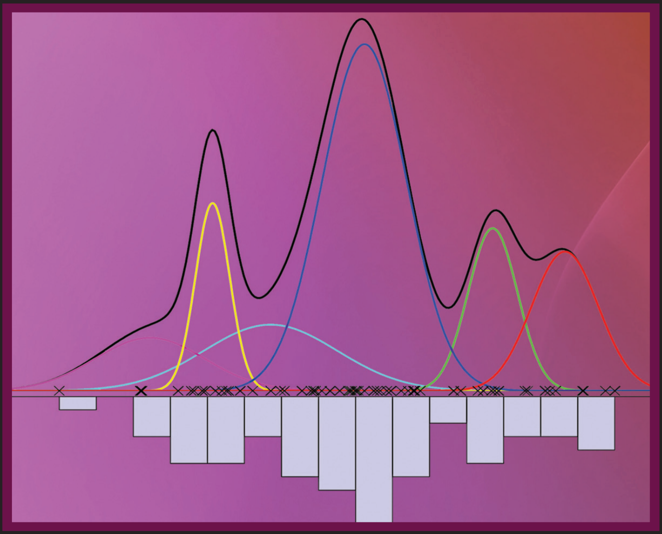
Computer Science and Data Analysis Series

Computational Statistics Handbook with MATLAB®

Third Edition



Wendy L. Martinez
Angel R. Martinez

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Computational Statistics Handbook with MATLAB®

Third Edition

Chapman & Hall/CRC

Computer Science and Data Analysis Series

The interface between the computer and statistical sciences is increasing, as each discipline seeks to harness the power and resources of the other. This series aims to foster the integration between the computer sciences and statistical, numerical, and probabilistic methods by publishing a broad range of reference works, textbooks, and handbooks.

Proposals for the series should be sent directly to one of the series editors above, or submitted to:

Chapman & Hall/CRC
Taylor and Francis Group
3 Park Square, Milton Park
Abingdon, OX14 4RN, UK

Published Titles

Semisupervised Learning for Computational Linguistics
Steven Abney

Visualization and Verbalization of Data
Jörg Blasius and Michael Greenacre

Design and Modeling for Computer Experiments
Kai-Tai Fang, Runze Li, and Agus Sudjianto

Microarray Image Analysis: An Algorithmic Approach
Karl Fraser, Zidong Wang, and Xiaohui Liu

R Programming for Bioinformatics
Robert Gentleman

Exploratory Multivariate Analysis by Example Using R
François Husson, Sébastien Lê, and Jérôme Pagès

Bayesian Artificial Intelligence, Second Edition
Kevin B. Korb and Ann E. Nicholson

Published Titles cont.

Computational Statistics Handbook with MATLAB®, Third Edition
Wendy L. Martinez and Angel R. Martinez

Exploratory Data Analysis with MATLAB®, Second Edition
Wendy L. Martinez, Angel R. Martinez, and Jeffrey L. Solka

Statistics in MATLAB®: A Primer
Wendy L. Martinez and MoonJung Cho

Clustering for Data Mining: A Data Recovery Approach, Second Edition
Boris Mirkin

Introduction to Machine Learning and Bioinformatics
Sushmita Mitra, Sujay Datta, Theodore Perkins, and George Michailidis

Introduction to Data Technologies
Paul Murrell

R Graphics
Paul Murrell

Correspondence Analysis and Data Coding with Java and R
Fionn Murtagh

Pattern Recognition Algorithms for Data Mining
Sankar K. Pal and Pabitra Mitra

Statistical Computing with R
Maria L. Rizzo

Statistical Learning and Data Science
Mireille Gettler Summa, Léon Bottou, Bernard Goldfarb, Fionn Murtagh,
Catherine Pardoux, and Myriam Touati

Foundations of Statistical Algorithms: With References to R Packages
Claus Weihs, Olaf Mersmann, and Uwe Ligges

This page intentionally left blank

Computer Science and Data Analysis Series

Computational Statistics Handbook with MATLAB®

Third Edition

Wendy L. Martinez

Angel R. Martinez

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

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

Version Date: 20150904

International Standard Book Number-13: 978-1-4665-9274-2 (eBook - PDF)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To

Edward J. Wegman

Teacher, Mentor, and Friend

This page intentionally left blank

Table of Contents

Chapter 1
Introduction

Chapter 2
Probability Concepts

Chapter 3
Sampling Concepts

Chapter 4
Generating Random Variables

Chapter 5
Exploratory Data Analysis

Chapter 6
Finding Structure

Chapter 7
Monte Carlo Methods for Inferential Statistics

Chapter 8
Data Partitioning

Chapter 9
Probability Density Estimation

Chapter 10
Supervised Learning

Chapter 11
Unsupervised Learning

Chapter 12
Parametric Models

Chapter 13
Nonparametric Models

Chapter 14
Markov Chain Monte Carlo Methods

Appendix A
MATLAB® Basics

Appendix B
Projection Pursuit Indexes

Appendix C
Data Sets

Appendix D
Notation

This page intentionally left blank

Preface to the Third Edition

It has been almost ten years since the second edition of the *Computational Statistics Handbook with MATLAB®* was published, and MATLAB has evolved greatly in that time. There were also some relevant topics in computational statistics that we always wanted to include, such as support vector machines and multivariate adaptive regression splines. So, we felt it was time for a new edition.

To use all of the functions and code described in this book, one needs to have the MATLAB Statistics and Machine Learning Toolbox®, which is a product of The MathWorks, Inc. The name of this toolbox was changed to the longer title to reflect the connection with machine learning approaches, like supervised and unsupervised learning (Chapters 10 and 11). We will keep to the shorter name of "MATLAB Statistics Toolbox" in this text for readability.

We list below some of the major changes in the third edition.

- Chapter 10 has additional sections on support vector machines and nearest neighbor classifiers. We also added a brief description of naive Bayes classifiers.
- Chapter 12 was updated to include sections on stepwise regression, least absolute shrinkage and selection operator (lasso), ridge regression, elastic net, and partial least squares regression.
- Chapter 13 now has a section on multivariate adaptive regression splines.
- Spatial statistics is an area that uses many of the techniques covered in the text, but it is not considered part of computational statistics. So, we removed Chapter 15.
- The introduction to MATLAB given in Appendix A has been expanded and updated to reflect the new desktop environment, object-oriented programming, and more.
- The text has been updated for MATLAB R2015a and the corresponding version of the Statistics and Machine Learning Toolbox.

We retained the same philosophy and writing style used in the previous editions of the book. The theory is kept to a minimum and is included where it offers some insights to the data analyst. All MATLAB code, example files, and data sets are available for download at the CRC website for the book:

`http://www.crcpress.com/product/ISBN/9781466592735`

The latest version of the Computational Statistics Toolbox can be found at the CRC website and above at the following link:

`http://www.pi-sigma.info/`

We would like to acknowledge the invaluable help of the reviewers for the previous editions. Reviewers for this edition include Tom Lane, Terrance Savitsky, and Gints Jekabsons. We thank them for their insightful comments. We are especially indebted to Tom Lane (from The MathWorks, Inc.) for his vision and leadership, which we think were instrumental in making MATLAB a leading computing environment for data analysis and statistics. Finally, we are grateful for our editors at CRC Press (David Grubbs and Michele Dimont) and the MATLAB book program at The MathWorks, Inc.

Disclaimers

1. Any MATLAB programs and data sets that are included with the book are provided in good faith. The authors, publishers, or distributors do not guarantee their accuracy and are not responsible for the consequences of their use.
2. Some of the MATLAB functions provided with the Computational Statistics Toolbox were written by other researchers, and they retain the copyright. References are given in the **`help`** section of each function. Unless otherwise specified, the Computational Statistics Toolbox is provided under the GNU license specifications:

   `http://www.gnu.org/copyleft/gpl.html`
3. The views expressed in this book are those of the authors and do not necessarily represent the views of the United States government.

Wendy L. and Angel R. Martinez
November 2015

Preface to the Second Edition

We wrote a second edition of this book for two reasons. First, the Statistics Toolbox for MATLAB® has been significantly expanded since the first edition, so the text needed to be updated to reflect these changes. Second, we wanted to incorporate some of the suggested improvements that were made by several of the reviewers of the first edition. In our view, one of the most important issues that needed to be addressed is that several topics that should be included in a text on computational statistics were missing, so we added them to this edition.

We list below some of the major changes and additions of the second edition.

- Chapters 2 and 4 have been updated to include new functions for the multivariate normal and multivariate *t* distributions, which are now available in the Statistics Toolbox.
- Chapter 5 of the first edition was split into two chapters and updated with new material. Chapter 5 is still on exploratory data analysis, but it now has updated information on new MATLAB functionality for univariate and bivariate histograms, glyphs, parallel coordinate plots, and more.
- Topics that pertain to touring the data and finding structure can be found in Chapter 6. This includes new content on independent component analysis, nonlinear dimensionality reduction, and multidimensional scaling.
- Chapter 9 of the first edition was divided into two chapters (now Chapters 10 and 11), and new content was added. Chapter 10 of the new edition includes methods pertaining to supervised learning with new topics on linear classifiers, quadratic classifiers, and voting methods (e.g., bagging, boosting, and random forests).
- Methods for unsupervised learning or clustering have been moved to Chapter 11. This content has been expanded and updated to include model-based clustering and techniques for assessing the results of clustering.
- Chapter 12 on parametric models has been added. This has descriptions of spline regression models, logistic regression, and generalized linear models.

- Chapter 13 on nonparametric regression has been expanded. It now includes information on more smoothers, such as bin smoothing, running mean and line smoothers, and smoothing splines. We also describe additive models for nonparametric modeling when one has many predictors.
- The text has been updated for MATLAB R2007a and the Statistics Toolbox, V6.0.

We tried to keep to the same philosophy and style of writing that we had in the first book. The theory is kept to a minimum, and we provide references at the end of the text for those who want a more in-depth treatment. All MATLAB code, example files, and data sets are available for download at the CRC website and StatLib:

http://lib.stat.cmu.edu
http://www.crcpress.com/e_products/downloads/

We also have a website for the text, where up-to-date information will be posted. This includes the latest version of the Computational Statistics Toolbox, code fixes, links to useful websites, and more. The website can be found at

http://www.pi-sigma.info/

The first edition of the book was written using an older version of MATLAB. Most of the code in this text should work with earlier versions, but we have updated the text to include new functionality from the Statistics Toolbox, Version 6.0.

We would like to acknowledge the invaluable help of the reviewers for the second edition: Tom Lane, David Marchette, Carey Priebe, Jeffrey Solka, Barry Sherlock, Myron Katzoff, Pang Du, and Yuejiao Ma. Their many helpful comments made this book a much better product. Any shortcomings are the sole responsibility of the authors. We greatly appreciate the help and patience of those at CRC Press: Bob Stern, Jessica Vakili, Russ Heap, and James Yanchak. We are grateful to Hsuan-Tien Lin for letting us use his boosting code and Jonas Lundgren for his spline regression function. Finally, we are especially indebted to Tom Lane and those members of the MATLAB book program at The MathWorks, Inc. for their special assistance with MATLAB.

Disclaimers

1. Any MATLAB programs and data sets that are included with the book are provided in good faith. The authors, publishers, or distributors do not guarantee their accuracy and are not responsible for the consequences of their use.

2. Some of the MATLAB functions provided with the Computational Statistics Toolbox were written by other researchers, and they retain the copyright. References are given in the **`help`** section of each function. Unless otherwise specified, the Computational Statistics Toolbox is provided under the GNU license specifications:

   `http://www.gnu.org/copyleft/gpl.html`

3. The views expressed in this book are those of the authors and do not necessarily represent the views of the United States Department of Defense or its components.

We hope that readers will find this book and the accompanying code useful in their educational and professional endeavors.

Wendy L. and Angel R. Martinez
November 2007

This page intentionally left blank

Preface to the First Edition

Computational statistics is a fascinating and relatively new field within statistics. While much of classical statistics relies on parameterized functions and related assumptions, the computational statistics approach is to let the data tell the story. The advent of computers with their number-crunching capability, as well as their power to show on the screen two- and three-dimensional structures, has made computational statistics available for any data analyst to use.

