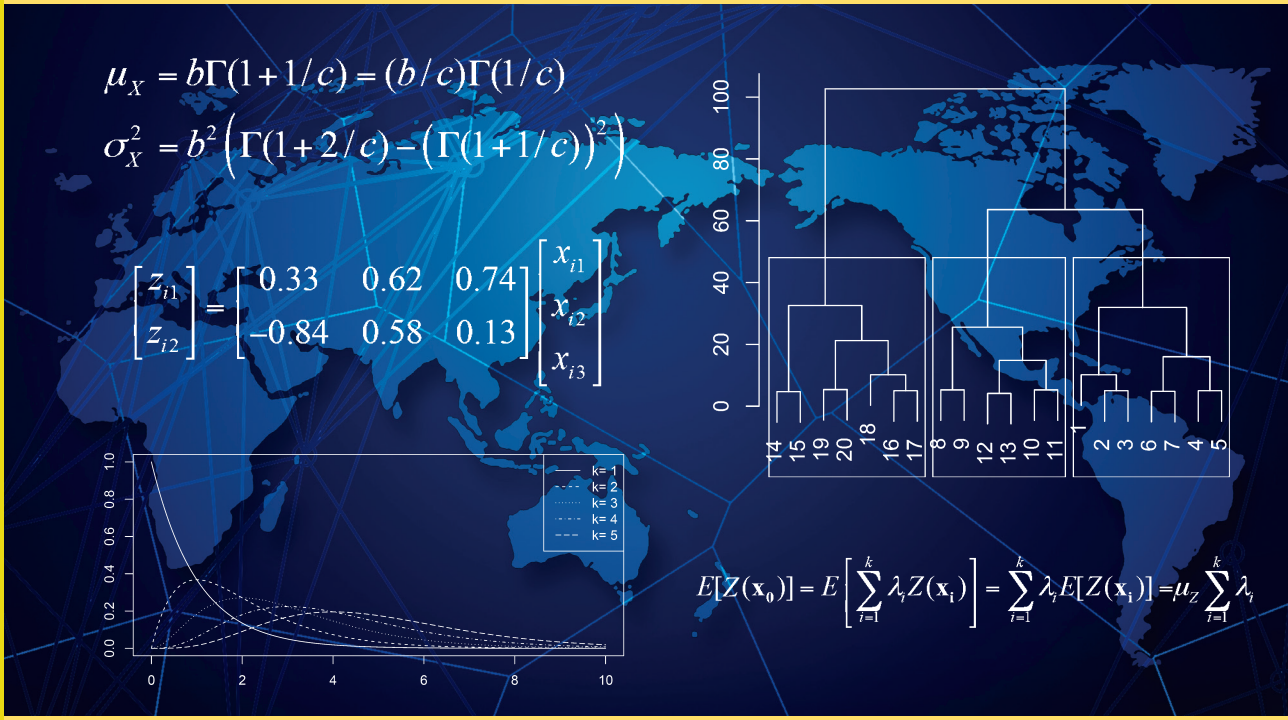
DATA ANALYSIS and STATISTICS

for Geography, Environmental Science, and Engineering

MIGUEL F. ACEVEDO



CRC Press
Taylor & Francis Group

DATA ANALYSIS and STATISTICS

for Geography, Environmental Science, and Engineering

DATA ANALYSIS and STATISTICS

for Geography, Environmental Science, and Engineering

MIGUEL F. ACEVEDO

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20121213

International Standard Book Number-13: 978-1-4665-9221-6 (eBook - PDF)

Contents

Preface

This book evolved from lecture notes and laboratory manuals that I have written over many years to teach data analysis and statistics to first-year graduate and fourth-year undergraduate students. I have developed this material during 15 years while teaching a first-year graduate course in quantitative techniques for the Applied Geography and the Environmental Sciences program at the University of North Texas (UNT). In that course, we focus on data analysis methods for problem solving in geographical and environmental sciences, emphasizing hands-on experience. Quantitative methods applied in these sciences share many attributes; of these, we emphasize the capabilities to analyze multiple factors that vary both spatially and temporally.

Statistical and probabilistic methods are the same in a broad range of disciplines in science and engineering, and so are the computational tools that we can use. Methods may vary by discipline either because of academic tradition or because of the priority given to certain problems. However, methods not traditionally employed in a discipline sometimes become part of its arsenal as priorities shift and methods are "imported" from other fields where they have shown to be effective.

Some of the principles inspiring this book are that educating twenty-first-century scientists and engineers in statistical and probabilistic analysis requires a unified presentation of methods, the inclusion of how to treat data that vary in space and time, as well as multiple dimensions, and a practical training of how to perform analysis using computers. Furthermore, given the importance of interdisciplinary work in sustainability, this book attempts to bring together methods applicable across a variety of science and engineering disciplines dealing with earth systems, the environment, ecology, and human–nature interactions. Therefore, this book contributes to undergraduate and graduate education in geography and earth science, biology, environmental science, social sciences, and engineering.

OVERVIEW

I have divided the book into two parts:

- Part I, Chapters 1 through 8: Probability, random variables and inferential statistics, applications of regression, time series analysis, and analysis of spatial point patterns
- Part II, Chapters 9 through 15: Matrices, multiple regression, dependent random processes and autoregressive time series, spatial analysis using geostatistics and spatial regression, discriminant analysis, and a variety of multivariate analyses based on eigenvector methods

The main divide between the two parts is the use of matrix algebra in Part II to address multidimensional problems, with Chapter 9 providing a review of matrices.

Although this organization may seem unconventional, it allows flexibility in using the book in various countries, various types of curricula, and various levels of student progress into the curriculum. In the United States, for example, most undergraduate students in the sciences do not take a linear algebra course, and in some engineering programs, linear algebra is not required until the third year (juniors) upon completion of a second calculus class. In other countries and many U.S. engineering programs, colleges expose their undergraduate students to matrix algebra earlier in the curriculum; for example, engineering students in China typically take linear algebra in their second year of college. Therefore, I have left the multidimensional material for last after a substantial review of matrix algebra, allowing the students to become familiar with time series and spatial analysis at an earlier stage.

USE OF THE BOOK

There are several ways to use this book. For example, a junior-level third-year course for undergraduate students can cover The eight chapters of Part I at a rhythm of about two chapters per week during a typical 15-week semester. A more challenging or honors section could include the review of matrices (Chapter 9) and a couple of chapters from Part II. A senior-level combined with first-year graduate course can be based on the entire book. Depending on the students' background, the course could cover the material of Part I as a review (except spatial analysis and time series) in order to spend the majority of time on the topics presented in Part II. My experience has been that last-year (senior) undergraduate and first-year graduate students in the sciences are unfamiliar with matrix algebra and would need Chapter 9. However, a senior-level undergraduate or graduate engineering course may not need coverage of this chapter, except for the computer session. Many graduate students would be familiar with the material in the first two chapters of Part I, and they could read it rapidly to refresh the concepts. For other students, this material may be new and may require additional reading beyond the basics provided here.

PEDAGOGY

Each chapter starts with conceptual and theoretical material covered with enough mathematical detail to serve as a firm foundation to understand how the methods work. Over the many years that I have used this material, I have confirmed my belief that students rise to the challenge of understanding the mathematical concepts and develop a good understanding of basic statistical analysis that facilitates their future learning of more advanced and specialized methods needed in their profession or research area. To facilitate learning these concepts, I have included examples that illustrate the applications and how to go from concepts to problem solving. The conceptual and theoretical section ends with exercises similar to the examples.

In each chapter, a hands-on computer session follows the theoretical foundations, which helps the student to "learn by doing." In addition, this computer session allows the reader to grasp the practical implications of the theoretical background. This book is not really a software manual, but the computer examples are developed with sufficient detail that the students can follow and perform themselves either in an instructor-supervised computer classroom or lab environment or unassisted at their own pace. This design gives maximum flexibility to the instructor and the student. In a similar fashion to the theoretical section, the computer session ends with exercises similar to the examples.

COMPUTER EXAMPLES AND EXERCISES

I have organized the computer examples using the R system, which is open source. This is very simple to download, install, and run. As some authors put it, R has evolved into the *lingua franca* of statistical computing (Everitt and Hothorn, 2010). R competes with major systems of scientific computing, yet because it is open source, it is free of commercial license cost while having access to thousands of packages to perform a tremendous variety of analysis. At the time of this writing, there are 3398 packages available. Even students with no prior knowledge of programming are quickly acquainted with the basics of programming in R. For those users who still prefer a graphical user interface (GUI), there is diversity of GUIs also available as open source.

R is a GNU project system (GNU stands for Gnu's Not Unix). The GNU project includes free software and general public license. R is available from the comprehensive R archive network (CRAN), the major repository mirrored all over the world. The simplest approach is to download the precompiled binary distribution for your operating system (Linux, Mac, or Windows). In this book, we will assume a Windows installation because it is a very common situation in university environments, but the computer exercises given here would work under all platforms.

In addition to the R GUI, there are several other GUIs available to simplify or extend R. For example, (1) a web GUI to enter commands over a web browser, which can be used from smart phones and pads with web access, and (2) the R Commander GUI and its several plug-in packages to simplify entering commands.

As mentioned earlier, students can execute the computer examples and exercises in the classroom environment or at their own pace using their computers. Over the years, I have tested this material in both modes. I conduct a weekly instructor-supervised session in the computer classroom, where I run demonstrations from the instructor machine equipped with a projector or systematic instructions followed by students in their assigned computer or simply letting the students follow the instructions given in the book and asking for help as needed. Students can go to the computer lab to work on their assigned exercises or complete the assignments off-campus by installing R on their computers or running R from a browser.

HOW TO USE THE BIBLIOGRAPHY

In each chapter, I provide suggestions for supplementary reading. These items link to several textbooks that cover the topics at similar levels and have been written for different audiences. This can help students read tutorial explanations from other authors. Often, reading the same thing in different words or looking at different figures helps students to understand a topic better. In addition, the supplementary readings point to several introductory texts that can serve to review. I have also included references that provide entry points or hooks to specialized books and articles on some topics, thus helping advanced students access the bibliography for their research work.

