the score distribution into 10 intervals, but this can be changed to create histograms with greater resolution. You could comment out both lines of "**if"** and **"else"** statements that scale the histogram by prefixing them with **"#"** signs to see the effect of not scaling it properly to fit the plot; replace these statements with the ***PlotHeight* <- 1** statement by removing the # sign in front of it. Some rounding error does occur in the program when summing the relative frequencies to obtain the cumulative relative frequencies.

FREQUENCY Program Output

	Freq.	Relative Freq.	Cum. Rel. Freq.
Class 1	50	0.182	0.182
Class 2	45	0.164	0.345
Class 3	40	0.145	0.491
Class 4	35	0.127	0.618
Class 5	30	0.109	0.727
Class 6	25	0.091	0.818
Class 7	20	0.073	0.891
Class 8	15	0.055	0.945
Class 9	10	0.036	0.982
Class 10	5	0.018	1.000



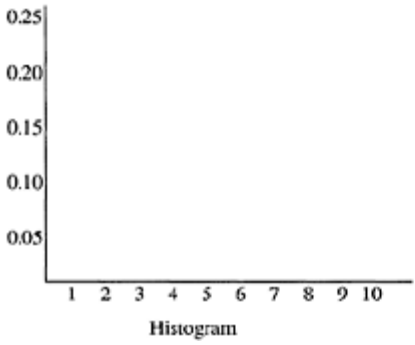


CHAPTER 3 EXERCISES

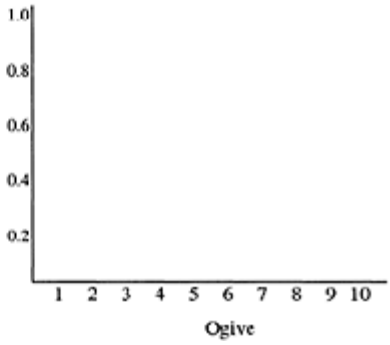
1. Run FREQUENCY program six times (a to f). Enter the frequencies listed for each type of histogram in the Class array statement. For each run, complete the frequency table and draw sketches of the histogram and corresponding ogive.

   a. A UNIFORM HISTOGRAM

CLASS	FREQ	REL FREQ	CUM REL FREQ
1	5	_____	_____
2	5	_____	_____
3	5	_____	_____
4	5	_____	_____
5	5	_____	_____
6	5	_____	_____
7	5	_____	_____
8	5	_____	_____
9	5	_____	_____
10	5	_____	_____



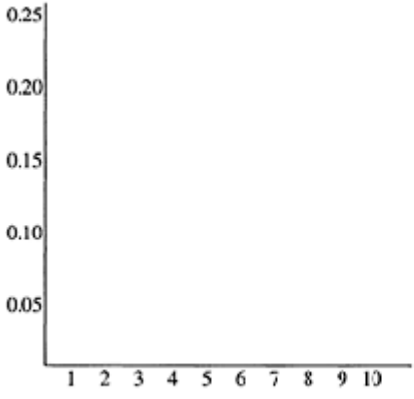
Histogram



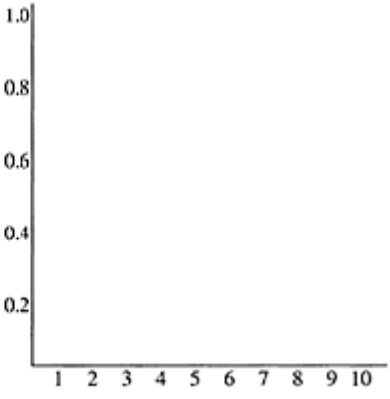
Ogive

b. AN INCREASING HISTOGRAM

CLASS	FREQ	REL FREQ	CUM REL FREQ
1	10	_____	_____
2	12	_____	_____
3	14	_____	_____
4	16	_____	_____
5	18	_____	_____
6	20	_____	_____
7	22	_____	_____
8	24	_____	_____
9	26	_____	_____
10	28	_____	_____



Histogram



Ogive