Computational statistics has a lot to offer the researcher faced with a file full of numbers. The methods of computational statistics can provide assistance ranging from preliminary exploratory data analysis to sophisticated probability density estimation techniques, Monte Carlo methods, and powerful multi-dimensional visualization. All of this power and novel ways of looking at data are accessible to researchers in their daily data analysis tasks. One purpose of this book is to facilitate the exploration of these methods and approaches and to provide the tools to make of this, not just a theoretical exploration, but a practical one. The two main goals of this book are

- To make computational statistics techniques available to a wide range of users, including engineers and scientists, and
- To promote the use of MATLAB® by statisticians and other data analysts.

We note that MATLAB and Handle Graphics® are registered trademarks of The MathWorks, Inc.

There are wonderful books that cover many of the techniques in computational statistics and, in the course of this book, references will be made to many of them. However, there are very few books that have endeavored to forgo the theoretical underpinnings to present the methods and techniques in a manner immediately usable to the practitioner. The approach we take in this book is to make computational statistics accessible to a wide range of users and to provide an understanding of statistics from a computational point of view via methods applied to real applications.

This book is intended for researchers in engineering, statistics, psychology, biostatistics, data mining, and any other discipline that must deal with the analysis of raw data. Students at the senior undergraduate level or beginning graduate level in statistics or engineering can use the book to supplement course material. Exercises are included with each chapter, making it suitable as a textbook for a course in computational statistics and data analysis. Scientists who would like to know more about programming methods for analyzing data in MATLAB should also find it useful.

We assume that the reader has the following background:

- Calculus: Since this book is computational in nature, the reader needs only a rudimentary knowledge of calculus. Knowing the definition of a derivative and an integral is all that is required.
- Linear Algebra: Since MATLAB is an array-based computing language, we cast several of the algorithms in terms of matrix algebra. The reader should have a familiarity with the notation of linear algebra, array multiplication, inverses, determinants, an array transpose, etc.
- Probability and Statistics: We assume that the reader has had introductory probability and statistics courses. However, we provide a brief overview of the relevant topics for those who might need a refresher.

We list below some of the major features of the book.

- The focus is on implementation rather than theory, helping the reader understand the concepts without being burdened by the theory.
- References that explain the theory are provided at the end of each chapter. Thus, those readers who need the theoretical underpinnings will know where to find the information.
- Detailed step-by-step algorithms are provided to facilitate implementation in any computer programming language or appropriate software. This makes the book appropriate for computer users who do not know MATLAB.
- MATLAB code in the form of a Computational Statistics Toolbox is provided. These functions are available for download.
- Exercises are given at the end of each chapter. The reader is encouraged to go through these because concepts are sometimes explored further in them. Exercises are computational in nature, which is in keeping with the philosophy of the book.
- Many data sets are included with the book, so the reader can apply the methods to real problems and verify the results shown in the book. The data are provided in MATLAB binary files (**.mat**) as well as text, for those who want to use them with other software.
- Typing in all of the commands in the examples can be frustrating. So, MATLAB scripts containing the commands used in the examples are also available for download.

- A brief introduction to MATLAB is provided in Appendix A. Most of the constructs and syntax that are needed to understand the programming contained in the book are explained.
- Where appropriate, we provide references to Internet resources for computer code implementing the algorithms described in the chapter. These include code for MATLAB, S-plus, Fortran, etc.

We would like to acknowledge the invaluable help of the reviewers: Noel Cressie, James Gentle, Thomas Holland, Tom Lane, David Marchette, Christian Posse, Carey Priebe, Adrian Raftery, David Scott, Jeffrey Solka, and Clifton Sutton. Their many helpful comments made this book a much better product. Any shortcomings are the sole responsibility of the authors. We owe a special thanks to Jeffrey Solka for some programming assistance with finite mixtures. We greatly appreciate the help and patience of those at CRC Press: Bob Stern, Joanne Blake, and Evelyn Meany. We also thank Harris Quesnell and James Yanchak for their help with resolving font problems. Finally, we are indebted to Naomi Fernandes and Tom Lane at The MathWorks, Inc. for their special assistance with MATLAB.

Disclaimers

1. Any MATLAB programs and data sets that are included with the book are provided in good faith. The authors, publishers, or distributors do not guarantee their accuracy and are not responsible for the consequences of their use.
2. The views expressed in this book are those of the authors and do not necessarily represent the views of the Department of Defense or its components.

Wendy L. and Angel R. Martinez
August 2001

This page intentionally left blank

Chapter 1

Introduction

1.1 What Is Computational Statistics?

Obviously, computational statistics relates to the traditional discipline of statistics. So, before we define computational statistics proper, we need to get a handle on what we mean by the field of statistics. At a most basic level, statistics is concerned with the transformation of raw data into knowledge [Wegman, 1988].

When faced with an application requiring the analysis of raw data, any scientist must address questions such as:

- What data should be collected to answer the questions in the analysis?
- How many data points should we obtain?
- What conclusions can be drawn from the data?
- How far can those conclusions be trusted?

Statistics is concerned with the science of uncertainty and can help the scientist deal with these questions. Many classical methods (regression, hypothesis testing, parameter estimation, confidence intervals, etc.) of statistics developed over the last century are familiar to scientists and are widely used in many disciplines [Efron and Tibshirani, 1991].

Now, what do we mean by computational statistics? Here we again follow the definition given in Wegman [1988]. Wegman defines ***computational statistics*** as a collection of techniques that have a strong "focus on the exploitation of computing in the creation of new statistical methodology."

Many of these methodologies became feasible after the development of inexpensive computing hardware since the 1980s. This computing revolution has enabled scientists and engineers to store and process massive amounts of data. However, these data are typically collected without a clear idea of what they will be used for in a study. For instance, in the practice of data analysis today, we often collect the data and then we design a study to gain some useful information from them. In contrast, the traditional approach has been to first design the study based on research questions and then collect the required data.

Because the storage and collection is so cheap, the data sets that analysts must deal with today tend to be very large and high-dimensional. It is in situations like these where many of the classical methods in statistics are inadequate. As examples of computational statistics methods, Wegman [1988] includes parallel coordinates for visualizing high dimensional data, nonparametric functional inference, and data set mapping, where the analysis techniques are considered fixed.

Efron and Tibshirani [1991] refer to what we call computational statistics as *computer-intensive statistical methods*. They give the following as examples for these types of techniques: bootstrap methods, nonparametric regression, generalized additive models, and classification and regression trees. They note that these methods differ from the classical methods in statistics because they substitute computer algorithms for the more traditional mathematical method of obtaining an answer. An important aspect of computational statistics is that the methods free the analyst from choosing methods mainly because of their mathematical tractability.

Volume 9 of the *Handbook of Statistics*: *Computational Statistics* [Rao, 1993] covers topics that illustrate the "... trend in modern statistics of basic methodology supported by the state-of-the-art computational and graphical facilities...." It includes chapters on computing, density estimation, Gibbs sampling, the bootstrap, the jackknife, nonparametric function estimation, statistical visualization, and others.

Gentle [2005] also follows the definition of Wegman [1988] where he states that computational statistics is a discipline that includes a "... class of statistical methods characterized by computational intensity...". His book includes Monte Carlo methods for inference, cross-validation and jackknife methods, data transformations to find structure, visualization, probability density estimation, and pattern recognition.

We mention the topics that can be considered part of computational statistics to help the reader understand the difference between these and the more traditional methods of statistics. Table 1.1 [Wegman, 1988] gives an excellent comparison of the two areas.

TABLE 1.1

Comparison Between Traditional Statistics and Computational Statistics [Wegman, 1988]

Traditional Statistics	Computational Statistics
Small to moderate sample size	Large to very large sample size
Independent, identically distributed data sets	Nonhomogeneous data sets
One or low dimensional	High dimensional
Manually computational	Computationally intensive
Mathematically tractable	Numerically tractable
Well focused questions	Imprecise questions
Strong unverifiable assumptions: Relationships (linearity, additivity) Error structures (normality)	Weak or no assumptions: Relationships (nonlinearity) Error structures (distribution free)
Statistical inference	Structural inference
Predominantly closed form algorithms	Iterative algorithms possible
Statistical optimality	Statistical robustness

Reprinted with permission from the *Journal of the Washington Academy of Sciences*

1.2 An Overview of the Book

Philosophy

The focus of this book is on methods of computational statistics and how to implement them. We leave out much of the theory, so the reader can concentrate on how the techniques may be applied. In many texts and journal articles, the theory obscures implementation issues, contributing to a loss of interest on the part of those needing to apply the theory. The reader should not misunderstand, though; the methods presented in this book are built on solid mathematical foundations. Therefore, at the end of each chapter, we include a section containing references that explain the theoretical concepts associated with the methods covered in that chapter.

What Is Covered

In this book, we cover some of the most commonly used techniques in computational statistics. While we cannot include all methods that might be a part of computational statistics, we try to present those that have been in use for several years.

Since the focus of this book is on the implementation of the methods, we include step-by-step descriptions of the procedures. We also provide examples that illustrate the use of the methods in data analysis. It is our hope that seeing how the techniques are implemented will help the reader understand the concepts and facilitate their use in data analysis.

Some background information is given in Chapters 2, 3, and 4 for those who might need a refresher in probability and statistics. In Chapter 2, we discuss some of the general concepts of probability theory, focusing on how they will be used in later chapters of the book. Chapter 3 covers some of the basic ideas of statistics and sampling distributions. Since many of the approaches in computational statistics are concerned with estimating distributions via simulation, this chapter is fundamental to the rest of the book. For the same reason, we present some techniques for generating random variables in Chapter 4.

Some of the methods in computational statistics enable the researcher to explore the data before other analyses are performed. These techniques are especially important with high dimensional data sets or when the questions to be answered using the data are not well focused. In Chapters 5 and 6, we present some graphical exploratory data analysis techniques that could fall into the category of traditional statistics (e.g., box plots, scatterplots). We include them in this text so statisticians can see how to implement them in MATLAB® and to educate scientists and engineers as to their usage in exploratory data analysis. Other graphical methods in this book *do* fall into the category of computational statistics. Among these are isosurfaces, parallel coordinates, the grand tour, and projection pursuit.

In Chapters 7 and 8, we present methods that come under the general heading of resampling. We first cover some of the main concepts in hypothesis testing and confidence intervals to help the reader better understand what follows. We then provide procedures for hypothesis testing using simulation, including a discussion on evaluating the performance of hypothesis tests. This is followed by the bootstrap method, where the data set is used as an estimate of the population and subsequent sampling is done from the sample. We show how to get bootstrap estimates of standard error, bias, and confidence intervals. Chapter 8 continues with two closely related methods called the jackknife and cross-validation.

One of the important applications of computational statistics is the estimation of probability density functions. Chapter 9 covers this topic, with an emphasis on the nonparametric approach. We show how to obtain estimates using probability density histograms, frequency polygons, averaged shifted histograms, kernel density estimates, finite mixtures, and adaptive mixtures.

Chapters 10 and 11 describe statistical pattern recognition methods for supervised and unsupervised learning. For supervised learning, we discuss Bayes decision theory, classification trees, and ensemble classifier methods. We present several unsupervised learning methods, such as hierarchical clustering, *k*-means clustering, and model-based clustering. In addition, we cover the issue of assessing the results of our clustering, including how one can estimate the number of groups represented by the data.

In Chapters 12 and 13, we describe methods for estimating the relationship between a set of predictors and a response variable. We cover parametric methods, such as linear regression, spline regression, and logistic regression. This is followed by generalized linear models and model selection methods. Chapter 13 includes several nonparametric methods for understanding the relationship between variables. First, we present several smoothing methods that are building blocks for additive models. For example, we discuss local polynomial regression, kernel methods, and smoothing splines. What we have just listed are methods for one predictor variable. Of course, this is rather restrictive, so we conclude the chapter with a description of regression trees, additive models, and multivariate adaptive regression splines.

An approach for simulating a distribution that has become widely used over the last several years is called Markov chain Monte Carlo. Chapter 14 covers this important topic and shows how it can be used to simulate a posterior distribution. Once we have the posterior distribution, we can use it to estimate statistics of interest (means, variances, etc.).