SUPPLEMENTARY MATERIAL

Packages seeg and RcmdrPlugin.seeg available from CRAN provide all data files and scripts employed here. These are also available via links provided at the Texas Environmental Observatory (TEO) website www.teo.unt.edu and the author's website, which is reachable from his departmental affiliation website www.ee.unt.edu. The publisher also offers supplementary materials available with qualifying course adoption. These include a solutions manual and PowerPoint® slides with figures and equations to help in preparing lectures.

Miguel F. Acevedo
Denton, Texas

Acknowledgments

I am very grateful to the many students who took classes with me and used preliminary versions of this material. Their questions and comments helped shape the contents and improve the presentation. My sincere thanks to several students who have worked as teaching assistants for classes taught with this material and helped improve successive drafts over the years, in particular H. Goetz and K. Anderle who kindly made many suggestions. I would also like to thank the students whom I have guided on how to process and analyze their thesis and dissertation research data. Working with them provided insight about the type of methods that would be useful to cover in this textbook.

Many colleagues have been inspirational, to name just a few: T.W. Waller and K.L. Dickson, of the UNT Environmental Science program (now emeritus faculty), M.A. Harwell (Harwell Gentile & Associates, LC), D.L. Urban (Duke University), M. Ataroff and M. Ablan (Universidad de Los Andes), and J. Raventós (Universidad de Alicante), and S. García-Iturbe (Edelca).

Many thanks to Irma Shagla-Britton, editor for environmental science and engineering at CRC Press, who was enthusiastic from day one, and Laurie Schlags, project coordinator, who helped immensely in the production process. Several reviewers provided excellent feedback that shaped the final version and approach of the manuscript.

Special thanks to my family and friends, who were so supportive and willing to postpone many important things until I completed this project. Last, but not least, I would like to say special thanks to the open source community for making R such a wonderful tool for research and education.

Author

Miguel F. Acevedo has 38 years of academic experience, the last 20 of these as faculty member of the University of North Texas (UNT). His career has been interdisciplinary, especially at the interface of science and engineering. He has served at UNT in the Department of Geography, the Graduate Program in Environmental Sciences of the Department of Biology, and more recently in the Department of Electrical Engineering.

Dr. Acevedo received his PhD in biophysics from the University of California, Berkeley (1980) and his MS and ME in computer science and electrical engineering from the University of Texas at Austin (1972) and from Berkeley (1978), respectively. Before joining UNT, he was at the Universidad de Los Andes, Merida, Venezuela, where he taught since 1973 in the School of Systems Engineering, the Graduate Program in Tropical Ecology, and the Center for Simulation and Modeling (CESIMO).

Dr. Acevedo has served on the Science Advisory Board of the U.S. Environmental Protection Agency and on many review panels of the U.S. National Science Foundation. He has received numerous research grants and has written many journal and proceeding articles as well as book chapters. UNT has recognized him with the Regents Professor rank, the Citation for Distinguished Service to International Education, and the Regent's Faculty Lectureship.

Part I

Introduction to Probability, Statistics, Time Series, and Spatial Analysis

1 Introduction

In this introductory chapter, we start with a brief historical perspective of statistics and probability, not to exactly account for its development, but to give the reader a sense of why statistics, while firmly grounded in mathematics, has an empirical and applied flavor.

1.1 BRIEF HISTORY OF STATISTICAL AND PROBABILISTIC ANALYSIS

In the western world, **statistics** started in the seventeenth century a little over 300 years ago attributed by many to John Graunt who attempted to relate data of mortality to public health by constructing life tables (Glass, 1964). Thus, the birth of statistics relates to solving problems in demography and epidemiology, subjects very much at the heart of geographical science and ecology. It is interesting to note that Graunt's 1662 book *Natural and Political Observations Made upon the Bills of Mortality* took an interdisciplinary approach linking counts of human mortality at various ages to public health. In addition, the word "observations" in the title of his book illustrates statistics' **empirical** heritage, i.e., collecting and organizing data and performing analysis driven by these data.

Subsequently, and for more than a century, statistics helped governments, or **stat**es, analyze demographical and economical issues. That is how statistics got its name; from the word "Statistik" used in Germany and the Italian word "statista" for public leader or official.

There has been some historical interest in determining whether statistics was born much earlier because of the Greek's achievements in mathematics and science, the Egyptian's censuses conducted to build the pyramids, and evidence of tabular form of data in China. However, the Greeks did not build an axiomatic apparatus as the one they had for geometry (Hald, 2003). Some attribute early use of data in tables in China to Yu the Great (~2000 years BC), founder of the Xia dynasty, who also developed flood control, a human–nature interaction of relevance to sustainability even today in many parts of the world. It seems that early use of data compilation was associated with geographical description of the Chinese state including keeping data in tabular form on number of households, in the various provinces and economic production (Bréard, 2006). Summation and calculation of means is present in the seventh-century compilation *Ten Books of Mathematical Classics*. Officials used means to calculate grain consumed per person and tax payment required per household (Bréard, 2006).

Back to Europe, also during the seventeenth century, Pierre de Fermat and Blaise Pascal, motivated by attempts to analyze games of chance, established the mathematical basis of **probability theory**. Chevelier de Mere piqued mathematical interest to solve a famous game of chance problem that was around for a century. Jacob Bernoulli and Abraham de Moivre continued to develop probability theory during the eighteenth century. A legacy of this period is the cornerstone discovery that averaging the outcomes for a **large number** of trials yields results approaching those **expected** by theory. Also from the eighteenth century, Thomas Bayes contributed the concept of conditional probabilities, and Pierre-Simon Laplace contributed the central limit theorem, inspired in calculating the distribution of meteor elevation angles. Laplace's paper on the central limit theorem appeared in the first years of the nineteenth century and he continued to be very active well into that century.

Major thrusts of the nineteenth century included how to use probability to deal with uncertainty in the natural sciences. This quest furthered the link between theory and observations, as the theory of errors was developed to solve problems in geodesy and astronomy. Inspired by this problem,

Carl Friedrich Gauss contributed a jewel for prediction models, the **least-squares** method. Then, as that century ended, statistical mechanics became a milestone in physics employing probability theory to explain macroscopic behavior (e.g., temperature of a gas) from microscopic behavior (random motion of a large number of particles).

During the ending years of the nineteenth century and first part of the twentieth century, statistical approaches and probability theory were further integrated. In the first quarter of the twentieth century, R.A. Fisher introduced the concepts of inductive inference, level of significance, and parametric analysis. These concepts served as the basis for estimation and hypothesis testing and were further developed by J. Neyman and E.S. Pearson. Together with this enhanced linking of the empirical approaches of statistics and theoretical approaches of probability, there was an increased use in social sciences (e.g., economics, psychology, sociology), natural sciences (biology, physics, meteorology), as well as in industry and engineering (e.g., ballistics, telephone systems, computer systems, quality control). Major methods in factor analysis and time series were developed for a variety of applications. Such an integration of empirical and theoretical approaches saw an increase through the twentieth century in specialized fields of science and engineering.

In the latest part of the twentieth century at AT&T Bell Laboratories (now Lucent Technology), major development included theoretical contributions by J. Tukey and several others, and the language S, predecessor of R used in this book (Becker et al., 1988; Chambers and Hastie, 1993; Crawley, 2002).

This brief historical account helps us understand how statistics came to be firmly grounded on mathematics, and at the same time be rather unique in its emphasis due to its empirical heritage and applicability in many fields. Thus, statistics and probability play a central role in applied mathematics. Interested students can read more on prospects for directions of statistics in the twenty-first century in several references (Gordon and Gordon, 1992; Raftery et al., 2002).

1.2 COMPUTERS

Computers and networks of computers became readily accessible in the latest part of the twentieth century, having two major kinds of impacts on statistics and probabilistic analysis: (1) we can now easily perform complicated calculations and visualize the results, and (2) we now have an amazing availability of data in an increasingly networked cyber-infrastructure.

Therefore, in addition to understanding the theory, it is important to acquire the computer skills to be a successful practitioner of statistical and probabilistic analysis in any modern profession. In particular, the integration of theoretical and computational aspects facilitates statistical analysis as applied in many areas of social and natural sciences, as well as medicine and engineering. Moreover, the computational aspects are not limited to the ability to perform a calculation but to manage data files and visualize results.

1.3 APPLICATIONS

Many examples in this book are from ecology, geosciences, environmental science, and engineering, not only because of my experience in these areas, but because this book attempts to bring together methods that can be applied in sustainability science and engineering. The complexity of environmental problem solving requires professionals skilled in computer-aided data and statistical analysis.

1.4 TYPES OF VARIABLES

A **variable** represents a characteristic of an object or a system that we intend to measure or to assign values, and of course, that varies (Sprinthall, 1990). Take for example time t and water level of a stream h. A variable can take one of several or many values; for example, time $t = 2$ days denotes

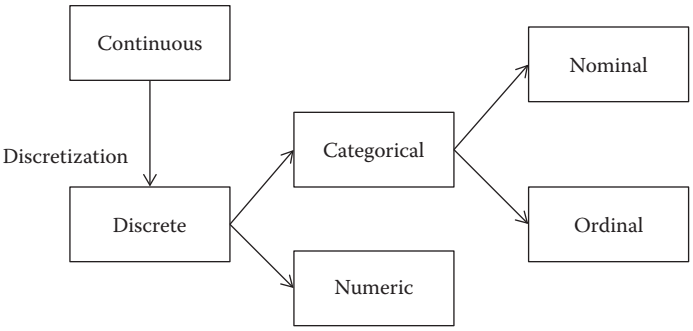


FIGURE 1.1 Simple classification of variables according to the nature of the values.

that t takes the value 2 days, and $h = 1.1$ m that water level takes the value 1.1 m. It is common to classify variables in several types (Figure 1.1).

1.4.1 Discrete

There are variables that can take values from a **discrete** set of values, say the integers from 0 to 10, or colors red, green, and blue. The values can be numbers and the variable is **numeric**, or just categories (as colors in the example) and the variable would be **categorical** (Figure 1.1). We can use numbers 1, 2, 3 to code the categories, but the variable is still categorical. Numeric discrete variables often result from counting; for example, number of red balls and number of individuals in a classroom.

Moreover, we can distinguish **nominal** vs. **ordinal** categorical variables. When the categories have no underlying order, the variable is **nominal**, but if there is a meaningful order, the variable is **ordinal** (Figure 1.1). For an example of a nominal variable, think of three values given by colors red, blue, and green; or of two values head and tails from a coin toss. For an example of ordinal variables, think of qualitative scores low, medium, and high, or a scale used in polling or assessment 1 to 5, where 1 denotes "strongly disagree" and 5 the other extreme "strongly agree," with 3 denoting "agree" in the middle. In this last case, the ordinal values are coded with numbers but the variable is still categorical, not numeric.