We also provide several appendices to aid the reader. Appendix A contains a brief introduction to MATLAB, which should help readers understand the code in the examples and exercises. Appendix B has some information on indexes for projection pursuit. In Appendix C, we include a brief description of the data sets that are mentioned in the book. Finally, we present a brief overview of notation that we use in Appendix D.

A Word About Notation

The explanation of the methods in computational statistics (and the understanding of them!) depends a lot on notation. In most instances, we follow the notation that is used in the literature for the corresponding method. Rather than try to have unique symbols throughout the book, we think it is more important to be faithful to the convention to facilitate understanding of the theory and to make it easier for readers to make the connection between the theory and the text. Because of this, the same symbols might be used in several places to denote different entities or different symbols could be used for the same thing depending on the topic. However, the meaning of the notation should be clear from the context.

In general, we *try* to stay with the convention that random variables are capital letters, whereas small letters refer to realizations of random variables. For example, X is a random variable, and x is an observed value of that random variable. When we use the term *log*, we are referring to the natural logarithm.

A symbol that is in bold refers to an array. Arrays can be row vectors, column vectors, or matrices. Typically, a matrix is represented by a bold capital letter such as **B**, while a vector is denoted by a bold lowercase letter such as **b**. Sometimes, arrays are shown with Greek symbols. For the most part, these will be shown in bold font, but we do not *always* follow this convention. Again, it should be clear from the context that the notation denotes an array.

When we are using explicit matrix notation, then we specify the dimensions of the arrays. Otherwise, we do not hold to the convention that a vector always has to be in a column format. For example, we might represent a vector of observed random variables as (x_1, x_2, x_3) or a vector of parameters as (μ, σ).

Our observed data sets will always be arranged in a matrix of dimension $n \times d$, which is denoted as X. Here n represents the number of observations we have in our sample, and d is the number of variables or dimensions. Thus, each row corresponds to a d-dimensional observation or data point. The ij-th element of X will be represented by x_{ij}. Usually, the subscript i refers to a row in a matrix or an observation, and a subscript j references a column in a matrix or a variable.

For the most part, examples are included after we explain the procedures, which include MATLAB code as we describe next. We indicate the end of an example by using a small box (❑), so the reader knows when the narrative resumes.

1.3 MATLAB® Code

Along with the explanation of the procedures, we include MATLAB commands to show how they are implemented. To make the book more readable, we will indent MATLAB code when we have several lines of code, and this can always be typed in as you see it in the book. Since all examples are available for download, you could also copy and paste the code into the MATLAB command window and execute them.

Any MATLAB commands, functions, or data sets are in courier bold font. For example, **`plot`** denotes the MATLAB plotting function. We note that due to typesetting considerations, we often have to continue a MATLAB command using the continuation punctuation (`...`). See Appendix A for more information on how this punctuation is used in MATLAB.

Since this is a book about computational statistics, we assume the reader has the MATLAB Statistics and Machine Learning Toolbox. In the rest of the book, we will refer to this toolbox with its shortened name—the Statistics Toolbox. We note in the text what functions are part of the main MATLAB software package and which functions are available only in the Statistics Toolbox.

We try to include information on MATLAB functions that are relevant to the topics covered in this text. However, this book is about the methods of computational statistics and is not meant to be a user's guide for MATLAB. Therefore, we do not claim to include all relevant MATLAB capabilities. Please see the documentation for more information on statistics in MATLAB and current functionality in the Statistics Toolbox. We also recommend the text by Martinez and Cho [2014] for those who would like a short user's guide with a focus on statistics.

The choice of MATLAB for implementation of the methods in this text is due to the following reasons:

- The commands, functions, and arguments in MATLAB are not cryptic. It is important to have a programming language that is easy to understand and intuitive, since we include the programs to help teach the concepts.
- It is used extensively by scientists and engineers.
- Student versions are available.
- It is easy to write programs in MATLAB.
- The source code or M-files can be viewed, so users can learn about the algorithms and their implementation.
- User-written MATLAB programs are freely available.
- The graphics capabilities are excellent.

It is important to note that the MATLAB code given in the body of the book is for ***learning purposes***. In many cases, it is not the most efficient way to program the algorithm. One of the purposes of including the MATLAB code is to help the reader understand the algorithms, especially how to implement them. So, we try to have the code match the procedures and to stay away from cryptic programming constructs. For example, we use **`for`** loops at times (when unnecessary!) to match the procedure. We make no claims that our code is the best way or the only way to program the algorithms.

When presenting the syntax for a MATLAB function we usually just give the basic command and usage. Most MATLAB functions have a lot more capabilities, so we urge the reader to look at the documentation. Another very useful and quick way to find out more is to type **`help`** ***`function_name`*** at the command line. This will return information such as alternative syntax, definitions of the input and output variables, and examples.

In some situations, we do not include all of the code in the text. These are cases where the MATLAB program does not provide insights about the algorithms. Including these in the body of the text would distract the reader from the important concepts being presented. However, the reader can always consult the M-files for the functions, if more information is needed.

Computational Statistics Toolbox

Some of the methods covered in this book are not available in MATLAB. So, we provide functions that implement most of the procedures that are given in the text. Note that these functions are a little different from the MATLAB code provided in the examples. In most cases, the functions allow the user to implement the algorithms for the general case. A list of the functions and their purpose is given at the end of each chapter.

The MATLAB functions for the book are in the Computational Statistics Toolbox. To make it easier to recognize these functions, we put the letters **cs** in front of *most* of the functions. We included several new functions with the second edition, some of which were written by others. We did not change these functions to make them consistent with the naming convention of the toolbox. The latest toolbox can be downloaded from

http://www.pi-sigma.info/

http://www.crcpress.com/product/ISBN/9781466592735

Information on installing the toolbox is given in the **readme** file.

Internet Resources

One of the many strong points about MATLAB is the availability of functions written by users, most of which are freely available on the Internet. With each chapter, we provide information about Internet resources for MATLAB programs (and other languages) that pertain to the techniques covered in the chapter.

The following are some Internet sources for MATLAB code. Note that these are not necessarily specific to statistics, but are for all areas of science and engineering.

- The main website at The MathWorks, Inc. has downloadable code written by MATLAB users. The website for contributed M-files and other useful information is called MATLAB Central. The link below will take you to the website where you can find MATLAB code, links to websites, news groups, webinar information, blogs, and a lot more.

http://www.mathworks.com/matlabcentral/

- A good website for user-contributed statistics programs is StatLib at Carnegie Mellon University. They have a section containing MATLAB code. The home page for StatLib is

http://lib.stat.cmu.edu

1.4 Further Reading

To gain more insight on what is computational statistics, we refer the reader to the seminal paper by Wegman [1988]. Wegman discusses many of the differences between traditional and computational statistics. He also includes a discussion on what a graduate curriculum in computational statistics should consist of and contrasts this with the more traditional course work. A later paper by Efron and Tibshirani [1991] presents a summary of the new focus in statistical data analysis that came about with the advent of the computer age. Other papers in this area include Hoaglin and Andrews [1975] and Efron [1979]. Hoaglin and Andrews discuss the connection between computing and statistical theory and the importance of properly reporting the results from simulation experiments. Efron's article presents a survey of computational statistics techniques (the jackknife, the bootstrap, error estimation in discriminant analysis, nonparametric methods, and more) for an audience with a mathematics background, but little knowledge of statistics. Chambers [1999] looks at the concepts underlying computing with data, including the challenges this presents and new directions for the future.

There are very few general books in the area of computational statistics. One is a compendium of articles edited by C. R. Rao [1993]. This is a fairly comprehensive summary of many topics pertaining to computational statistics. The texts by Gentle [2005; 2009] are excellent resources for the student or researcher. The edited volume by Gentle, Härdle, and Mori [2004] is a wonderful resource with up-to-date articles on statistical computing, statistical methodology, and applications. The book edited by Raftery, Tanner, and Wells [2002] is another source for articles on many of the topics covered in this text, such as nonparametric regression, the bootstrap, Gibbs sampling, dimensionality reduction, and many others.

For those who need a resource for learning MATLAB, we recommend a book by Hanselman and Littlefield [2011]. This gives a comprehensive overview of MATLAB, and it has information about the many capabilities of MATLAB, including how to write programs, graphics and GUIs, and much more. Martinez and Cho [2014] published a primer or short user's guide on using MATLAB for statistics.

The documentation for the Statistics Toolbox and base MATLAB is also a very good resource for learning about many of the approaches discussed in this book. See Appendix A for information about accessing these documents.

Chapter 2

Probability Concepts

2.1 Introduction

A review of probability is covered here at the outset because it provides the foundation for what is to follow: computational statistics. Readers who understand probability concepts may safely skip over this chapter.

Probability is the mechanism by which we can manage the uncertainty underlying all real world data and phenomena. It enables us to gauge our degree of belief and to quantify the lack of certitude that is inherent in the process that generates the data we are analyzing. For example:

- To understand and use statistical hypothesis testing, one needs knowledge of the sampling distribution of the test statistic.
- To evaluate the performance (e.g., standard error, bias, etc.) of an estimate, we must know its sampling distribution.
- To adequately simulate a real system, one needs to understand the probability distributions that correctly model the underlying processes.
- To build classifiers to predict what group an object belongs to based on a set of features, one can estimate the probability density function that describes the individual classes.

In this chapter, we provide a brief overview of probability concepts and distributions as they pertain to computational statistics. In Section 2.2, we define probability and discuss some of its properties. In Section 2.3, we cover conditional probability, independence, and Bayes' theorem. Expectations are defined in Section 2.4, and common distributions and their uses in modeling physical phenomena are discussed in Section 2.5. In Section 2.6, we summarize some MATLAB® functions that implement the ideas from Chapter 2. Finally, in Section 2.7 we provide additional resources for the reader who requires a more theoretical treatment of probability.

2.2 Probability

Background

A ***random experiment*** is defined as a process or action whose outcome cannot be predicted with certainty and would likely change when the experiment is repeated. The variability in the outcomes might arise from many sources: slight errors in measurements, choosing different objects for testing, etc. The ability to model and analyze the outcomes from experiments is at the heart of statistics. Some examples of random experiments that arise in different disciplines are given next.

- Engineering: Data are collected on the number of failures of piston rings in the legs of steam-driven compressors. Engineers would be interested in determining the probability of piston failure in each leg and whether the failure varies among the compressors [Hand, et al.; Davies and Goldsmith, 1972].
- Medicine: The oral glucose tolerance test is a diagnostic tool for early diabetes mellitus. The results of the test are subject to variation because of different rates at which people absorb the glucose, and the variation is particularly noticeable in pregnant women. Scientists would be interested in analyzing and modeling the variation of glucose before and after pregnancy [Andrews and Herzberg, 1985].
- Manufacturing: Manufacturers of cement are interested in the tensile strength of their product. The strength depends on many factors, one of which is the length of time the cement is dried. An experiment is conducted where different batches of cement are tested for tensile strength after different drying times. Engineers would like to determine the relationship between drying time and tensile strength of the cement [Hand, et al., 1994; Hald, 1952].
- Software Engineering: Engineers measure the failure times in CPU seconds of a command and control software system. These data are used to obtain models to predict the reliability of the software system [Hand, et al., 1994; Musa, et al., 1987].

The ***sample space*** is the set of all outcomes from an experiment. It is possible sometimes to list all outcomes in the sample space. This is especially true in the case of some discrete random variables. Examples of these sample spaces are listed next.

- When observing piston ring failures, the sample space is $\{1, 0\}$, where 1 represents a failure and 0 represents a non-failure.
- If we roll a six-sided die and count the number of dots on the face, then the sample space is $\{1, 2, 3, 4, 5, 6\}$.

The outcomes from random experiments are often represented by an uppercase variable such as X. This is called a ***random variable***, and its value is subject to the uncertainty intrinsic to the experiment. Formally, a random variable is a real-valued function defined on the sample space. As we see in the remainder of the text, a random variable can take on different values according to a probability distribution. Using our examples of experiments from above, a random variable X might represent the failure time of a software system or the glucose level of a patient. The observed value of a random variable X is denoted by a lowercase x. For instance, a random variable X might represent the number of failures of piston rings in a compressor, and $x = 5$ would indicate we observed 5 piston ring failures.