1.4.2 Continuous

Other variables can take **real** values and therefore can theoretically have infinite number of values. For example, h = water level in a stream could take any value in the interval from zero to a maximum. We refer to this type of variable as **continuous**. Many measurements of physical, chemical, and biological properties yield continuous variables. For example, air temperature, wind speed, salinity, and height of individuals.

1.4.3 Discretization

We can convert continuous variables into discrete variables by **discretization**, i.e., dividing the range in intervals, and using these to define the values (Figure 1.1). Counting how many times the measurements fall in each interval yields a discrete numeric variable. For example, divide air temperature T in three intervals—cold = $T < 0$°C, medium = 0°C $\leq T <$ 10°C, and hot = 10°C $\leq$ T—and we now have a discrete variable with values cold, medium, and hot. Note that we need to make sure that boundaries are clear, for example, cold extends to less than zero but does not include zero, which is included in the medium interval. Counting the number of days such that the maximum temperature of the day falls in these intervals yields a numeric variable.

1.4.4 Independent vs. Dependent Variables

In addition, two major classes of variables are **independent** and **dependent** variables. We assume that the independent forces or drives the dependent, as in cause–effect, or factor–consequence. In some cases, this distinction is clear and, in other cases, it is an arbitrary choice to establish the predictor of a dependent variable Y based on the independent variable X. As the popular cautionary statement warns us, we could draw wrong conclusions about cause and effect when the quantitative method employed can only tell about the existence of a relationship.

1.5 PROBABILITY THEORY AND RANDOM VARIABLES

A good grasp of **probability** concepts is essential to understand basic statistical analysis, and in turn learning basic statistics is essential to understand advanced methods. These include regression, multivariate analysis, spatial analysis, and time series. Defining events and using probability trees are basic and very useful concepts, as well as using conditional probabilities and Bayes' theorem.

Random variables are those with values associated to a function defining the probability of their occurrence. **Distribution** is a general word used for this correspondence between the values and their probabilities. When the variable is discrete, this function is a **probability mass function (pmf)**, and when the variable is continuous, the function is a **probability density function (pdf)**. Accumulation of probability along the possible values of the variable leads the definition of **cumulative** functions. Then, calculations on the distributions are **moments**, such as the **mean** and **variance**. As we will see in forthcoming chapters, the law of large numbers and the central limit theorem are crucial to linking theoretical distributions and their moments to **samples**. Using computers, we can generate numbers that look random and seemingly drawn from an assumed distribution.

1.6 METHODOLOGY

Terms and methods will become clear as we go through the different chapters. When tackling quantitative problems, we can use the following general methodological framework.

1. Problem definition
   a. Define the questions to be answered by the analysis
   b. Define possible assumptions that could simplify the questions
   c. Identify components and their relationships aided by graphical block diagrams and concept maps
   d. Identify domains and scales in time and space
   e. Identify data required and sources
   f. Design experiments for data collection (in this case would need to go to step 4 and return to step 2)
2. Data collection and organization
   a. Measurements and experiments
   b. Collecting available data
   c. Organize data files and metadata
3. Data exploration
   a. Variables, units
   b. Independent and dependent variables
   c. Exploratory data analysis
   d. Data correlations
4. Identification of methods, hypotheses, and tests
   a. Identify hypotheses
   b. Identify methods and their validity given the data

5. Analysis and interpretation
   a. Perform calculations
   b. Answer to the questions that motivated the analysis
   c. Describe limits of these answers given the assumptions
   d. Next steps and new hypotheses
6. Based on the results, return to one of steps 1–4 as needed and repeat

1.7 DESCRIPTIVE STATISTICS

Step 3 of the methodological process just outlined includes applying **descriptive statistics**. By descriptive we mean that we are not attempting to make inferences or predictions but mainly to characterize the data, typically a **sample** of the theoretical population defined by the distribution. In simpler words, we want to tell what the data look like. Here, we use the term **statistic** to refer to a calculation on the sample, such as the sample mean and variance that we will discuss in Section 1.14. **Exploratory Data Analysis** refers to performing descriptive statistics, and using a collection of visual and numerical tools such as quantile–quantile plots, boxplots, and autocorrelation.

1.8 INFERENTIAL STATISTICS

Once we require answers to specific questions about the samples, we enter the realm of **inferential statistics**. The following are examples of questions we typically pose. Is a sample drawn from a **normal** distribution? Is it drawn from the same distribution as this other sample? Is there a trend? The question is often posed as a **hypothesis** that can be tested to be falsified. We must learn two important classes of methods: **parametric** (e.g., t and F tests) and **nonparametric** (e.g., Wilcoxon, Spearman). The first type is applied when the assumed distribution complies with certain conditions, and thus more conclusive, whereas the second type is less restrictive in assumptions but less conclusive.

Similarities between distributions are studied by goodness of fit (GOF) methods, which can be parametric (χ^2) and nonparametric (Kolmogorov–Smirnov) methods. Some simple and useful inferential methods are based on counts and proportions, and others such as contingency tables allow unveiling associations between categorical variables.

Providing sound **design of experiments** to test hypotheses has been an important mission of statistics in science and engineering. Analysis of variance (ANOVA) is a well-known method as its nonparametric counterpart (Kruskal–Wallis and Friedman). The mathematical formulation of ANOVA share basis with prediction, making their joint study helpful.

1.9 PREDICTORS, MODELS, AND REGRESSION

Prediction is at the core of building empirical models. We assume that there are drivers and effects. In other words, we want to build predictors of dependent variables Y based on the independent variables X. Regression techniques are the basis for many prediction methods. There are many types defined according to a variety of criteria and uses. The mathematical nature or structure of the predictor determines the type of method, such as linear regression vs. nonlinear regression; the number of variables determines the dimensionality, simple regression vs. multiple regression; the nature of the variables and their explicit variation with time and space determines specific methods, such as spatial autoregressive vs. autoregressive time series.

Varying with time and space is so pervasive in geographical and environmental systems that their study becomes essential even at an introductory level. Traditionally, the geography student is familiar with spatial analysis, and the engineering student with time series analysis. However, it is one of main tenets of the book that it is important to understand both spatial analysis and time series analysis.

1.10 TIME SERIES

A **random or stochastic process** is a collection of random variables with a distribution at each time *t*. When the distribution, or being less restrictive its moments, do not vary with time, we have the special case of **stationary** process that facilities analysis. A **time series** is a realization or sample of a random process. In Chapter 7, we cover independent random processes or processes such that the value at a given time is independent of past values. Then, after covering matrices in Chapter 9, we will study dependent random processes in Chapter 11. In these processes, the value at a given time is dependent of past values; we will study these focusing on **Markov** and **autoregressive** models.

Many methods in time series deal with finding out correlations between values at different times, and building predictors based on autoregressive and moving average models. Periodic process is amenable to spectral or periodical analysis, where we find the differences in the various or many periods or frequencies contained in the data.

1.11 SPATIAL DATA ANALYSIS

Throughout the book, we will look at two main types of spatial data: (1) **point patterns** and (2) **lattice** arrangements. A point pattern is a collection of points placed (often irregularly) over a spatial domain. We may have values of variables at each point. Lattice data correspond to values given to regions. These can be regularly arranged (as in a grid) or irregularly arranged (as in polygons). In Chapter 8, we will cover the analysis of point patterns, but we will wait until after Chapter 9 to cover lattice data because it requires matrix algebra. We will restrict ourselves to cover only the fundamentals of spatial data analysis since nowadays many of these methods are available in Geographic Information Systems (GIS) software.

A frequently encountered problem is examining the spatial distribution of points, for example, to check whether points are **clustered** or **uniformly** distributed, as well as its spatial variability. We can address this question by two alternative procedures. One procedure is dividing the spatial domain in **quadrats** and examining how many points fall in each quadrat and how it compares to the expected number of points. Another procedure is to calculate the distance to **nearest-neighbors**, examining the rhythm of change of this distance and comparing it to the one expected if it were to be uniform.

Once we consider that each point is **marked** or that there is a value of some variable associated to the point, we can also calculate the relationships between these values in the form of covariance. This leads to the concepts of variograms and semivariance that constitute the basis of **geostatistics**. Central to this collection of methods that emerged from engineering is the **kriging** method to predict values of variables in non-sampled points using a collection of sampled points.

Departing from point patterns are polygon patterns. **Spatial regression** helps predict values of variables in polygons from values at the neighboring polygons. Crucial to this method is defining the neighbor structure. Traditional methods include spatial autocorrelation (Moran and Geary) and several types of spatial regression models.

1.12 MATRICES AND MULTIPLE DIMENSIONS

Matrices are fundamental to understand the analysis of multiple variables and methods related to regression. All professionals in science and engineering benefit from understanding **matrix algebra**. It is important to know how to perform algebra operations with matrices, including the concepts of determinant and inverse. Formulating and solving linear equations using matrices, covariance matrices of sets of variables, and eigenvalues and eigenvectors, and singular value decomposition are crucial to all **multidimensional analysis** methods.

Many methods relate to inference such as multiple analysis of variance (MANOVA) and discriminant analysis. Others are related to prediction of a dependent variable based on a set of independent variables, as in multiple regression. We will cover extensions of some of the simple methods, particularly multiple linear regression, in order to build a model to predict Y from several independent variables X_i.

In addition, we will study methods to analyze a set of multiple variables to uncover relationships among all the variables and ways of reducing the dimensionality of the dataset. In some cases, we have basis to assume that we have several independent variables X_i, influencing several dependent or response variable Y_j. There are many multivariate analysis methods, but we will concentrate on those based on eigenvectors; such as principal components, factor analysis, correspondence analysis, multiple discriminant functions, canonical correlation, multidimensional scaling, and cluster analysis.

1.13 OTHER APPROACHES: PROCESS-BASED MODELS

A model is a simplified representation of reality, based on concepts, hypotheses, and theories of how the real system works. This book focuses on **empirical** models that build a quantitative relationship between variables based on data without an explicit consideration of the process yielding that relation. For example, using regression we can derive a predictor of tree height as a function of tree diameter based on measured data from 20 trees of different heights and diameters.

In contrast, some models are formulated by a set of mathematical **equations** based on the **processes** or **mechanisms** at work. For example, a differential equation representing tree growth over time based on increment of its diameter. For this purpose, we use the concept that diameter increases faster when the tree is smaller and that growth decreases when the tree is large. I have written a related book emphasizing **process-based** or **mechanistic** models, as opposed to **empirical** models (Acevedo, 2012). Simulation modeling is often the subject of a different course and taken at the senior undergraduate and first-year graduate level.

Both approaches complement each other. Empirical models help estimate parameters of the process-based models based on data from field and laboratory experiments. For example, we can use a mechanistic model to calculate flow of a stream using water velocity and cross-sectional area, but estimate velocity using an empirical relation of velocity to water depth. In addition, we will use empirical models to convert output variables of process-based models to other variables. For example, we can predict tree diameter increase from a process-based model of tree growth and then convert diameter to height using an empirical relation of height vs. diameter.

1.14 BABY STEPS: CALCULATIONS AND GRAPHS

1.14.1 Mean, Variance, and Standard Deviation of a Sample

In this chapter, we introduce three simple concepts: the mean, variance, and standard deviation of a set of values (Carr, 1995; Davis, 2002). Most likely, you know these concepts but let us review them for the sake of common terminology and notation. A **statistic** known as the **"sample mean"**, which is the arithmetic average of n data values x_i comprising a sample

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1.1}$$

The sample mean or average is denoted with a bar on top of X, i.e., $\overline{X}$.

Second, the **statistic** known as the **"sample variance"**, which is the variability measured relative to the arithmetic average of n data values x_i comprising a sample

$$s_X{}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^2 \tag{1.2}$$

This is the average of the square of the deviations from the sample mean. Alternatively

$$s_X{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2 \tag{1.3}$$

where $n - 1$ is used to account for the fact that the sample mean was already estimated from the n values. We can convert this equation to a more practical one by using Equation 1.1 in Equation 1.3 and doing algebra to obtain

$$s_X{}^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i{}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right] \tag{1.4}$$

This is easier to calculate because we can sum the squares of x_i and subtract the square of the sum of the x_i. Now, a third well-known concept is the standard deviation calculated as the square root of the variance

$$sd(X) = s_X = \sqrt{s_X{}^2}. \tag{1.5}$$

1.14.2 Simple Graphs as Text: Stem-and-Leaf Plots

The easiest visual display of data is a **"stem-and-leaf"** plot. It is a way of displaying a tally of the numbers and the shape of the distribution. In a stem-and-leaf plot, each data value is split into a "stem" and a "leaf." The "leaf" is usually the last digit of the number and the other digits to the left of the "leaf" form the "stem." For example, the number 25 would be split as stem = 2, leaf 5, shortened as 2|5.

First, list the data values in numerical ascending order. For example,

```
23, 25, 26, 30, 33, 33, 34, 35, 35, 37, 40, 40, 41
```

Then separate each number into stem|leaf. Since these are two-digit numbers, the tens digit is the stem and the units digit is the leaf. Group the numbers with the same stems in numerical order

```
2|356
3|0334557
4|001
```

A stem-and-leaf plot shows the shape and distribution of data. It can be clearly seen in the diagram that the data cluster around the row with a stem of 3 and has equal spread above and below 3.

1.14.3 Histograms

A histogram is a graphical display of the distribution of the data; it graphs the frequency with which you obtain a value in a sample (if discrete numbers) or values falling in intervals ("bins") of the range of the variable (if continuous). Given a large enough sample, a histogram can help to characterize the distribution of a variable (Carr, 1995; Davis, 2002).

1.15 EXERCISES

Exercise 1.1
Use a variable *X* to denote human population on Earth. Explain why it varies in time and space and give examples of a value at a particular location or region and time.

Exercise 1.2
Suppose you build a model of light transmission through a forest canopy using measured light (treated as dependent variable) at various heights (treated as independent variable) and use it to predict light at those heights where it is not measured. Would this be a process-based model or an empirical model?

Exercise 1.3
Extend Exercise 1.2 to use the concept from physics that light is attenuated as it goes through a medium. Propose that attenuation is proportional to the density of foliage at various heights, and then propose a model based on an equation before you collect data. Would this be a process-based model or an empirical model?

1.16 COMPUTER SESSION: INTRODUCTION TO R

1.16.1 Working Directory

The computer sessions of this book assume that you have access to write and read files in a **working directory** or **working folder** typically located in a local hard disk **c:** or a network home drive **h:** or in a removable drive ("flash drive") say **e:**. For the purpose of following the examples in this book, a convenient way to manage files is to create a working directory, for example, **c:\labs**, to store all the files to be used with this book. Then a folder or directory for each computer session will be created within **c:\labs** working directory. For example, for session 1, it would be the folder given by path **c:\labs\lab1**. In each folder, you will store your data files and programs for that session.

1.16.2 Installing R

Download R from the Comprehensive R Archive Network (CRAN) repository http://cran.us.r-project.org/ by looking for the **precompiled binary distribution** for your operating system (Linux, Mac, or Windows). In this book, we will assume a Windows installation. Thus, for Windows, select the **base** and then the executable download for the current release; for example at the time this chapter was last updated, the release was R-2.14.0. Save in your disk and run this program, following installation steps. It takes just a few minutes. During installation, it is convenient to choose the option to install manuals in PDF.

1.16.3 Personalize the R GUI Shortcut

You can establish your working directory as default by including it in the **Start In** option of the shortcut and avoid changing directory every time. It takes setup time but it will be worthwhile because it saves time in the end.

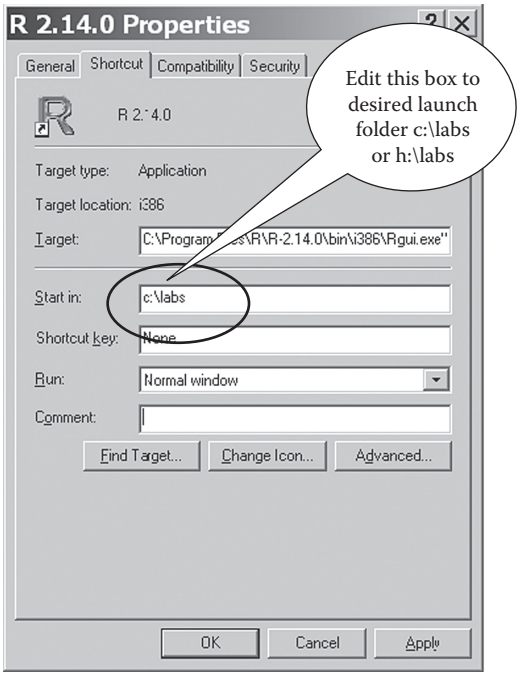


FIGURE 1.2 Modifying Start in folder of R shortcut properties.

To do this, Right click on shortcut, go to **Properties**, then type for example **c:\labs** or **h:\labs** as your launch or work directory (Figure 1.2). Double click to Run. R will start from this folder named labs. This remains valid unless edited. You need to create this folder beforehand if it does not exist yet. Note that now when you select Change dir, the working folder will be the one selected as Start In for the shortcut and therefore there is no need to reset.

When working on machines that are shared by many individuals (e.g., a university computer lab), a more permanent solution is to create a new personalized Shortcut in your working directory. Find the Shortcut on the desktop; right click, select **Create Shortcut**, and browse to the desired location (folder **labs**) as shown in Figure 1.3. Then right click on this new shortcut and select properties, edit **Start In** as shown before. Thereafter run R by a double click on this new personalized shortcut.

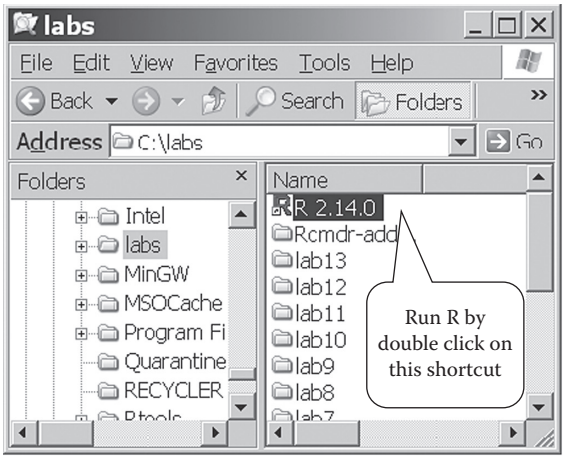


FIGURE 1.3 New R shortcut to reside in your working folder h:\labs.

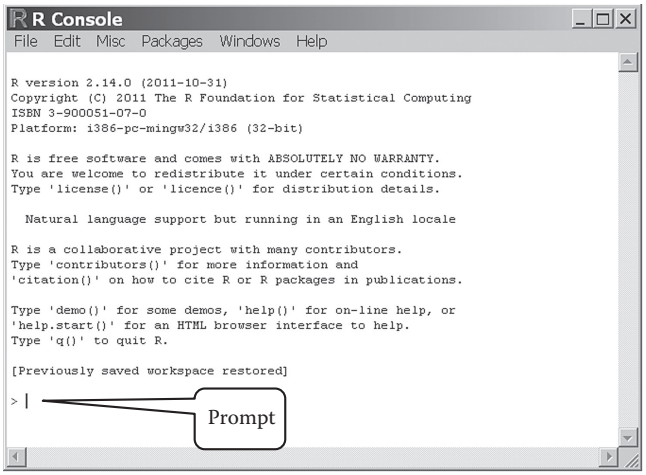


FIGURE 1.4 Start of R GUI and the R Console.

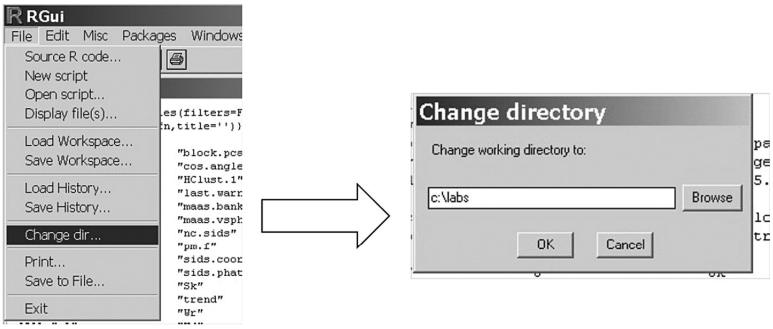


FIGURE 1.5 "Change dir" option in File menu to change working directory.