Random variables can be discrete or continuous. A ***discrete random variable*** can take on values from a finite or countably infinite set of numbers. Examples of discrete random variables are the number of defective parts or the number of typographical errors on a page. A ***continuous random variable*** is one that can take on values from an interval of real numbers. Examples of continuous random variables are the inter-arrival times of planes at a runway, the average weight of tablets in a pharmaceutical production line, or the average voltage of a power plant at different times.

We cannot list all outcomes from an experiment when we observe a continuous random variable because there are an infinite number of possibilities. However, we could specify the interval of values that X can take on. For example, if the random variable X represents the tensile strength of cement, then the sample space might be $(0, \infty)$ kg/cm^2.

An ***event*** is a subset of outcomes in the sample space. An event might be that a piston ring is defective or that the tensile strength of cement is in the range 40 to 50 kg/cm^2. The probability of an event is usually expressed using the random variable notation illustrated next.

- Discrete Random Variables: Letting 1 represent a defective piston ring and letting 0 represent a good piston ring, then the probability of the event that a piston ring is defective would be written as

$$P(X = 1) .$$

- Continuous Random Variables: Let X denote the tensile strength of cement. The probability that an observed tensile strength is in the range 40 to 50 kg/cm^2 is expressed as

$$P(40 \text{ kg/cm}^2 \leq X \leq 50 \text{ kg/cm}^2) .$$

Some events have a special property when they are considered together. Two events that cannot occur simultaneously or jointly are called ***mutually exclusive events***. This means that the intersection of the two events is the empty set and the probability of the events occurring together is zero. For example, a piston ring cannot be both defective and good at the same time. So, the event of getting a defective part and the event of getting a good part are mutually exclusive events. The definition of mutually exclusive events can be extended to any number of events by considering all pairs of events. Every pair of events must be mutually exclusive for all of them to be mutually exclusive.

Probability

Probability is a measure of the likelihood that some event will occur. It is also a way to quantify or to gauge the likelihood that an observed measurement or random variable will take on values within some set or range of values. Probabilities always range between 0 and 1. A ***probability distribution*** of a random variable describes the probabilities associated with each possible value for the random variable.

We first briefly describe two somewhat classical methods for assigning probabilities: the ***equal likelihood model*** and the ***relative frequency method***. When we have an experiment where each of n outcomes is equally likely, then we assign a probability mass of $1/n$ to each outcome. This is the equal likelihood model. Some experiments where this model can be used are flipping a fair coin, tossing an unloaded die, or randomly selecting a card from a deck of cards.

With the relative frequency method, we conduct the experiment n times and record the outcome. The probability of event E is then assigned by $P(E) = f/n$, where f denotes the number of experimental outcomes that satisfy event E.

Another way to find the desired probability that an event occurs is to use a ***probability density function*** when we have continuous random variables or a ***probability mass function*** in the case of discrete random variables. Section 2.5 contains several examples of probability density (mass) functions. In this text, $f(x)$ is typically used to represent the probability mass or density function for either discrete or continuous random variables, respectively. We now discuss how to find probabilities using these functions, first for the continuous case and then for discrete random variables.

To find the probability that a continuous random variable falls in a particular interval of real numbers, we have to calculate the appropriate area under the curve of $f(x)$. Thus, we have to evaluate the integral of $f(x)$ over the interval of random variables corresponding to the event of interest. This is represented by

$$P(a \leq X \leq b) = \int_a^b f(x)dx. \tag{2.1}$$

The area under the curve of $f(x)$ between a and b represents the probability that an observed value of the random variable X will assume a value between a and b. This concept is illustrated in Figure 2.1 where the shaded area represents the desired probability.

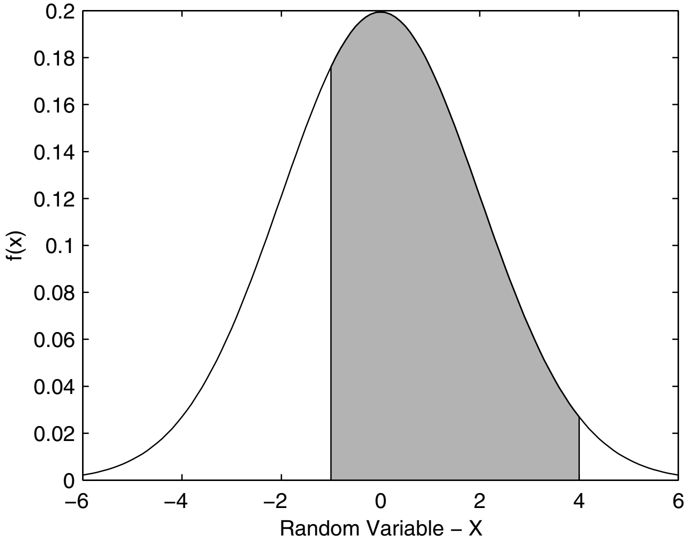


FIGURE 2.1
The area under the curve of f(x) between -1 and 4 is the same as the probability that an observed value of the random variable will assume a value in the same interval.

It should be noted that a valid probability density function should be nonnegative, and the total area under the curve must equal 1. If this is not the case, then the probabilities will not be properly restricted to the interval $[0, 1]$. This will be an important consideration in Chapter 9 when we discuss probability density estimation techniques.

The ***cumulative distribution function*** $F(x)$ is defined as the probability that the random variable X assumes a value less than or equal to a given x. This is calculated from the probability density function, as follows:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt. \tag{2.2}$$

It is obvious from Equation 2.2 that the cumulative distribution function takes on values between 0 and 1, so $0 \le F(x) \le 1$. A probability density function, along with its associated cumulative distribution function, are illustrated in Figure 2.2.

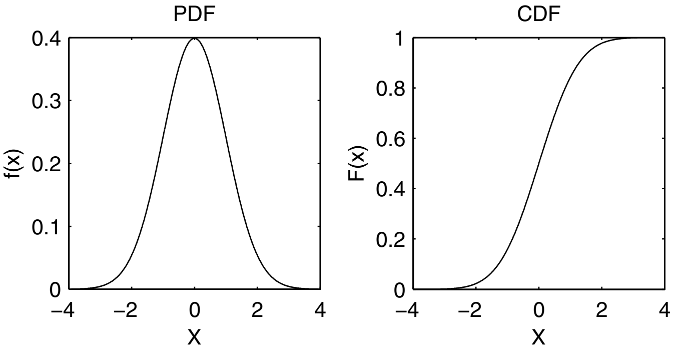


FIGURE 2.2
This shows the probability density function on the left with the associated cumulative distribution function on the right. Notice that the cumulative distribution function takes on values between 0 and 1.

For a discrete random variable X, that can take on values $x_1, x_2, \ldots$, the probability mass function is given by

$$f(x_i) = P(X = x_i); \qquad i = 1, 2, \ldots, \tag{2.3}$$

and the cumulative distribution function is

$$F(a) = \sum_{x_i \le a} f(x_i); \qquad i = 1, 2, \ldots. \tag{2.4}$$

Axioms of Probability

Probabilities follow certain axioms that can be useful in computational statistics. We let S represent the sample space of an experiment and E represent some event that is a subset of S.

AXIOM 1
The probability of event E must be between 0 and 1:

$$0 \le P(E) \le 1 .$$

AXIOM 2

$$P(S) = 1 .$$

AXIOM 3
For mutually exclusive events, $E_1, E_2, \ldots, E_k$,

$$P(E_1 \cup E_2 \cup \ldots \cup E_k) = \sum_{i=1}^{k} P(E_i) .$$

Axiom 1 has been discussed before and simply states that a probability must be between 0 and 1. Axiom 2 says that an outcome from our experiment must occur. Axiom 3 enables us to calculate the probability that at least one of the mutually exclusive events $E_1, E_2, \ldots, E_k$ occurs by summing the individual probabilities.

2.3 Conditional Probability and Independence

Conditional Probability

Conditional probability is an important concept. It is used to define independent events and enables us to revise our degree of belief given that another event has occurred. Conditional probability arises in situations where we need to calculate a probability based on some partial information concerning the experiment, and we will see that it plays a vital role in supervised learning applications.

The ***conditional probability*** of event E given event F is defined as follows:

CONDITIONAL PROBABILITY

$$P(E|F) = \frac{P(E \cap F)}{P(F)}; \quad P(F) > 0. \tag{2.5}$$

Here $P(E \cap F)$ represents the ***joint probability*** that both E and F occur together, and $P(F)$ is the probability that event F occurs. We can rearrange Equation 2.5 to get the following rule:

MULTIPLICATION RULE

$$P(E \cap F) = P(F)P(E|F) . \tag{2.6}$$

Independence

Often we can assume that the occurrence of one event does not affect whether or not some other event happens. For example, say a couple would like to have two children, and their first child is a boy. The gender of their second child does not depend on the gender of the first child. The fact that we know they have a boy already does not change the probability that the second child is a boy. Similarly, we can sometimes assume that the value we observe for a random variable is not affected by the observed value of other random variables.

These types of events and random variables are called ***independent***. If *events* are independent, then knowing that one event has occurred does not change our degree of belief or the likelihood that the other event occurs. If *random variables* are independent, then the observed value of one random variable does not affect the observed value of another.

In general, the conditional probability $P(E|F)$ is not equal to $P(E)$. In these cases, the events are called ***dependent***. Sometimes, we can assume independence based on the situation or the experiment, which was the case with our example above. However, to show independence mathematically, we must use the following definition.

INDEPENDENT EVENTS

Two events E and F are said to be independent if and only if any of the following are true:

$$\begin{aligned} P(E \cap F) &= P(E)P(F), \\ P(E) &= P(E|F). \end{aligned} \tag{2.7}$$

Note that if events E and F are independent, then the Multiplication Rule in Equation 2.6 becomes

$$P(E \cap F) = P(F)P(E) ,$$

which means that we simply multiply the individual probabilities for each event together. This can be extended to k events to give

$$P(E_1 \cap E_2 \cap \ldots \cap E_k) = \prod_{i=1}^{k} P(E_i), \tag{2.8}$$

where events E_i and E_j (for all i and j, $i \neq j$) are independent.

Bayes' Theorem

Sometimes we start an analysis with an initial degree of belief that an event will occur. Later on, we might obtain some additional information about the event that would change our belief about the probability that the event will occur. The initial probability is called a ***prior probability***. Using the new information, we can update the prior probability using Bayes' theorem to obtain the ***posterior probability***.

The experiment of recording piston ring failure in compressors mentioned at the beginning of the chapter is an example of where Bayes' theorem might be used, and we derive Bayes' theorem using this example. Suppose our piston rings are purchased from two manufacturers: 60% from *manufacturer A* and 40% from *manufacturer B*.

Let M_A denote the event that a part comes from manufacturer A and M_B represent the event that a piston ring comes from manufacturer B. If we select a part at random from our supply of piston rings, we would assign probabilities to these events as follows:

$$P(M_A) = 0.6,$$
$$P(M_B) = 0.4.$$

These are our prior probabilities that the piston rings are from the individual manufacturers.

Say we are interested in knowing the probability that a piston ring that subsequently failed came from *manufacturer A*. This would be the posterior probability that it came from *manufacturer A*, given that the piston ring failed. The additional information we have about the piston ring is that it failed, and we use this to update our degree of belief that it came from *manufacturer A*.

Bayes' theorem can be derived from the definition of conditional probability (Equation 2.5). Writing this in terms of our events, we are interested in the following probability:

$$P(M_A|F) = \frac{P(M_A \cap F)}{P(F)}, \tag{2.9}$$

where $P(M_A|F)$ represents the posterior probability that the part came from *manufacturer A*, and F is the event that the piston ring failed. Using the Multiplication Rule (Equation 2.6), we can write the numerator of Equation

2.9 in terms of event F and our prior probability that the part came from *manufacturer A*, as follows:

$$P(M_A|F) = \frac{P(M_A \cap F)}{P(F)} = \frac{P(M_A)P(F|M_A)}{P(F)}. \qquad (2.10)$$

The next step is to find $P(F)$. The only way that a piston ring will fail is if: (1) it failed and it came from *manufacturer A*, or (2) it failed and it came from *manufacturer B*. Thus, using the third axiom of probability, we can write

$$P(F) = P(M_A \cap F) + P(M_B \cap F).$$

Applying the Multiplication Rule as before, we have

$$P(F) = P(M_A)P(F|M_A) + P(M_B)P(F|M_B). \qquad (2.11)$$

Substituting this for $P(F)$ in Equation 2.10, we write the posterior probability as

$$P(M_A|F) = \frac{P(M_A)P(F|M_A)}{P(M_A)P(F|M_A) + P(M_B)P(F|M_B)}. \qquad (2.12)$$

Note that we need to find the probabilities $P(F|M_A)$ and $P(F|M_B)$. These are the probabilities that a piston ring will fail given it came from the corresponding manufacturer. These must be estimated in some way using available information (e.g., past failures). When we revisit Bayes' theorem in the context of statistical pattern recognition (Chapter 10), these are the probabilities that are estimated to construct a certain type of classifier.