1.16.4 Running R

You can just double click on the shortcut created during installation. Once the R system loads we get the R Graphical User Interface (**Rgui)** that opens up with the ">" prompt on the **R Console** (Figure 1.4). Another practical rule: make sure you are in the desired directory or folder. Use the **File|Change Dir** option in the RGui menu, and type the path or browse to your working directory.

In other words, under the **File** menu you can find **Change dir**, to select the working directory (Figure 1.5). You may have to repeat this **Change dir** operation every time you start the R system. However, once you have a **workspace** file .**Rdata** created in a folder, you can double click it to launch the program from that folder. We will explain more on this later when we describe the workspace.

1.16.5 Basic R Skills

This session is a very brief tutorial guide to use **R** interactively from the GUI under windows. There is a lot more to **R** than summarized in these notes; see supplementary readings at the end of the chapter. These indications illustrate only how to get started, input data from files, simple graphics and programming loops, and are intended only as a starting point. We will study more details on R in later sessions.

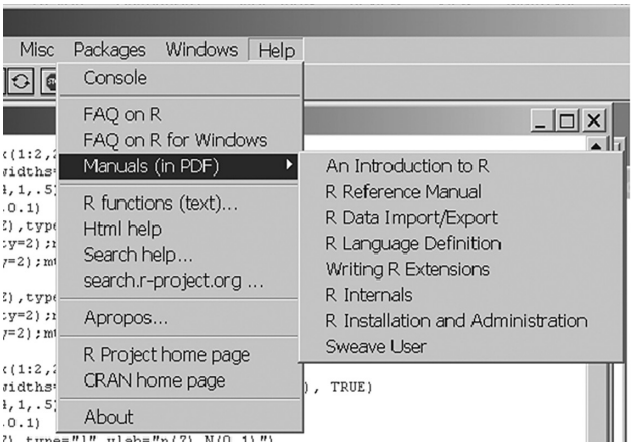


FIGURE 1.6 Finding R manuals from help menu item.

R Manuals are available online in PDF and in HTML formats via the help menu item. PDF (**P**ortable **D**ocument **F**iles) can be viewed and printed using the ***Acrobat Reader***. HTML (**H**yper**T**ext **M**arkup **L**anguage) can be viewed using a web browser. For example, *An introduction to R* in PDF (Figure 1.6). To obtain help in HTML format use the **Help** menu item, select **Html help** (Figure 1.7). This will run a browser with the help files and manuals. Just follow the links.

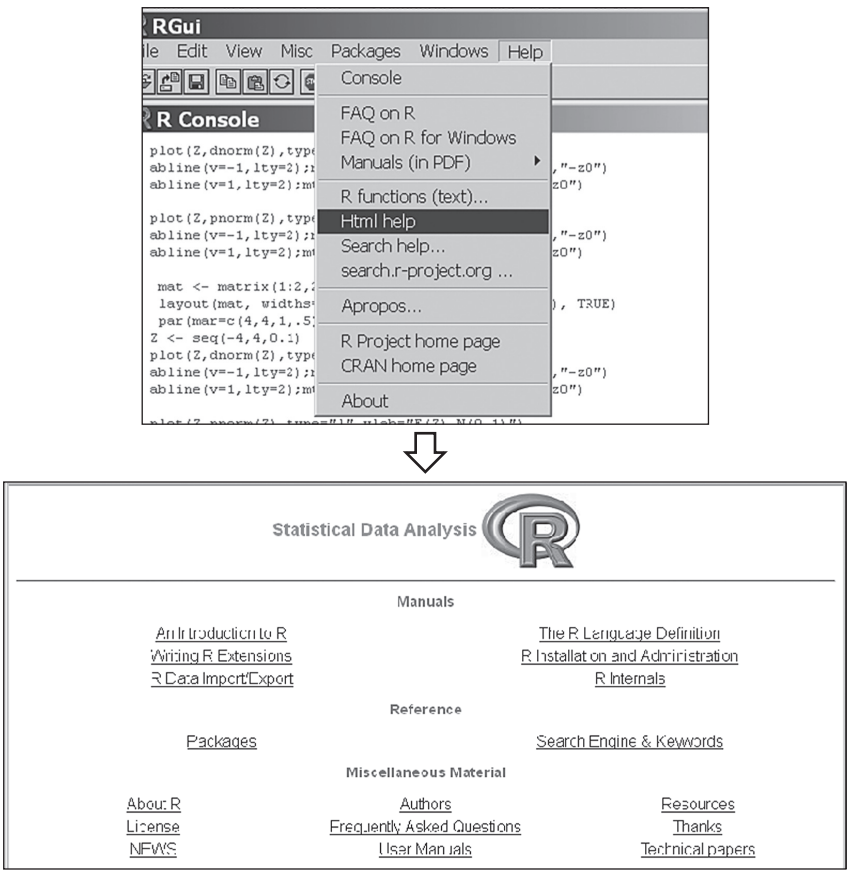


FIGURE 1.7 Help in HTML format and browser.

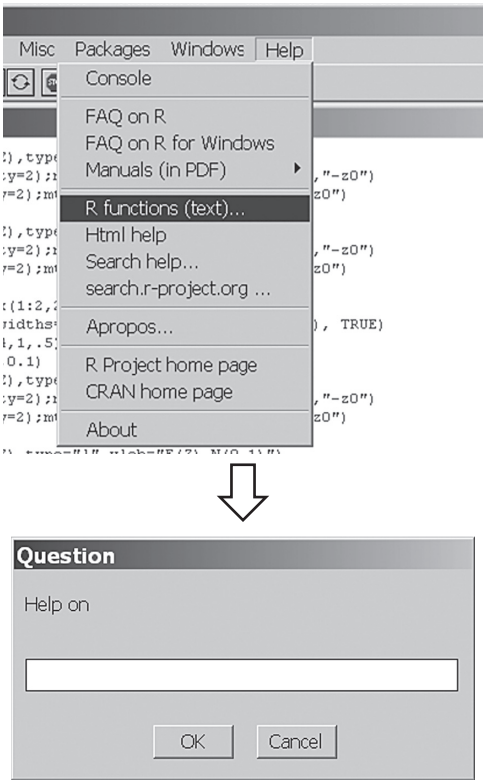


FIGURE 1.8 Help on specific functions.

You can obtain help on specific functions from Help menu, select R functions (text) as shown in Figure 1.8, then type the function name on the dialog box (Figure 1.8). From the R console, you can obtain help on specific functions. One way is to type **help(***name of function***)** at the prompt in the console; for example, **help(plot)** to get help on function **plot**. Also typing question mark followed by the function name would work, for example, **?plot**. You can also launch the help with **help.start** function. The following doc may help as simple reference to R at the CRAN website: http://cran.us.r-project.org/doc/contrib/Verzani-SimpleR.pdf.

1.16.6 R Console

This is where you type commands upon the > prompt and where you receive text output (Figure 1.4). We can also enter scripts or programs in a separate text editor, and then copy and paste to the R console for execution. Better you can use the **Script** facility of the Rgui described in Section 1.16.7.

1.16.7 Scripts

From the **File** menu, select **New Script** to type a new set of commands. This will generate an editor window where you can type your set of commands. In this window, you can type and edit commands, and then right click to run one line or a selection of lines by highlighting a section (Figure 1.9). Then save the script by using **File|Save as** followed by selecting a folder to save in, say lab1, and a name. Scripts are saved as files with name ***.R** that is with extension **".R"**. Later the script can be recalled by **File|Open Script**, browse to your folder, select the file, and edit the script. Further edits can be saved using **File|Save**.

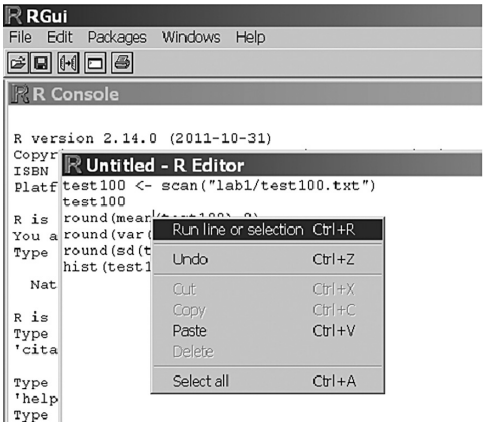


FIGURE 1.9 Script Editor: right click to run a line or lines of the script.

An alternative way of running your script (if you do not need to edit it before running it) is to use **File|Source R Code**, browse to folder, select the script. This is equivalent of using the function **source(“**myscript.R**”)** to execute the set of commands stored in file **myscript.R**.

1.16.8 Graphics Device

This is where you receive graphical output (Figure 1.10). To print hard copy of a graph, you can just select **File|Print** from menu while the graphics window is selected. To save in a variety of graphics formats use **File|Save as**. These formats include Metafile, Postscript, Bmp, PDF, and Jpeg. To capture on the clipboard and then paste on another application, you could simply use Copy and Paste from graphics window. To do this you can also use **File|Copy to clipboard** and then paste clipboard contents on selected cursor position of the file being edited in the application. Notice that one can work with windows as usual, go to **Windows** in the **menu** bar; here you can **Tile** the windows or can **Cascade** the windows, etc. In addition, this is where you can open the **Console** window.

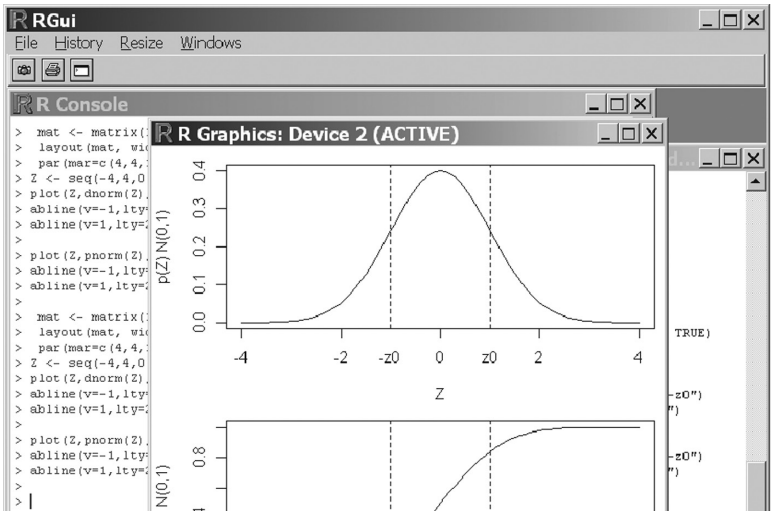


FIGURE 1.10 R graphics windows.

1.16.9 Downloading Data Files

Datasets and functions for the book are available as **archive** from the book website. After downloading, you can **unzip** and this will include several folders each containing several files. Each computer session corresponds to a folder. Download the archive **labs.zip** save to **labs** and unzip. This can be done under Windows using the Win Explorer and assuming you have a compression program available (e.g., WinZip and WinRAR). Then use extract button and select options in dialog box. Then folders **lab1, lab2, etc**., will be created within **labs**. We can examine the contents of each folder with the file explorer. It is convenient to configure the Explorer to show all the file extensions. Otherwise, you will not see the .**txt** part of the filename.

To keep files organized, I recommend that you store the files of each computer lab session in separate folders. To do this, you would use the subdirectory (e.g., **lab1**) in your working directory in your home drive **c:\labs**. In this directory, you will store your data files and functions (mostly ASCII text files), figures and results.

1.16.10 Read a Simple Text Data File

The first thing to do before you import a data file is to look at your data and understand the contents of the file. This is a practical rule, which applies regardless of what software you use. For example, let us look at file **test100.txt** in **lab1**. As we can see, this is a text file and it opens using the notepad. It does not have a header specifying the variable name and that it is just one field per line with no separator between fields (Figure 1.11). This is a straightforward way of entering a single variable.

A more convenient text editor is **Vim**, also an open-source program that you can download from the internet (www.vim.org). I recommend that use this editor instead of the notepad or wordpad to work with text files and scripts. Some nice features are that you get line numbers and position within a line (Figure 1.12), a more effective find/replace, and tool and color codes for your script.

Now, going back to file **lab1\test100.txt**, it contains 100 numbers starting like this

```
48
38
44
41
56
…
```

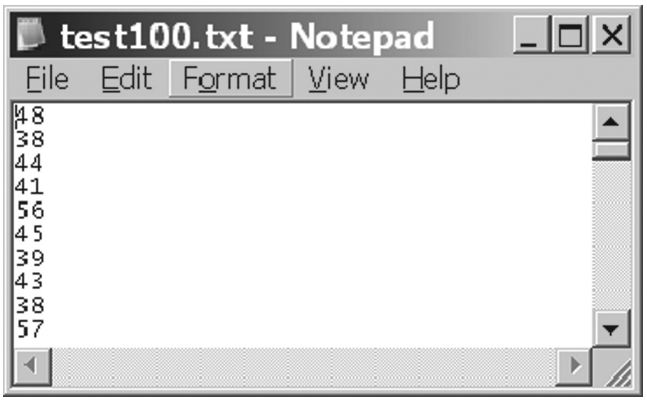


FIGURE 1.11 Example of a file with a single variable, one field per line. Viewed with the notepad text editor.

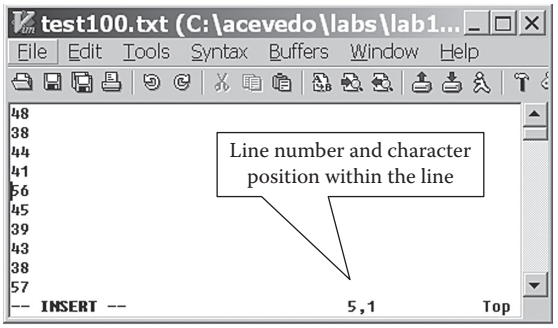


FIGURE 1.12 Same file as in previous figure viewed with the Vim editor.

Make sure you **File|Change Dir** to the working folder **labs** so that the path to the file is relative to this folder. If you have personalized the R shortcut to start in folder **labs**, then the path to the file is relative to this folder, for example, in this case, **lab1\ test100.txt**. Therefore, you could use this name to **scan** the file. Use forward slash "**/**" to separate folder and filename.

```
> scan("lab1/test100.txt")
```

On the console, we receive the response

```
Read 100 items
[1] 48 38 44 41 56 45 39 43 38 57 42 31 40 56 42 56 42 46 35 40 30
  49 36 28 55
[26] 29 40 53 49 45 32 35 38 38 26 38 26 49 45 30 40 38 38 36 45 41
  42 35 35 25
[51] 44 39 42 23 44 42 52 55 46 44 36 26 42 31 44 49 32 39 42 41 45
  50 39 55 48
[76] 49 26 50 46 56 31 54 26 29 32 34 40 53 37 27 45 37 34 32 33 35
  50 37 74 44
```

Alternatively, you could also use two backward slashes `> scan("lab1\\test100.txt")`. It has the same effect as one forward slash. We will use forward slashes for simplicity. Next, create an object by scanning an input data file

```
> x100 <- scan("lab1/test100.txt")
```

object `x100` is assigned the results scanned from the file. The operator "**<-**" is used for **assignment**. Equivalently you can write the same using the equal sign "=". However, the equal sign is used for other purposes, such as giving values to arguments of functions.

Double check that you have the newly created object by **Misc|List objects** or using `ls()`.

```
> ls()
[1] "x100"
```

The object `x100` is stored in the workspace **labs\.Rdata** but file **test100.txt** resides in **labs\lab1.** Double check the object contents by typing its name

```
> x100
[1] 48 38 44 41 56 45 39 43 38 57 42 31 40 56 42 56 42 46 35 40 30
  49 36 28 55
[26] 29 40 53 49 45 32 35 38 38 26 38 26 49 45 30 40 38 38 36 45 41
  42 35 35 25
[51] 44 39 42 23 44 42 52 55 46 44 36 26 42 31 44 49 32 39 42 41 45
  50 39 55 48
[76] 49 26 50 46 56 31 54 26 29 32 34 40 53 37 27 45 37 34 32 33 35
  50 37 74 44
```

We can see that this object is a one-dimensional array. The number given in brackets on the left-hand side is the position of the entry first listed in that row. For example, entry in position 26 is 29. Entry in position 51 is 44.

Since this object is a one-dimensional array, we can check the size of this object by using function `length()`

```
> length(x100)
[1] 100
```

Important tip: when entering commands at the console, you can recall previously typed commands using the up arrow key. For example, after you type

```
> x100
```

you can use the up arrow key and edit the line to add `length`

```
> length(x100)
```

1.16.11 Simple Statistics

Now we can calculate sample mean, variance, and standard deviation

```
> mean(x100)
[1] 40.86
> var(x100)
[1] 81.61657
> sd(x100)
[1] 9.034189
>
```

It is good practice to round the results, for example, to zero decimals

```
> round(mean(x100),0)
[1] 41
> round(var(x100),0)
[1] 82
> round(sd(x100),0)
[1] 9
>
```

We can concatenate commands in a single line by using the semicolon ";" character. Thus, for example, we can round the above to two decimals

```
> mean(x100); round(var(x100),2); round(sd(x100),2)
[1] 40.86
[1] 81.62
[1] 9.03
```

1.16.12 Simple Graphs as Text: Stem-and-Leaf Plots

We can do stem-and-leaf plots with a simple function `stem` on the Rconsole. For example,

```
> stem(x100)

The decimal point is 1 digit(s) to the right of the |

2 | 35666667899
3 | 00111222234455555666777888888889999
4 | 000001112222222234444445555556668899999
5 | 000233455566667
6 |
7 | 4
```

1.16.13 Simple Graphs to a Graphics Window

When applying a graphics command, a new graph window will open by default if none is opened; otherwise, the graph is sent to the active graph window. Plot a histogram by using function `hist` applied to a single variable or one-dimensional array object,

```
>hist(x100)
```

to obtain the graph shown in Figure 1.13. Bar heights are counts of how many measurements fall in the bin indicated in the horizontal axis.

1.16.14 Addressing Entries of an Array

We can refer to specific entries of an array using brackets or square braces. For example, entry in position 26 of array `x100` above

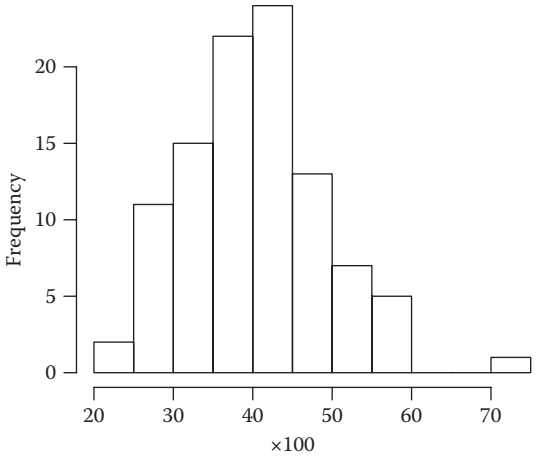


FIGURE 1.13 Histogram of x100.

```
> x100[26]
[1] 29
```

That is to say `x100[26] = 29`. A colon ":" is used to declare a sequence of entries. For example, the first 10 positions of `x100`

```
> x100[1:10]
[1] 48 38 44 41 56 45 39 43 38 57
```

Entries can be removed, for example, `x100[-1]` removes the first entry

```
> x100[-1]
[1] 38 44 41 56 45 39 43 38 57 42 31 40 56 42 56 42 46 35 40 30 49
  36 28 55 29
> length(x100[-1])
[1] 99
```

Using a blank or no character in the bracket means that all entries are used

```
> x100[]
[1] 48 38 44 41 56 45 39 43 38 57 42 31 40 56 42 56 42 46 35 40 30
  49 36 28 55
[26] 29 40 53 49 45 32 35 38 38 26 38 26 49 45 30 40 38 38 36 45 41
  42 35 35 25
[51] 44 39 42 23 44 42 52 55 46 44 36 26 42 31 44 49 32 39 42 41 45
  50 39 55 48
[76] 49 26 50 46 56 31 54 26 29 32 34 40 53 37 27 45 37 34 32 33 35
  50 37 74 44
> length(x100[])
[1] 100
```

1.16.15 Example: Salinity

Next, we work with an example of data collected in the field. Salinity is an environmental variable of great ecological and engineering importance. It conditions the type of plant and animals that can live in a body of water and impacts the quality of water and the potential use of saline water. At the interface between rivers and sea, such as estuaries, salinity experiences spatial and temporal gradients. It is traditional to express salinity in parts per thousand ‰ instead of percentage % because it is the same as approximately grams of salt per kilogram of solution. **Freshwater**'s salinity limit is 0.5‰, then water is considered **brackish** for the 0.5‰–30‰ range, above that we have **saline** water in the 30‰–50‰, and **brine** with more than 50‰.