Equation 2.12 is Bayes' theorem for a situation where only two outcomes are possible. In general, Bayes' theorem can be written for any number of mutually exclusive events, $E_1, \ldots, E_k$, whose union makes up the entire sample space. This is given next.

BAYES' THEOREM

$$P(E_i|F) = \frac{P(E_i)P(F|E_i)}{P(E_1)P(F|E_1) + \ldots + P(E_k)P(F|E_k)}. \qquad (2.13)$$

2.4 Expectation

Expected values and variances are important concepts in statistics. They are used to describe distributions, to evaluate the performance of estimators, to obtain test statistics in hypothesis testing, and many other applications.

Mean and Variance

The ***mean*** or ***expected value*** of a random variable is defined using the probability density or mass function. It provides a measure of central tendency of the distribution. If we observe many values of the random variable and take the average of them, we would expect that value to be close to the mean. The expected value is defined below for the discrete case.

EXPECTED VALUE - DISCRETE RANDOM VARIABLES

$$\mu = E[X] = \sum_{i=1}^{\infty} x_i f(x_i) . \tag{2.14}$$

We see from the definition that the expected value is a sum of all possible values of the random variable where each one is weighted by the probability that X will take on that value.

The ***variance*** of a discrete random variable is given by the following definition.

VARIANCE - DISCRETE RANDOM VARIABLES

For $\mu < \infty$,

$$\sigma^2 = V(X) = E[(X-\mu)^2] = \sum_{i=1}^{\infty} (x_i - \mu)^2 f(x_i) . \tag{2.15}$$

From Equation 2.15, we see that the variance is the sum of the squared distances from the mean, each one weighted by the probability that $X = x_i$. Variance is a measure of dispersion in the distribution. If a random variable has a large variance, then an observed value of the random variable is more likely to be far from the mean μ. The standard deviation σ is the square root of the variance.

The mean and variance for continuous random variables are defined similarly, with the summation replaced by an integral. The mean and variance of a continuous random variable are given next.

EXPECTED VALUE - CONTINUOUS RANDOM VARIABLES

$$\mu = E[X] = \int_{-\infty}^{\infty} xf(x)dx. \tag{2.16}$$

VARIANCE - CONTINUOUS RANDOM VARIABLES

For $\mu < \infty$,

$$\sigma^2 = V(X) = E[(X-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx. \tag{2.17}$$

We note that Equation 2.17 can also be written as

$$V(X) = E[X^2] - \mu^2 = E[X^2] - (E[X])^2.$$

Other expected values that are of interest in statistics are the ***moments*** of a random variable. These are the expectation of powers of the random variable. In general, we define the ***r-th moment*** as

$$\mu'_r = E[X^r], \tag{2.18}$$

and the ***r-th central moment*** as

$$\mu_r = E[(X-\mu)^r]. \tag{2.19}$$

The mean corresponds to μ'_1, and the variance is given by μ_2.

Skewness

The third central moment μ_3 is often called a measure of asymmetry or skewness in the distribution. The uniform and the normal distribution are examples of symmetric distributions. The gamma and the exponential are examples of skewed or asymmetric distributions. The following ratio is called the ***coefficient of skewness***, which is often used to measure this characteristic:

$$\gamma_1 = \frac{\mu_3}{\mu_2^{3/2}}. \tag{2.20}$$

Distributions that are skewed to the left will have a negative coefficient of skewness, and distributions that are skewed to the right will have a positive value [Hogg and Craig, 1978]. The coefficient of skewness is zero for symmetric distributions. However, a coefficient of skewness equal to zero does not imply that the distribution must be symmetric.

Kurtosis

Skewness is one way to measure a type of departure from normality. ***Kurtosis*** measures a different type of departure from normality by indicating the extent of the peak (or the degree of flatness near its center) in a distribution. The ***coefficient of kurtosis*** is given by the following ratio:

$$\gamma_2 = \frac{\mu_4}{\mu_2^2}. \tag{2.21}$$

We see that this is the ratio of the fourth central moment divided by the square of the variance. If the distribution is normal, then this ratio is equal to 3. A ratio greater than 3 indicates more values in the neighborhood of the mean (is more peaked than the normal distribution). If the ratio is less than 3, then it is an indication that the curve is flatter than the normal.

Sometimes the ***coefficient of excess kurtosis*** is used as a measure of kurtosis. This is given by

$$\gamma_2' = \frac{\mu_4}{\mu_2^2} - 3. \tag{2.22}$$

In this case, distributions that are more peaked than the normal correspond to a positive value of γ_2', and those with a flatter top have a negative coefficient of excess kurtosis.

2.5 Common Distributions

In this section, we provide a review of some useful probability distributions and briefly describe some applications to modeling data. Most of these distributions are used in later chapters, so we take this opportunity to define them and to fix our notation. We first cover two important discrete distributions: the binomial and the Poisson. These are followed by several continuous distributions: the uniform, the normal, the exponential, the gamma, the chi-square, the Weibull, the beta, the Student's *t* distribution, the multivariate normal, and the multivariate *t* distribution.

Binomial

Let's say that we have an experiment, whose outcome can be labeled as a "success" or a "failure." If we let $X = 1$ denote a successful outcome and $X = 0$ represent a failure, then we can write the probability mass function as

$$\begin{aligned} f(0) &= P(X=0) = 1-p, \\ f(1) &= P(X=1) = p, \end{aligned} \tag{2.23}$$

where p represents the probability of a successful outcome. A random variable that follows the probability mass function in Equation 2.23 for $0 < p < 1$ is called a ***Bernoulli random variable***.

Now suppose we repeat this experiment for n trials, where each trial is independent (the outcome from one trial does not influence the outcome of another) and results in a success with probability p. If X denotes the number of successes in these n trials, then X follows the binomial distribution with parameters n and p. Examples of binomial distributions with different parameters are shown in Figure 2.3.

To calculate a binomial probability, we use the following formula:

$$f(x;n,p) = P(X=x) = \binom{n}{x} p^x (1-p)^{n-x}; \qquad x = 0, 1, \ldots, n. \tag{2.24}$$

The mean and variance of a binomial distribution are given by

$$E[X] = np,$$

and

$$V(X) = np(1-p).$$

Some examples where the results of an experiment can be modeled by a binomial random variable are

- A drug has probability 0.90 of curing a disease. It is administered to 100 patients, where the outcome for each patient is either cured or not cured. If X is the number of patients cured, then X is a binomial random variable with parameters (100, 0.90).

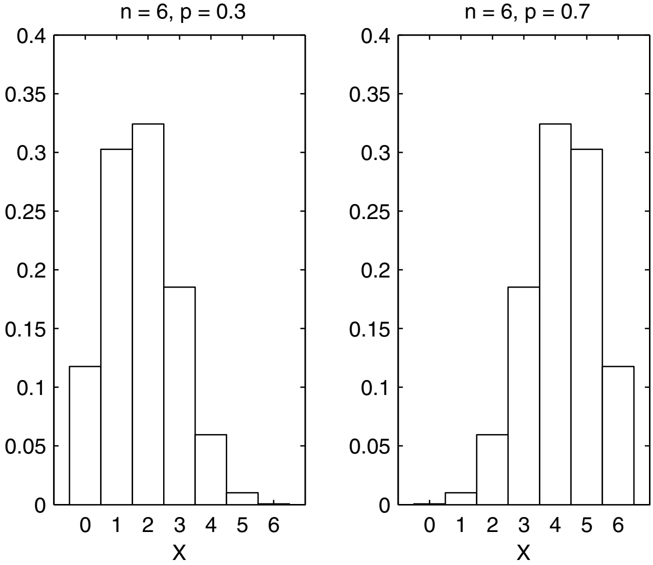


FIGURE 2.3
Examples of the binomial distribution for different success probabilities.

- The National Institute of Mental Health estimates that there is a 20% chance that an adult American suffers from a psychiatric disorder. Fifty adult Americans are randomly selected. If we let X represent the number who have a psychiatric disorder, then X takes on values according to the binomial distribution with parameters (50, 0.20).
- A manufacturer of computer chips finds that on the average 5% are defective. To monitor the manufacturing process, they take a random sample of size 75. If the sample contains more than five defective chips, then the process is stopped. The binomial distribution with parameters (75, 0.05) can be used to model the random variable X, where X represents the number of defective chips.

Example 2.1

Suppose there is a 20% chance that an adult American suffers from a psychiatric disorder. We randomly sample 25 adult Americans. If we let X represent the number of people who have a psychiatric disorder, then X is a binomial random variable with parameters $(25, 0.20)$. We are interested in the probability that at most 3 of the selected people have such a disorder. We can use the MATLAB Statistics Toolbox function **`binocdf`** to determine $P(X \leq 3)$, as follows:

```
prob = binocdf(3,25,0.2);
```

We could also sum up the individual values of the probability mass function from $X = 0$ to $X = 3$:

```
prob2 = sum(binopdf(0:3,25,0.2));
```

Both of these commands return a probability of 0.234. We now show how to generate the binomial distributions shown in Figure 2.3.

```
% Get the values for the domain, x.
x = 0:6;
% Get the values of the probability mass function.
% First for n = 6, p = 0.3:
pdf1 = binopdf(x,6,0.3);
% Now for n = 6, p = 0.7:
pdf2 = binopdf(x,6,0.7);
```

Now we have the values for the probability mass function (or the heights of the bars). The plots are obtained using the following code:

```
% Do the plots.
subplot(1,2,1),bar(x,pdf1,1,'w')
title(' n = 6, p = 0.3')
xlabel('X'),ylabel('f(X)')
axis square
subplot(1,2,2),bar(x,pdf2,1,'w')
title(' n = 6, p = 0.7')
xlabel('X'),ylabel('f(X)')
axis square
```

❑

Poisson

If a random variable X is a ***Poisson random variable*** with parameter λ, $\lambda > 0$, then it has the probability mass function given by

$$f(x;\lambda) = P(X = x) = e^{-\lambda}\frac{\lambda^x}{x!}; \quad x = 0, 1, \dots , \tag{2.25}$$

where $x!$ denotes the factorial of x. The factorial of a non-negative integer x is the product of all positive integers less than or equal to x.

The expected value and variance of a Poisson random variable are both λ, thus,

$$E[X] = \lambda ,$$

and

$$V(X) = \lambda .$$

The Poisson distribution can be used in many applications. Examples where a discrete random variable might follow a Poisson distribution are

- the number of typographical errors on a page,
- the number of vacancies in a company during a month, or
- the number of defects in a length of wire.

The Poisson distribution is often used to approximate the binomial. When n is large and p is small (so np is moderate), then the number of successes occurring can be approximated by the Poisson random variable with parameter $\lambda = np$.

The Poisson distribution is also appropriate for some applications where events occur at points in time or space. Examples include the arrival of jobs at a business, the arrival of aircraft on a runway, and the breakdown of machines at a manufacturing plant. The number of events in these applications can be described by a ***Poisson process***.