Examine **lab1/salinity.txt** file. It consists of four lines of 10 numbers each. It corresponds to salinity of water from 40 measurements at Bayou Chico, a small estuary in Pensacola Bay, FL. The values are from the same location and taken every 15 min. The salinity.txt file is also a single variable but given in 10 values per line with blank separations (Figure 1.14). We will practice how to scan a file and plot histograms using this dataset. Create an object containing these data. What is the length of the object? Obtain a stem-and-leaf plot. Obtain a histogram. Save the graph as a Jpeg file.

```
> x <- scan("lab1/salinity.txt")
Read 40 items
> x
[1] 24.2 23.9 24.0 24.0 24.2 24.1 24.2 24.0 24.0 23.8 23.9 23.8 23.8
  23.8 23.8
[16] 23.7 23.6 23.5 23.3 23.2 23.3 23.2 23.1 23.1 23.1 23.2 23.0
  22.8 22.8 22.8
[31] 22.8 22.7 22.7 22.7 22.7 22.7 22.7 22.7 22.7 22.8
> length(x)
[1] 40
> stem(x)

The decimal point is 1 digit(s) to the left of the |

226 | 00000000
228 | 00000
230 | 0000
232 | 00000
234 | 0
236 | 00
238 | 0000000
240 | 00000
242 | 000

> hist(x)
> round(mean(x),1)
[1] 23.4
> round(var(x),1)
[1] 0.3
> round(sd(x),1)
[1] 0.5
>
>
```

To import the data we are using the `scan` command because, even though there are rows and columns in the data file, all of the numbers will be read as only **one** data stream; that is to say, a one-dimensional array. See the histogram in Figure 1.15. This water is brackish and does not get to be saline because it is under 30‰.

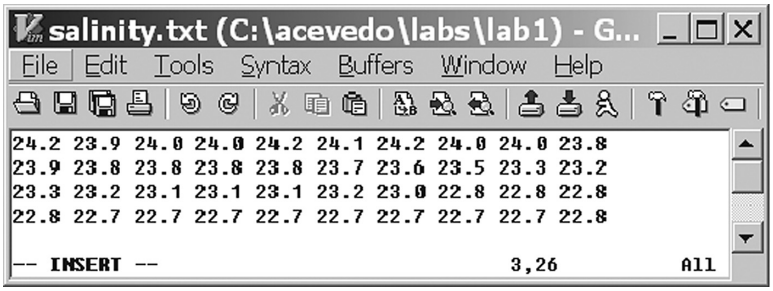


FIGURE 1.14 File with single variable, but several fields per line.

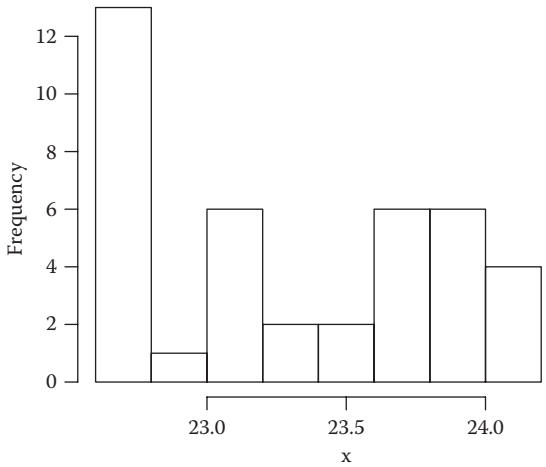


FIGURE 1.15 Histogram of salinity.

1.16.16 CSV Text Files

A CSV file is a text file with fields separated by commas. CSV stands for the format of **comma separated values**. In Windows, a default program to open the **CSV** files is Excel. Double click on the **salinity.csv** file to obtain Figure 1.16. To see more numbers you would have to scroll to the right. A CSV file is just a **text** file, and therefore it also opens with the notepad and Vim. Right click on the file name, select open with, and then select Vim. As you can see, commas separate the numbers and the lines are "word wrapped" (Figure 1.17). If using the notepad, you can choose the option **Format|Word wrap** to show all the numbers.

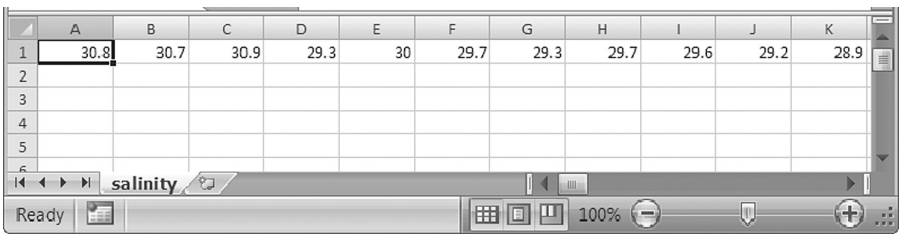


FIGURE 1.16 Example of a CSV file opened in MS Excel.

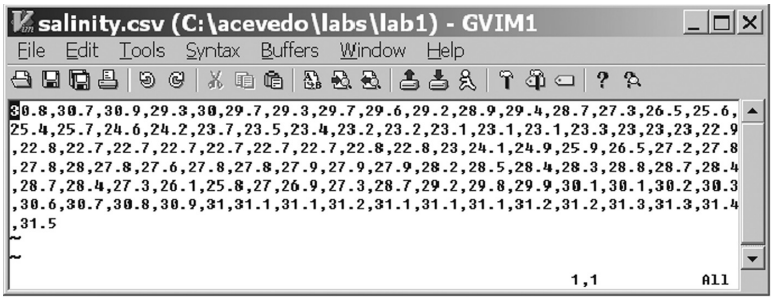


FIGURE 1.17 A CSV file opened in the notepad.

To read a CSV file into R, all you have to do is use scan with `sep= ","` argument.

```
> x<- scan("lab1/salinity.csv", sep=",")
Read 98 items
```

Then operate on object x as before

```
> length(x)
[1] 98
> stem(x)

The decimal point is at the |

22 | 7777778889
23 | 0000111223457
24 | 1269
25 | 46789
26 | 1559
27 | 02333688888999
28 | 0234445777789
29 | 2233467789
30 | 011236778899
31 | 0111112223345

> hist(x)
> round(mean(x),1)
[1] 27.4
> round(var(x),1)
[1] 9
> round(sd(x),1)
[1] 3
>
```

I am not showing the resulting histogram for the sake of saving space. These values indicate that this water is brackish on the average but close to saline. In fact 25 observations out of 98 were 30 or above.

1.16.17 Store Your Data Files and Objects

The workspace with "objects", functions, and results from object-making commands can be stored in a file .**Rdata**. Use **File|Save workspace** menu to store an image of your objects. File .**Rdata** is

created in the launch folder specified in the **Start In** field of the R shortcut or you can browse to find the desired working folder to store. For example, **c:/labs**, the console will inform of the save operation

```
> save.image("C:/labs/.RData")
```

To follow this book, it is convenient to store the workspace in your working folder. Once you save the workspace, right after opening the commands window, you will see the following message:

```
[Previously saved workspace restored]
```

When done this way, .**Rdata** resides in your working drive and you can use this .**Rdata** file for all computer sessions in order to facilitate access to functions created by various exercises. After launching the program, you could load the workspace using **File|Load Workspace** and browse to find the desired .**Rdata** file. Alternatively, you can also double click on the .**Rdata** file to launch the program and load the workspace.

You may want to have control of where the .**Rdata** file is stored and to store objects in different .**Rdata** files. You can use different files for storing objects. For example, we could use file name **other.Rdata** to save the workspace related to a different project. After launching the console window, you could load the workspace using **File|Load Workspace** and browse to find the desired .**Rdata** file. Again, for now I recommend that you use the same .**Rdata** folder in the launch folder for all lab sessions in order to facilitate access to functions you will create using the various computer exercises of this book.

To list objects in your workspace type `ls()` or using **Misc|List objects** of the RGUI menu. Check your objects with `ls()` or with `objects()`; if this is your first run, then you will not have existing objects and you will get `character(0)`.

```
> ls()
character(0)
> objects()
character(0)
>
```

1.16.18 Command History and Long Sequences of Commands

When editing through a long sequence of commands by using the arrow keys, one could use **History** to visualize all the previous commands at once. However, it is typically more convenient (especially if writing functions) to type the commands first in a script using the script editor or a text editor (say Vim). Then, Copy and Paste from the text editor to the R console to execute.

1.16.19 Editing Data in Objects

Example: want to edit object `x100`. Use

```
>edit(x100)
```

to invoke data editor or from **Edit** menu item GUI use **data editor**.

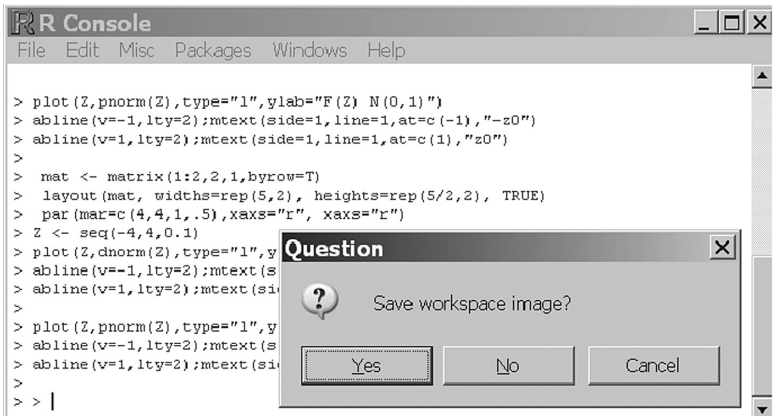


FIGURE 1.18 Closing the R session: reply yes.

1.16.20 Cleanup and Close R Session

Many times we generate objects that may not be needed after we use them. In this case, it is good practice to clean up after a session by removing objects using the `rm` function. A convenient way of doing this is to get a list of objects with `ls()` and then see what we need to remove.