Let $N(t)$, $t \geq 0$, represent the number of events that occur in the time interval $[0, t]$. For each interval $[0, t]$, $N(t)$ is a random variable that can take on values $0, 1, 2, \ldots$. If the following conditions are satisfied, then the counting process $\{N(t), t \geq 0\}$ is said to be a Poisson process with mean rate λ [Ross, 2000]:

1. $N(0) = 0$.
2. The process has independent increments.
3. The number $N(t)$ of events in an interval of length t follows a Poisson distribution with mean λt. Thus, for $s \geq 0$ and $t \geq 0$,

$$P(N(t+s) - N(s) = k) = e^{-\lambda t}\frac{(\lambda t)^k}{k!}; \qquad k = 0, 1, \ldots . \tag{2.26}$$

From the third condition, we know that the process has stationary increments. This means that the distribution of the number of events in an interval depends only on the length of the interval and not on the starting point. The second condition specifies that the number of events in one interval does not affect the number of events in other intervals. The first condition states that the counting starts at time $t = 0$. The expected value of $N(t)$ is given by

$$E[N(t)] = \lambda t .$$

Example 2.2

In preparing this text, we executed the spell check command, and the editor reviewed the manuscript for typographical errors. In spite of this, some mistakes might be present. Assume that the number of typographical errors per page follows the Poisson distribution with parameter $\lambda = 0.25$. We calculate the probability that a page will have at least two errors as follows:

$$P(X \geq 2) = 1 - \{P(X=0) + P(X=1)\} = 1 - e^{-0.25} - e^{-0.25}0.25 \approx 0.0265.$$

We can get this probability using the MATLAB Statistics Toolbox function **poisscdf**. Note that $P(X=0) + P(X=1)$ is the Poisson cumulative distribution function for $a = 1$ (see Equation 2.4), which is why we use **1** as the argument to **poisscdf**.

```
prob = 1-poisscdf(1,0.25);
```

❑

Example 2.3

Suppose that accidents at a certain intersection occur in a manner that satisfies the conditions for a Poisson process with a rate of 2 per week ($\lambda = 2$). What is the probability that at most 3 accidents will occur during the next 2 weeks? Using Equation 2.26, we have

$$P(N(2) \leq 3) = \sum_{k=0}^{3} P(N(2) = k).$$

Expanding this out yields

$$P(N(2) \leq 3) = e^{-4} + 4e^{-4} + \frac{4^2}{2!}e^{-4} + \frac{4^3}{3!}e^{-4} \approx 0.4335.$$

As before, we can use the **poisscdf** function with parameter given by $\lambda t = 2 \cdot 2$.

```
prob = poisscdf(3,2*2);
```

❑

Uniform

Perhaps one of the most important distributions is the ***uniform distribution*** for continuous random variables. One reason is that the uniform (0, 1) distribution is used as the basis for simulating most random variables as we discuss in Chapter 4.

A random variable that is uniformly distributed over the interval (*a, b*) follows the probability density function given by

$$f(x;a, b) = \frac{1}{b-a}; \quad a < x < b. \tag{2.27}$$

The parameters for the uniform are the interval endpoints, *a* and *b*. The mean and variance of a uniform random variable are given by

$$E[X] = \frac{a+b}{2}$$

and

$$V(X) = \frac{(b-a)^2}{12}.$$

The cumulative distribution function for a uniform random variable is

$$F(x) = \begin{cases} 0; & x \le a \\ \dfrac{x-a}{b-a}; & a < x < b \\ 1; & x \ge b. \end{cases} \tag{2.28}$$

Example 2.4

In this example, we illustrate the uniform probability density function over the interval (0, 10), along with the corresponding cumulative distribution function. The MATLAB Statistics Toolbox functions **unifpdf** and **unifcdf** are used to get the desired functions over the interval.

```
% First get the domain over which we will
% evaluate the functions.
x = -1:.1:11;
% Now get the probability density function
% values at x.
pdf = unifpdf(x,0,10);
% Now get the cdf.
cdf = unifcdf(x,0,10);
```

Plots of the functions are provided in Figure 2.4, where the probability density function is shown in the left plot and the cumulative distribution on the right. These plots are constructed using the following MATLAB commands.

```
% Do the plots.
subplot(1,2,1),plot(x,pdf)
title('PDF')
xlabel('X'),ylabel('f(X)')
axis([-1 11 0 0.2])
axis square
subplot(1,2,2),plot(x,cdf)
title('CDF')
xlabel('X'),ylabel('F(X)')
axis([-1 11 0 1.1])
axis square
```

❑

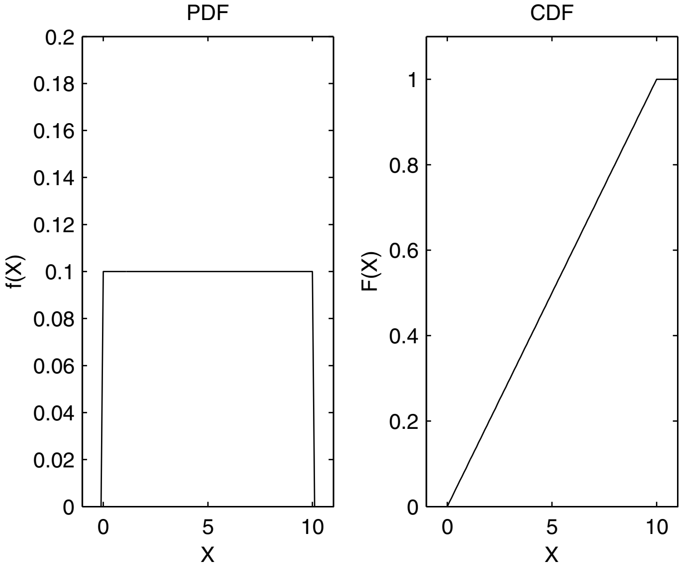


FIGURE 2.4
On the left is a plot of the probability density function for the uniform (0, 10). Note that the height of the curve is given by $1/(b-a) = 1/10 = 0.10$. *The corresponding cumulative distribution function is shown on the right.*

Normal

A well-known distribution in statistics and engineering is the ***normal distribution***. Also called the ***Gaussian distribution***, it has a continuous probability density function given by

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \tag{2.29}$$

where

$$-\infty < x < \infty;\ -\infty < \mu < \infty;\ \sigma^2 > 0\ .$$

The normal distribution is completely determined by its parameters (μ and σ^2), which are also the expected value and variance for a normal random variable. The notation $X \sim N(\mu, \sigma^2)$ is used to indicate that a random variable X is normally distributed with mean μ and variance σ^2. Several normal distributions with different parameters are shown in Figure 2.5.

Some special properties of the normal distribution are given here.

- The value of the probability density function approaches zero as x approaches positive and negative infinity.
- The probability density function is centered at the mean μ, and the maximum value of the function occurs at $x = \mu$.
- The probability density function for the normal distribution is symmetric about the mean μ.

The special case of a ***standard normal*** random variable is one whose mean is zero ($\mu = 0$) and whose standard deviation is one ($\sigma = 1$). If X is normally distributed, then

$$Z = \frac{X-\mu}{\sigma} \tag{2.30}$$

is a standard normal random variable.

Traditionally, the cumulative distribution function of a standard normal random variable is denoted by

$$\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z}\exp\left\{-\frac{y^2}{2}\right\}dy\,. \tag{2.31}$$

The cumulative distribution function for a standard normal random variable can be calculated using the error function, denoted by *erf*. The relationship between these functions is given by

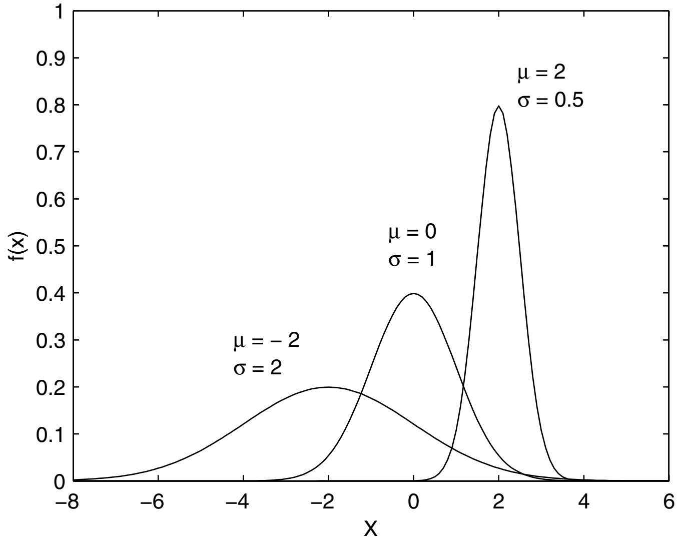


FIGURE 2.5
Examples of probability density functions for normally distributed random variables. Note that as the variance increases, the height of the probability density function at the mean decreases.

$$\Phi(z) = \frac{1}{2}erf\left(\frac{z}{\sqrt{2}}\right)+\frac{1}{2}. \tag{2.32}$$

The error function can be calculated in MATLAB using **`erf(x)`**. The MATLAB Statistics Toolbox has a function called **`normcdf(x,mu,sigma)`** that will calculate the cumulative distribution function for values in x. Its use is illustrated in the example given next.

Example 2.5

Similar to the uniform distribution, the functions **`normpdf`** and **`normcdf`** are available in the MATLAB Statistics Toolbox for calculating the probability density function and cumulative distribution function for the Gaussian. There is another special function called **`normspec`** that determines the probability that a random variable X assumes a value between two limits, where X is normally distributed with mean μ and standard deviation σ. This function also plots the normal density, where the area between the specified limits is shaded. The syntax is shown next.

```
% Set up the parameters for the normal distribution.
mu = 5;
```

```
sigma = 2;
% Set up the upper and lower limits. These are in
% the two element vector 'specs'.
specs = [2, 8];
prob = normspec(specs, mu, sigma);
```

The resulting plot is shown in Figure 2.6. By default, MATLAB will put the probability between the limits in the title, which in this case is 0.87. Note that the default title and labels can be changed easily using the **`title, xlabel,`** and **`ylabel`** functions. You can also obtain tail probabilities by using **`-Inf`** as the first element of **`specs`** to designate no lower limit or **`Inf`** as the second element to indicate no upper limit.

❑

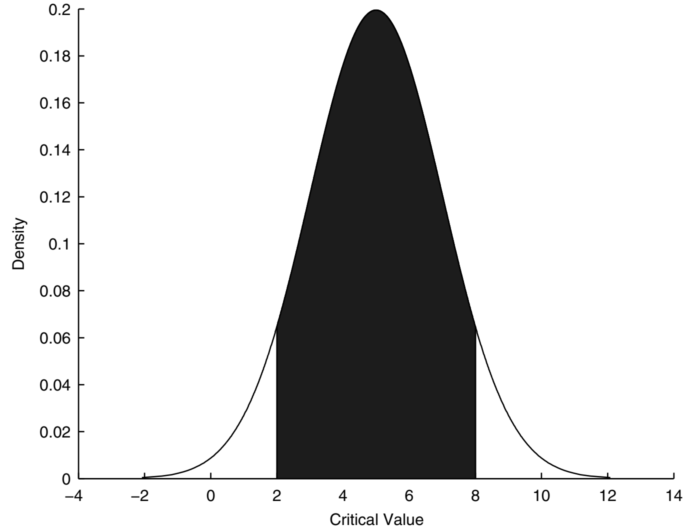


FIGURE 2.6
This shows the output from the function **`normspec`**. *Note that it shades the area between the lower and upper limits that are specified as input arguments. The probability between the limits is approximately 0.87.*

Exponential

The ***exponential distribution*** can be used to model the amount of time until a specific event occurs or to model the time between independent events. Some examples where an exponential distribution is appropriate are

- The time until the computer locks up,

- The time between arrivals of telephone calls, or
- The time until a part fails.

The exponential probability density function with parameter λ is

$$f(x;\lambda) = \lambda e^{-\lambda x}; \quad x \geq 0; \quad \lambda > 0\,. \tag{2.33}$$

The mean and variance of an exponential random variable are given by the following:

$$E[X] = \frac{1}{\lambda}$$

and

$$V(X) = \frac{1}{\lambda^2}\,.$$

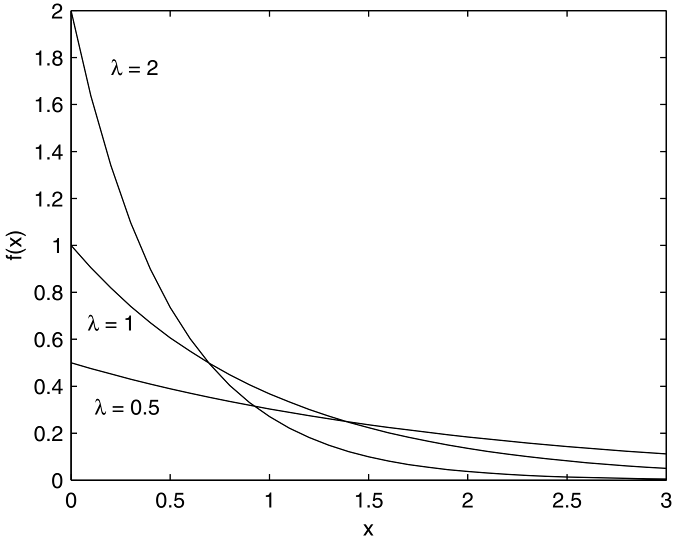


FIGURE 2.7
Exponential probability density functions for various values of λ.