For example, suppose at this point we may want to keep object `x100` because they contain data we may need later but remove object `x`.

```
> ls()
[1] "x100" .."x"
> rm(x)
```

You can also confirm that objects were indeed removed and get an update of the list to see if there is some more cleanup required.

```
> ls()
[1] "x100"
```

The objects can also be listed using **Misc|List Objects** in the Rgui menu.

Some of the clutter can be avoided by being careful about not generating objects unnecessarily. We will discuss how to do this later.

You can use `q()` or **File|Exit** to finalize R and close the session. When you do this, you will be prompted to save the workspace (Figure 1.18). Reply yes. This way you will have the objects created available for the next time you use R.

1.16.21 Computer Exercises

Exercise 1.4

To make sure you understand the workspace. Save your workspace **.Rdata** file. Then close R and start a new R session, Load the workspace, make sure you have the objects created before.

Exercise 1.5

Use the notepad or Vim to create a simple text file **myfile.txt**. Type 10 numbers in a row separated by a blank space, trying to type numbers around a value of 10. Save in folder **lab1**. Now read the

file using scan, calculate sample mean, variance, and standard deviation, plot a stem-and-leaf diagram and a histogram and discuss.

Exercise 1.6
Use file **lab1\exercise.csv**. Examine the file contents using the notepad or Vim. Read the file, list numbers on the R Console rounding to 2 decimals. Calculate sample mean, variance, and standard deviation, plot a stem-and-leaf diagram and a histogram and discuss.

Exercise 1.7
Separate the first 20 and last 20 elements of salinity × array into two objects. Plot a stem-and-leaf plot and a histogram for each.

SUPPLEMENTARY READING

Several textbooks cover similar and related topics and can be used for a tutorial and supplementary reading for geography and geosciences (Burt et al., 2009; Carr, 1995, 2002; Davis, 2002; Fotheringham et al., 2000; Jensen et al., 1997; Rogerson, 2001), ecology and environmental science (Gotelli and Ellison, 2004; Manly, 2009; Qian, 2010; Quinn and Keogh, 2002; Reimann et al., 2008; Zuur et al., 2009), and engineering (DeCoursey, 2003; Ledolter and Hogg, 2010; Petruccelli et al., 1999; Schiff and D'Agostino, 1996; Wadsworth, 1998).

In recent years, there has been an increase in books using R including empirical and mechanistic models (Bolker, 2008; Clark, 2007; Crawley, 2005; Dalgaard, 2008; Everitt and Hothorn, 2010; Jones et al., 2009; Manly, 2009; Qian, 2010; Reimann et al., 2008; Soetaert and Herman, 2009; Stevens, 2009; Zuur et al., 2009).

Software papers, manuals, and books are very useful to supplement the computer sessions (Chambers and Hastie, 1993; Clark, 2007; Venables et al. 2012; Crawley, 2002; Deutsch and Journel, 1992; Fox, 2005; Kaluzny et al., 1996; MathSoft, 1999; Middleton, 2000; Oksanen, 2011).

Several introductory texts serve to review basic concepts (Drake, 1967; Gonick and Smith, 1993; Griffith and Amrhein, 1991; Mann Prem, 1998; Sprinthall, 1990; Sullivan, 2004).

2 Probability Theory

2.1 EVENTS AND PROBABILITIES

Probability theory is the basis for the analysis of uncertainty in science and engineering. The concept of probability ties to a numerical measure of the likelihood of an **event**, which is one outcome of an experiment or measurement. The **sample space** is the set of all possible outcomes, and therefore an event is a subset of the sample space. Probability can thus be defined as a real number between zero and one (0 and 1 included) assigned to the likelihood of an event. As a shorthand for the probability of an event, we can write Pr[event] or P[event]. For example, for event A, $Pr[A]$ or $P[A]$, is a real number between 0 and 1 (0 and 1 included).

It is common to give examples of games of chance when illustrating probability. Consider rolling a six-sided die. The sample space has six possible outcomes, U = {side facing up is 1, side facing up is 2, …, side facing up is 6}. Note the use of curly brackets to define set of events. Define event A = {side facing up is number 3}, then $P[A] = 1/6$ or 1 out of 6 possible and equally likely outcomes.

2.2 ALGEBRA OF EVENTS

For didactic purposes, events are usually illustrated using Venn diagrams and set theory. Events are represented by shapes or areas located in a box or domain. The **universal** event is the sample space U (includes all possible events), and therefore occurs with absolute certainty $P[U] = 1$. For example, U = {any number 1 to 6 faces up after rolling a die}. See Figure 2.1.

The **null** event is an impossible event or one that includes none of the possible events. Therefore, its probability is zero, $P[\phi] = 0$. For example, ϕ = {the side with number 0 will face up}. This is not possible because the die does not have a side with number 0.

An oval shape represents an event A within U as shown in Figure 2.1. We also refer to B as the **complement** of A, i.e., the only other event that could occur. Therefore, the only outcomes are that A happens or B happens. Also, A and B are **mutually exclusive** and collectively exhaustive. The complement is an important concept often used to simplify solving problems. It is the same as B is NOT A, which in shorthand is $B = \bar{A}$ where the bar on top of the event means complement or logical operation NOT.

In the Venn diagram of Figure 2.1, B is shaded. The box represents U and the clear oval represents A. The key numeric relation is

$$P[B] = 1 - P[A] \tag{2.1}$$

Also, note that the complement of U is the null event.

Example from rolling a six-sided die, define B = {any side up except a six}, A = {side six faces up}, determine $P[B]$. Solution: first note that $B = \bar{A}$ and therefore we can use $P[B] = 1 - 1/6 = 5/6$. We did not have to enumerate B with detail, just subtracted from 1.

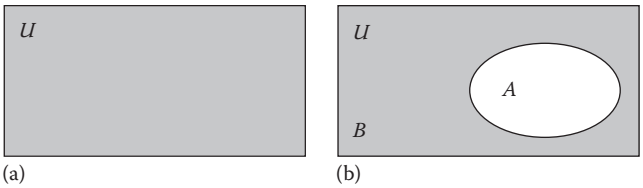


FIGURE 2.1 Universal event U or sample space (a), event A in U (b) showing B as the complement of A.

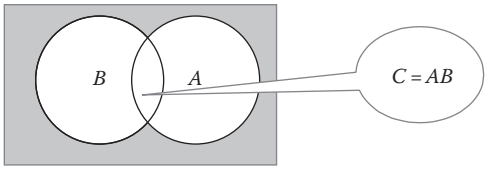


FIGURE 2.2 Intersection of two events; C is the sliver shared by events A and B.

When two events share common outcomes, we define the **intersection** of two events as the common or shared events. In other words, the intersection of A and B is the event C that is in both A and B. Denote the intersection by $C = AB$, then the probability of the intersection is $P[C] = P[AB]$. In the popular diagram illustrated in Figure 2.2, AB is contained in A and in B. It corresponds to the AND logical operation.

Back to the rolling die example. Define A = {side 1 faces up, side 2 faces up}, B = {side 2 faces up, side 3 faces up}. Obviously event {side 2 faces up} is common to A and B, therefore $C = AB$ = {side 2 faces up} and we know that this event has probability 1/6, thus $P[C] = 1/6$.

When A and B do not intersect, then AB is the null event $AB = \phi$ and therefore $P[AB] = 0$.

Example: A = {side 1 faces up, side 2 faces up}, B = {side 3 faces up, side 4 faces up}, C = {null} and this event has probability 0, thus $P[C] = 0$.

The **union** of A and B is the event C defined as A happens or B happens. It is the OR logical operation and is denoted by $A + B$. In reference to Figure 2.2, it would be the addition of the two circles but we have to avoid double counting the sliver of the intersection. Therefore, we discount the intersection AB once.

$$P[C] = P[A + B] = P[A] + P[B] - P[AB] \tag{2.2}$$

Example, A = {side 1 faces up, side 2 faces up}, B = {side 2 faces up, side 3 faces up}, then C = {side 1 faces up, side 2 faces up, side 3 faces up} and AB = {side 2 faces up}. Assigning probabilities, $P[A] = 2/6$, $P[B] = 2/6$, $P[AB] = 1/6$, $P[A + B] = 2/6 + 2/6 - 1/6 = 3/6$.

An event B is included in A when event B is a subset of A, in set notation $B \subset A$ and therefore $P[B] < P[A]$. See Figure 2.3.

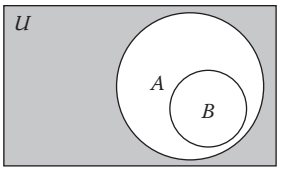


FIGURE 2.3 Event B is included in event A.

2.3 COMBINATIONS

When we complicate the experiment, for example, tossing a coin three times in a sequence, rolling a die five times in a sequence, we can combine the probabilities from the simpler components of the experiments to obtain probabilities of the more complex outcomes. The number n of independent repetitions (trials) and the number k of outcomes of each repetition determine the total number of possible outcomes. In general,

$$N = k^n \tag{2.3}$$

For example, consider tossing a coin twice and denote H for head and T for tail. We have two outcomes $k = 2$ for each trial and $n = 2$ trials. The outcome of a toss is independent of the other. Possible combinations lead to four events $N = 2^2$ and this constitutes the sample space $U = \{HH, HT, TH, TT\}$. Each outcome is equally likely with probability ¼. To see this we reason that the probability of getting H in first toss, $P[H] = 1/2$, and to get H in the second toss is the same because of independence, therefore $P[HH] = 1/4$.

The combinations of n items taken r at a time are of great interest

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{2.4}$$

The exclamation "!" symbol is a factorial operation defined as

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1 \tag{2.5}$$

For example, how many events have exactly one tail in two tosses of a coin? What is the probability of obtaining event $A = \{\text{exactly one tail in two coin tosses}\}$? Using Equation 2.4 yields

$$\binom{2}{1} = \frac{2!}{1!(2-1)!} = \frac{1 \times 2}{1 \times 1} = 2.$$

Thus, there are two possible combinations of one head in two tosses. This makes sense because from the previous example we know that we have four possible events. Only a set of these would have one tail in two trials and we can count them HT, TH. We can calculate the probability as $P[A] = P[HT] + P[TH] = 0.25 + 0.25 = 0.5 = 1/2$.