The cumulative distribution function of an exponential random variable is given by

$$F(x) = \begin{cases} 0; & x < 0 \\ 1 - e^{-\lambda x}; & x \geq 0. \end{cases} \qquad (2.34)$$

The exponential distribution is the only continuous distribution that has the ***memoryless property***. This property describes the fact that the remaining lifetime of an object (whose lifetime follows an exponential distribution) does not depend on the amount of time it has already lived. This property is represented by the following equality, where $s \geq 0$ and $t \geq 0$:

$$P(X > s + t | X > s) = P(X > t).$$

In words, this means that the probability that the object will operate for time $s + t$, given it has already operated for time s, is simply the probability that it operates for time t.

When the exponential distribution is used to represent interarrival times, then the parameter λ is a rate with units of arrivals per time period. When the exponential is used to model the time until a failure occurs, then λ is the failure rate. Several examples of the exponential distribution are shown in Figure 2.7.

Example 2.6

The time between arrivals of vehicles at an intersection follows an exponential distribution with a mean of 12 seconds. What is the probability that the time between arrivals is 10 seconds or less? We are given the average interarrival time, so $\lambda = 1/12$. The required probability is obtained from Equation 2.34 as follows

$$P(X \leq 10) = 1 - e^{-(1/12)10} \approx 0.57.$$

You can calculate this using the MATLAB Statistics Toolbox function **`expcdf(x, 1/λ)`**. Note that this MATLAB function is based on a different definition of the exponential probability density function, which is given by

$$f(x;\mu) = \frac{1}{\mu} e^{-\frac{x}{\mu}}; \quad x \geq 0; \quad \mu > 0. \qquad (2.35)$$

In the Computational Statistics Toolbox, we include a function called **`csexpoc(x, λ)`** that calculates the exponential cumulative distribution function using Equation 2.34.

❑

Gamma

The ***gamma probability density function*** with parameters $\lambda > 0$ and $t > 0$ is given by

$$f(x;\lambda, t) = \frac{\lambda e^{-\lambda x}(\lambda x)^{t-1}}{\Gamma(t)}; \quad x \geq 0, \tag{2.36}$$

where t is a shape parameter, and λ is the scale parameter. The gamma function $\Gamma(t)$ is defined as

$$\Gamma(t) = \int_0^{\infty} e^{-y} y^{t-1} dy. \tag{2.37}$$

For integer values of t, Equation 2.37 becomes

$$\Gamma(t) = (t-1)!\,. \tag{2.38}$$

Note that for $t = 1$, the gamma density is the same as the exponential. When t is a positive integer, the gamma distribution can be used to model the amount of time one has to wait until t events have occurred, if the inter-arrival times are exponentially distributed.

The mean and variance of a gamma random variable are

$$E[X] = \frac{t}{\lambda}$$

and

$$V(X) = \frac{t}{\lambda^2}.$$

The cumulative distribution function for a gamma random variable is calculated using [Meeker and Escobar, 1998; Banks, et al., 2001]

$$F(x;\lambda, t) = \begin{cases} 0; & x \leq 0 \\ \dfrac{1}{\Gamma(t)} \displaystyle\int_0^{\lambda x} y^{t-1} e^{-y} dy; & x > 0\,. \end{cases} \tag{2.39}$$

Equation 2.39 can be evaluated in MATLAB using the **`gammainc(`**λ**`*x,t)`** function, where the above notation is used for the arguments.

Example 2.7

We plot the gamma probability density function for $\lambda = t = 1$ (this should look like the exponential), $\lambda = t = 2$, and $\lambda = t = 3$. You can use the MATLAB Statistics Toolbox function `gampdf(x,t,1/λ)` or the function `csgammp(x,t,λ)`. The resulting curves are shown in Figure 2.8.

```
% First get the domain over which to
% evaluate the functions.
x = 0:.1:3;
% Now get the functions values for
% different values of lambda.
y1 = gampdf(x,1,1/1);
y2 = gampdf(x,2,1/2);
y3 = gampdf(x,3,1/3);
% Plot the functions.
plot(x,y1,'r',x,y2,'g',x,y3,'b')
title('Gamma Distribution')
xlabel('X')
ylabel('f(x)')
```

❑

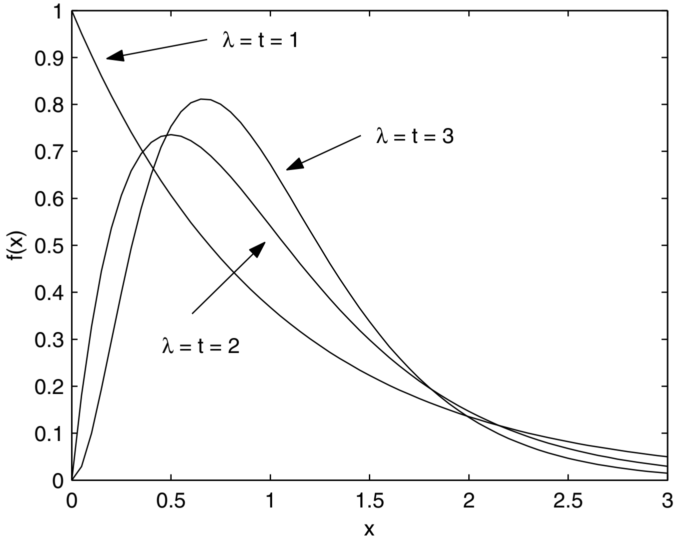


FIGURE 2.8
We show three examples of the gamma probability density function. We see that when $\lambda = t = 1$, we have the same probability density function as the exponential with parameter $\lambda = 1$.

Chi-Square

A gamma distribution where $\lambda = 0.5$ and $t = \nu/2$, with ν a positive integer, is called a ***chi-square distribution*** (denoted as χ^2_ν) with ν degrees of freedom. The chi-square distribution is used to derive the distribution of the sample variance and is important for goodness-of-fit tests in statistical analysis [Mood, Graybill, and Boes, 1974].

The probability density function for a chi-square random variable with ν degrees of freedom is

$$f(x;\nu) = \frac{1}{\Gamma(\nu/2)}\left(\frac{1}{2}\right)^{\nu/2} x^{\nu/2-1} e^{-\frac{1}{2}x}; \qquad x \geq 0 . \tag{2.40}$$

The mean and variance of a chi-square random variable can be obtained from the gamma distribution. These are given by

$$E[X] = \nu$$

and

$$V(X) = 2\nu .$$

Weibull

The ***Weibull distribution*** has many applications in engineering. In particular, it is used in reliability analysis. It can be used to model the distribution of the amount of time it takes for objects to fail. For the special case where $\nu = 0$ and $\beta = 1$, the Weibull reduces to the exponential with $\lambda = (1/\alpha)$.

The Weibull density for $\alpha > 0$ and $\beta > 0$ is given by

$$f(x;\nu, \alpha, \beta) = \left(\frac{\beta}{\alpha}\right)\left(\frac{x-\nu}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x-\nu}{\alpha}\right)^{\beta}}; \qquad x > \nu, \tag{2.41}$$

and the cumulative distribution is

$$F(x;\nu, \alpha, \beta) = \begin{cases} 0; & x \leq \nu \\ 1 - e^{-\left(\frac{x-\nu}{\alpha}\right)^{\beta}}; & x > \nu . \end{cases} \tag{2.42}$$

The location parameter is denoted by ν, and the scale parameter is given by α. The shape of the Weibull distribution is governed by the parameter β.

The mean and variance [Banks, et al., 2001] of a random variable from a Weibull distribution are given by

$$E[X] = \nu + \alpha\Gamma(1/\beta + 1)$$

and

$$V(X) = \alpha^2\left\{\Gamma(2/\beta + 1) - [\Gamma(1/\beta + 1)]^2\right\}.$$

Example 2.8

Suppose the time to failure of piston rings for stream-driven compressors can be modeled by the Weibull distribution with a location parameter of zero, $\beta = 1/3$, and $\alpha = 500$. We can find the mean time to failure using the expected value of a Weibull random variable, as follows

$$E[X] = \nu + \alpha\Gamma(1/\beta + 1) = 500 \times \Gamma(3 + 1) = 3000 \text{ hours.}$$

Let's say we want to know the probability that a piston ring will fail before 2000 hours. We can calculate this probability using

$$F(2000;0, 500, 1/3) = 1 - \exp\left\{-\left(\frac{2000}{500}\right)^{1/3}\right\} \approx 0.796 .$$

❑

You can use the MATLAB Statistics Toolbox function for applications where the location parameter is zero ($\nu = 0$). This function is called **weibcdf** (for the cumulative distribution function), and the input arguments are **(x,**$\alpha^{-\beta}$**,**β**)**. The reason for the different parameters is that MATLAB uses an alternate definition for the Weibull probability density function given by

$$f(x; a, b) = abx^{b-1}e^{-ax^b}; \qquad x > 0 . \tag{2.43}$$

Comparing this with Equation 2.41, we can see that $\nu = 0$, $a = \alpha^{-\beta}$, and $b = \beta$. You can also use the function **csweibc(x,**ν, α, β**)** to evaluate the cumulative distribution function for a Weibull.

Beta

The ***beta distribution*** is very flexible because it covers a range of different shapes depending on the values of the parameters. It can be used to model a random variable that takes on values over a bounded interval and assumes one of the shapes governed by the parameters. A random variable has a beta distribution with parameters $\alpha > 0$ and $\beta > 0$ if its probability density function is given by

$$f(x;\alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha - 1}(1 - x)^{\beta - 1}; \qquad 0 < x < 1, \tag{2.44}$$

where

$$B(\alpha, \beta) = \int_0^1 x^{\alpha - 1}(1 - x)^{\beta - 1} dx = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}. \tag{2.45}$$

The function $B(\alpha, \beta)$ can be calculated in MATLAB using the **beta(**α,β**)** function. The mean and variance of a beta random variable are

$$E[X] = \frac{\alpha}{\alpha + \beta}$$

and

$$V(X) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$

The cumulative distribution function for a beta random variable is given by integrating the beta probability density function as follows

$$F(x;\alpha, \beta) = \int_0^x \frac{1}{B(\alpha, \beta)} y^{\alpha - 1}(1 - y)^{\beta - 1} dy. \tag{2.46}$$

The integral in Equation 2.46 is called the ***incomplete beta function***. This can be calculated in MATLAB using the function **betainc(x,alpha,beta)**.

Example 2.9

We use the following MATLAB code to plot the beta density over the interval (0,1). We let $\alpha = \beta = 0.5$ and $\alpha = \beta = 3$.

```
% First get the domain over which to evaluate
% the density function.
x = 0.01:.01:.99;
% Now get the values for the density function.
y1 = betapdf(x,0.5,0.5);
y2 = betapdf(x,3,3);
% Plot the results.
plot(x,y1,'r',x,y2,'g')
title('Beta Distribution')
xlabel('x')
ylabel('f(x)')
```

The resulting curves are shown in Figure 2.9. You can use the MATLAB Statistics Toolbox function **betapdf(x,**α,β**)**, as we did in the example, or the function **csbetap(x,**α,β**)**.

❑

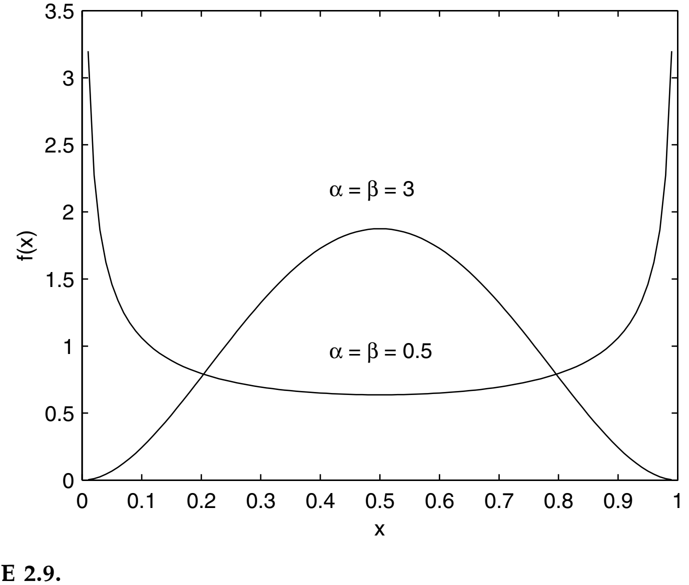


FIGURE 2.9.
Beta probability density functions for various parameters.

Student's *t* Distribution

An important distribution often used in inferential statistics is the *t* distribution. This distribution was first developed by William Gossett in 1908. He published his results under the pseudonym "Student," hence the distribution is sometimes known as the ***Student's t distribution***.

The t distribution comes from the ratio of a standard normal random variable Z to the square root of an independently distributed chi-square random variable U divided by its degrees of freedom ν:

$$X = \frac{Z}{\sqrt{U/\nu}}. \tag{2.47}$$

It can be shown that the density of the random variable in Equation 2.47 is given by

$$f(x;\nu) = \frac{1}{\sqrt{\pi\nu}} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-(\nu+1)/2}. \tag{2.48}$$

The probability density function for the t distribution is symmetric and bell-shaped, and it is centered at zero.

The mean and variance for the t random variable are given by

$$E[X] = 0, \qquad \nu \geq 2,$$

and

$$V(X) = \frac{\nu}{\nu - 2}, \qquad \nu \geq 3.$$

Since it is bell-shaped, the t distribution looks somewhat like the normal distribution. However, it has heavier tails and a larger spread. As the degrees of freedom gets large, the t distribution approaches a standard normal distribution.

Example 2.10

The MATLAB Statistics Toolbox has a function called **tpdf** that creates a probability density function for the Student's t distribution with ν degrees of freedom. The following steps will evaluate the density function for $\nu = 5$.

```
% First we get the domain for the function.
x = -6:.01:6;
% Now get the values for the density function.
y = tpdf(x,5);
% Plot the results.
plot(x,y)
xlabel('x')
ylabel('f(x)')
```

The resulting curve is shown in Figure 2.10. Compare this with the probability density function for the standard normal shown in Figure 2.5, and note the fatter tails with the *t* distribution.

❑

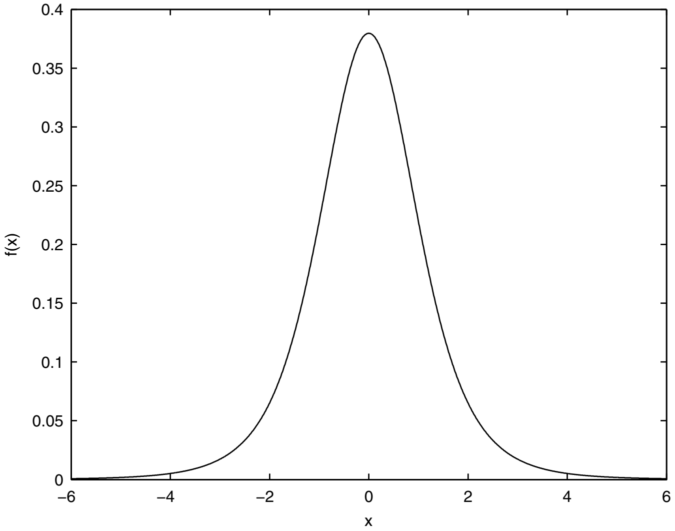


FIGURE 2.10
This illustrates the probability density function for a t random variable with 5 degrees of freedom.

Multivariate Normal

So far, we have discussed several univariate distributions for discrete and continuous random variables. In this section, we describe the ***multivariate normal distribution*** for continuous variables. This important distribution is used throughout the rest of the text. Some examples of where we use it are in exploratory data analysis, probability density estimation, and statistical pattern recognition.

The probability density function for a general multivariate normal density for *d* dimensions is given by

$$f(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}, \qquad (2.49)$$

where $\mathbf{x}$ is a d-component column vector, $\boldsymbol{\mu}$ is the $d \times 1$ column vector of means, and $\boldsymbol{\Sigma}$ is the $d \times d$ covariance matrix. The superscript T represents the transpose of an array, and the notation $||$ denotes the determinant of a matrix.

The mean and covariance are calculated using the following formulas:

$$\boldsymbol{\mu} = E[\mathbf{x}] \tag{2.50}$$

and

$$\boldsymbol{\Sigma} = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T], \tag{2.51}$$

where the expected value of an array is given by the expected values of its components. Thus, if we let X_i represent the i-th component of $\mathbf{x}$ and μ_i the i-th component of $\boldsymbol{\mu}$, then the elements of Equation 2.50 can be written as

$$\mu_i = E[X_i] .$$

If σ_{ij} represents the ij-th element of $\boldsymbol{\Sigma}$, then the elements of the ***covariance matrix*** (Equation 2.51) are given by

$$\sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)] .$$

The covariance matrix is symmetric ($\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}$) positive definite (all eigenvalues of $\boldsymbol{\Sigma}$ are greater than zero) for most applications of interest to statisticians and engineers.

We illustrate some properties of the multivariate normal by looking at the bivariate ($d = 2$) case. The probability density function for a bivariate normal is represented by a bell-shaped surface. The center of the surface is determined by the mean $\boldsymbol{\mu}$, and the shape of the surface is determined by the covariance $\boldsymbol{\Sigma}$. If the covariance matrix is diagonal (all of the off-diagonal elements are zero), and the diagonal elements are equal, then the shape is circular. If the diagonal elements are not equal, then we get an ellipse with the major axis vertical or horizontal. If the covariance matrix is not diagonal, then the shape is elliptical with the axes at an angle. Some of these possibilities are illustrated in the next example.

Example 2.11

We first provide the following MATLAB function to calculate the multivariate normal probability density function and illustrate its use in the bivariate case. The function is called **`csevalnorm`**, and it takes input arguments **`x,mu,cov_mat`**. The input argument **`x`** is a matrix containing the points in the domain where the function is to be evaluated, **mu** is a d-dimensional row vector, and **cov_mat** is the $d \times d$ covariance matrix.

```
function prob = csevalnorm(x,mu,cov_mat);
[n,d] = size(x);
% center the data points
x = x-ones(n,1)*mu;
a = (2*pi)^(d/2)*sqrt(det(cov_mat));
arg = diag(x*inv(cov_mat)*x');
prob = exp((-.5)*arg);
prob = prob/a;
```

We now call this function for a bivariate normal centered at zero and covariance matrix equal to the identity matrix. The density surface for this case is shown in Figure 2.11.

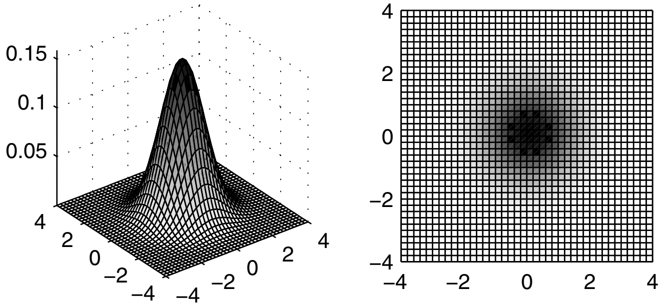


FIGURE 2.11
This figure shows a standard bivariate normal probability density function that is centered at the origin. The covariance matrix is given by the identity matrix. Notice that the shape of the surface looks circular. The plot on the right is for a viewpoint looking down on the surface.

```
% Get the mean and covariance.
mu = zeros(1,2);
cov_mat = eye(2);% Identity matrix
% Get the domain.
% Should range (-4,4) in both directions.
[x,y] = meshgrid(-4:.2:4,-4:.2:4);
% Reshape into the proper format for the function.
X = [x(:),y(:)];
Z = csevalnorm(X,mu,cov_mat);
% Now reshape the matrix for plotting.
z = reshape(Z,size(x));
subplot(1,2,1) % plot the surface
surf(x,y,z),axis square, axis tight
```

```
title('BIVARIATE STANDARD NORMAL')
```

Next, we plot the surface for a bivariate normal centered at the origin with non-zero off-diagonal elements in the covariance matrix. Note the elliptical shape of the surface shown in Figure 2.12.

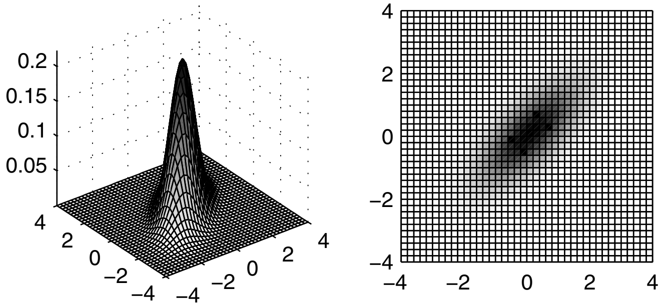


FIGURE 2.12
This shows a bivariate normal density where the covariance matrix has non-zero off-diagonal elements. Note that the surface has an elliptical shape. The plot on the right is for a viewpoint looking down on the surface.

```
subplot(1,2,2) % look down on the surface
pcolor(x,y,z),axis square
title('BIVARIATE STANDARD NORMAL')
% Now do the same thing for a covariance matrix
% with non-zero off-diagonal elements.
cov_mat = [1 0.7 ; 0.7 1];
Z = csevalnorm(X,mu,cov_mat);
z = reshape(Z,size(x));
subplot(1,2,1)
surf(x,y,z),axis square, axis tight
title('BIVARIATE NORMAL')
subplot(1,2,2)
pcolor(x,y,z),axis square
title('BIVARIATE NORMAL')
```

The Statistics Toolbox has a function called **mvnpdf** that will evaluate the multivariate normal density.

❑

The probability that a point $\mathbf{x} = (x_1, x_2)^T$ will assume a value in a region R can be found by integrating the bivariate probability density function over the region. Any plane that cuts the surface parallel to the x_1, x_2 plane intersects in an elliptic (or circular) curve, yielding a curve of constant density. Any plane perpendicular to the x_1, x_2 plane cuts the surface in a normal curve. This property indicates that in each dimension, the multivariate normal is a univariate normal distribution.

Multivariate *t* Distribution

The univariate Student's t distribution can also be generalized to the multivariate case. The ***multivariate t distribution*** is for d-dimensional random vectors where each variable has a univariate t distribution. Not surprisingly, it is used in applications where the distributions of the variables have fat tails, thus providing an alternative to the multivariate normal when working with real-world data.

A d-dimensional random vector

$$\mathbf{x}^T = (X_1, \ldots, X_d)$$

has the t distribution if its joint probability density function is given by

$$f(\mathbf{x};\mathbf{R},\nu,\boldsymbol{\mu}) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{(\pi\nu)^{d/2}\Gamma\left(\frac{\nu}{2}\right)|\mathbf{R}|^{1/2}}\left[1+\frac{(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{R}^{-1}(\mathbf{x}-\boldsymbol{\mu})}{\nu}\right]^{-(\nu+d)/2} \tag{2.52}$$

The multivariate t distribution has three parameters: the correlation matrix $\mathbf{R}$, the degrees of freedom ν, and the d-dimensional mean $\boldsymbol{\mu}$. If the mean is zero ($\boldsymbol{\mu} = \mathbf{0}$), then the distribution is said to be central. Similar to the univariate case, when the degrees of freedom gets large, then the joint probability density function approaches the d-variate normal.

The correlation matrix is related to the covariance matrix via the following relationship:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}},$$

where ρ_{ij} is the ij-th element of $\mathbf{R}$. Note that the diagonal elements of $\mathbf{R}$ are equal to one.

Example 2.12

The MATLAB Statistics Toolbox has several functions for the central multivariate t distribution. These include the **`mvtpdf`** (evaluates the multivariate t probability density function) and the **`mvtcdf`** (computes the cumulative probabilities). These functions use the d-variate t distribution